SİSTEM VE AĞ TEKNOLOJİLERİ

# Cisco Ağ Teknolojileri Yönetimi

Todd Lammle

SYBEX

2. Basım

**Çeviren** Ferhat Baş
**Editör** C. Banu Üçüncüoğlu

29-9

# Cisco Ağ Teknolojileri Yönetimi

## Todd Lammle

*CCNA: Cisco Certified Network Associate Study Guide: Exam 640-802, 6th Edition*

Çeviren: **Ferhat Baş**

Çeviri Editörü: **C. Banu Üçüncüoğlu**

Çeviri Danışmanları: **Fikri Bülent Çelik – Erdoğan Bilici**

Kapak Tasarımı: **Melih Sancar**

Grafik Uygulama: **Tuna Erkan**

Yayın Yönetmeni: **Selçuk Tüzel**

Genel Yayın Yönetmeni: **Mehmet Çömlekçi**

1. Basım: Nisan 2008

2. Basım: Kasım 2008

Rev: 00

Bilge Adam Yayınları: 31

Eğitim Yayınları Dizisi: 31

ISBN: 978-605-5987-29-9

Yayıncı Sertifika No: 1107-34-009150



Bilge Adam Bilgisayar ve Eğitim Hizmetleri San. ve Tic. A.Ş.

19 Mayıs Mahallesi, 19 Mayıs Caddesi, UBM Plaza, No: 59-61, Kat: 4-7; Şişli, İstanbul

Telefon: (212) 272 76 00 – (212) 217 05 55 Faks: (212) 272 76 01

www.bilgeadam.com - info@bilgeadam.com

# İçindekiler

# *Giriş*

Cisco sertifikasyonun heyecanlı dünyasına hoş geldiniz. Daha iyi olmak istediğinizden, bu kitaba başlıyorsunuz. İyi bir karar verdiğinize emin olabilirsiniz. Cisco sertifikasyon, sizin ilk network işinize girmenize veya zaten bu alanda çalışıyorsanız, daha fazla para kazanmanıza ve bir terfi almanıza yardımcı olabilir.

Cisco sertifikasyon, ağlar arası iletişimi Cisco ürünleri dışında da anlamanızı geliştirebilir: Siz, ağ kurulumu ve bir ağı şekillendirmek için farklı network topolojilerinin birlikte nasıl çalıştıkları anlayışını bir bütün olarak geliştireceksiniz. Bu, her network işi için faydalıdır ve çok az Cisco cihaza sahip şirketlerde bile Cisco sertifikasyon istenmesinin sebebidir.

Cisco, routing, switching ve güvenliğin kralıdır, yani ağlar arası iletişim dünyasının Microsoft'udur. Cisco sertifikasyon, günümüz ağını kavramada zorunlu sebeplerini size sağlamak için MCSE gibi popüler sertifikasyonların üzerine çıkmıştır. Cisco belgeli olmak istemeye karar vererek, routing ve switching'de en iyi olmak istediğinizi belirtmiş oluyorsunuz. Bu kitap size bu yolda kılavuzluk yapacaktır.

*Hem CCNA sertifikasyon sınavlarına ilaveleri ve değişiklikleri içeren en son güncellemeler hem de ilave çalışma araçları ve gözden geçirme soruları için www.lammle.com ve/veya www.sybex.com web sitelerini ve Todd Lammle forumunu ziyaret edin.*

**NOT**

## Cisco'nun Network Destek Sertifikasyonları

Başlangıçta hedeflenen Cisco CCIE sertifikasyonu elde etmek için sadece bir test alıyordunuz ve sonra, onun başarılmasını güçleştiren, ya hep ya hiç yaklaşımında bir (oldukça zor) pratik lab ile karşılaşırdınız.

Karşılığında, Cisco, hedeflenen CCIE'ye sahip olmanız için ayrıca beklenen çalışanların bilgi seviyelerinin ölçülmesinde yardım amaçlı olarak, yeni sertifika serileri oluşturdu. Bu sertifikasyonlarla beraber Cisco, daha önce çok az kişinin geçebildiği kapıları açtı.

*Bu kitap CCNA ile ilgili her şeyi içermektedir. Todd Lammle Cisco Yetkili CCNA, CCNP, CCSP, CCVP ve CCIE bootcamp'lerle ilgili güncel bilgiler için www.lammle.com ve/veya www.globalnettraining.com web sitelerini ziyaret edebilirsiniz.*

**NOT**

## Cisco Certified Network Associate (CCNA)

CCNA sertifikasyonu, Cisco sertifikasyonun yeni çizgisinde ilkti ve şimdiki Cisco sertifikasyonların habercisiydi. Bu nedenle gerçekten çok şey biliyor olmalısınız. Bir Cisco eğitimi almak veya pratik tecrübeyle aylarca zaman harcamak normaldir.

CCNA'inizi aldığınızda, orada kalmamalısınız. Cisco Certified Network Professional (CCNP) olarak bilinen daha ileri seviye sertifikasyon için çalışmayı tercih edebilirsiniz. CCNP'ye sahip birisi, Routing ve Switching CCIE lab'ını denemek için gerekli beceri ve bilgilerin hepsine sahiptir. Fakat sadece CCNA olmak, sizi hayal ettiğiniz işe kavuşturabilir.

## Neden Bir CCNA Olunur?

Cisco, Microsoft ve Novell'den (Linux) faksız şekilde, yöneticilere birçok beceri sağlamak ve geleceğin çalışanlarını, becerileri ölçme veya belirli kriteri karşılaştırmanın bir yoluyla donatmak için sertifikasyon prosesi oluşturdu. Bir CCNA olmak, oldukça istenilen ve güçlendirilebilir kariyere doğru başarılı bir yolculuğun ilk adımı olabilir.

CCNA programı, sadece Cisco Internetwork Operating System (IOS) ve Cisco donanım için değil, ayrıca genel ağlar arası iletişim içinde, güçlü bir giriş sağlamak için oluşturuldu. Bu onu, Cisco'nun olmadığı alanlarda da sizin için kullanışlı yapar. Sertifikasyon prosesinin bu noktasında, network müdürlerinin, Cisco ekipmanı olmaksızın dahi, iş başvuranları için Cisco sertifikalarını zorunlu tutmaları olmayacak şey değildir.

Hala Cisco ve ağlar arası iletişimle ilgileniyorsanız, kesin başarı yolunda ilerliyorsunuz.

## CCNA Olmak İçin Hangi Vasıflara Sahip Olmalısınız?

CCNA sertifika yetenek seviyesine erişmek için aşağıdakileri anlayabilmeli veya yapabilmelisiniz:

- Sertifikalı bir CCNA uzmanı, LAN, WAN ve wireless erişim servislerini güvenli olarak kurup, yapılandırıp, çalıştırabilmeli hem de küçük ölçekliden, orta ölçeklilere kadar ağları (500 düğüm ve daha azı) yapılandırıp, arıza tespiti yapabilmelidir.
- Bu bilgi, şu protokollerin kullanımını içerir (fakat bunla sınırlı değildir): IP, IPv6, EIGRP, RIP, RIPv2, OSPF, seri bağlantılar, Frame Relay, kablo, DSL, PPPoE, LAN switching, VLAN'lar, Ethernet, güvenlik ve access list'ler.

NOT *Anlık bir uyarı ile değişebilen en son Cisco CCNA konuları ve diğer Cisco sınavları, konuları ve sertifikasyonları için web sitemi ve/veya Cisco'nun web sitesini kontrol edin.*

## Nasıl CCNA Olursunuz?

CCNA olmanın yolu, küçük bir testi geçmektir (CCNA Composit 640-802 sınavını). Artık CCNA'sinizdir. (Bu kadar kolay olmasını dilemez misiniz?) Doğru, o tek bir test olabilir, fakat hala test hazırlayanların dediklerini anlamak için yeterli bilgiye sahip olmak zorundasınız.

Bununla beraber Cisco, CCNA olmak için alabileceğiniz iki aşamalı bir işleyişe sahiptir. (Bu kitap, tek aşamalı 640-802 temellidir). Yinede, bu kitap tüm üç sınavı geçmeniz için gerekli tüm bilgilere sahiptir.

İki-aşamalı yöntem, aşağıdakileri geçmeyi gerektirir:

- 640-822 Sınavı: Interconnecting Cisco Networking Devices 1 (ICND1)
- 640-816 Sınavı: Interconnecting Cisco Networking Devices 2 (ICND2)

Cisco router'larla bazı pratik tecrübeye sahip olmanız önemlidir. Bazı 1841 veya 2800 serisi router'larla tecrübe edebilirseniz, hazır olursunuz. Fakat yapamazsanız, network yöneticilerinin (veya network yöneticisi olmak isteyenlerin), CCNA sınavını geçmeleri için bilmeleri gerekenleri öğrenmesine yardımcı olması için, bu kitap boyunca yüzlerce konfigürasyon örneği hazırlamak için çok çalıştım.

Yeni 640-802 sınavı çok zor olduğundan Cisco, iki-test yaklaşımı için sizi ödüllendirmek istiyor ya da öyle görünüyor. ICND1 sınavını geçerseniz, CCENT (Cisco Certified Entry Networking Technician) olarak belirtilen bir sertifika alıyorsanız. Bu CCNA yolundaki ilk adımınızdır. CCNA'i almanız için, hala ICND2 sınavını geçmeniz gerekmektedir.

NOT *Todd Lammle ile Cisco yetkili pratik eğitimi için www.globalnettraining.com web sitesine bakın. Her öğrenci, ekipmanları paylaşmadan, en az üç router ve iki switch'i yapılandırarak pratik tecrübeye sahip olacaktır.*

Bu kitap, CCNA 640-802 Composit sınavı için yazılmıştır. Bu tek sınavdır ve sertifikanızı alırsınız.

## Bu Kitap Neleri Kapsar?

Bu kitap, CCNA 640-802 sınavını geçmeniz için bilmeniz gereken her şeyi içermektedir. Bununla beraber, router ve router simülatörleriyle çalışmak ve pratik için zaman ayırmak, başarının gerçek anahtarıdır.

Bu kitapta aşağıdakileri öğreneceksiniz:

- Bölüm 1, sizi ağlar arası iletişim ile tanıştıracak. Open Systems Interconnection (OSI) modelinin temellerini, Cisco'nun öğrenmenizi istediği yolla öğreneceksiniz. Ethernet ağı ve standartları, yine bu modülde detaylı şekilde incelenmektedir. Size yardımcı olması için, yazılı lab'lar ve bol miktarda gözden geçirme sorusu vardır. Bu modüldeki yazılı lab'ları sakın atlamayın!

- Bölüm 2, hem sınavın başarısı için hem de gerçek dünyada TCP/IP tartışmasında, gerekli altyapıyı sağlar. Bu, Internet Protokol yığınının oldukça başını içeren ve son olarak, network arıza tespiti ile sona ermeden önce, network ve broadcast adres arasındaki farklılığın anlaşılmasına ve IP adreslemesine değinen bölümdür.
- Bölüm 3, sizi subnet'leme ile tanıştırır. Bu bölümü okuduktan sonra, bir ağı, kafadan subnet'lerine ayırabileceksiniz. İlave olarak, Variable Length Subnet Mask'ı (VLSM) ve VLSM'leri kullanarak bir ağın nasıl tasarlanacağı hakkında bilgi alacaksınız. Bu bölüm, summarization teknikleri ve yapılandırmalar ile tamamlanacak. Yazılı lab ve gözden geçirme sorularını atlamazsanız, bu bölümde çok sayıda katkı bulursunuz.
- Bölüm 4, sizi Cisco Internetwork Operating System (IOS) ve command-line interface (CLI) ile tanıştırmaktadır. Bu bölümde, router'ın nasıl başlatılacağını ve şifre, banner ile daha fazlasını içerecek şekilde IOS'un temellerinin nasıl yapılandırılacağını öğreneceksiniz. Secure Device Manager (SDM) kullanarak IP yapılandırması işlenecek ve pratik lab'lar, bu bölümde öğretilen kavramların güçlü bir kavrayışını kazanmanıza yardım edecektir. Pratik Lab'lara geçmeden önce, Yazılı lab ve gözden geçirme sorularını tamamladığınızdan emin olun.
- Bölüm 5 size, bir Cisco IOS'ta çalışması için gerekli yönetim becerilerini kazandıracaktır. Hem IOS'u yedeklemek ve yeniden yüklemek hem de router yapılandırması, ağınızı çalışır olarak korumanız için gerekli hata tespit araçları olarak, kapsanmaktadır. Bu bölümdeki Pratik Lab'ları yapmadan önce, Yazılı Lab ve gözden geçirme sorularını tamamlayın.
- Bölüm 6, IP routing'i konusunu öğretecektir. Ağınızı kurmaya başlayacağınız, IP adresleri ekleyeceğiniz ve router'lar arasında veriyi route ettireceğiniz için bu bölüm eğlenceli bir bölümdür. Ayrıca RIP ve RIPv2 (IGRP'ye çok az değinerek) kullanarak statik, default ve dinamik routing konusunu da öğreneceksiniz. Yazılı ve Pratik Lab'lar, IP routing'i en iyi şekilde anlamanıza yardımcı olacaktır.
- Bölüm 7, Enhanced IGRP ve OSPF routing ile daha karmaşık dinamik routing'e gireceğim. Yazılı Lab, Pratik Lab ve Gözden Geçirme Soruları, bu routing protokollerini iyice anlamanıza yardımcı olacaktır.
- Bölüm 8, katman2 switching ve switch'lerin adres öğrenme ve iletme veya filtreleme kararlarını nasıl verdiğiyle ilgili iyi bir temel oluşturur. Hem network döngüleri ve Spanning Tree Protocol (STP) ile onlardan nasıl kaçınılacağı hem de 802.1w STP versiyonu konuşulacaktır. Bir ağ topluluğunda basit katman 2 switching'i anlamınıza gerçekten yardımcı olması için Yazılı Lab ve Gözden Geçirme Sorularını inceleyin.
- Bölüm 9, virtual LAN'ları ve onları ağ topluluğunuzda nasıl kullanabileceğinizi içermektedir. Bu bölüm ayrıca, hem VLAN'ların özünü, VLAN'larla kullanılan farklı kavram ve protokolleri hem de hata tespitini kavramaktadır. Yazılı Lab ve Gözden Geçirme soruları, VLAN materyallerini pekiştirecektir.
- Bölüm 10, güvenlik ve ağı filtrelemek için router'larda oluşturulan access list'leri içermektedir. IP standart, extended ve named access list'ler detaylı şekilde işlenmektedir. Gözden Geçirme soruları ile beraber, Yazılı Lab ve Pratik Lab'lar, CCNA Composit sınavının güvenlik ve access list'lerle ilgili bölümüne hazırlanmanıza yardımcı olacaktır.
- Bölüm 11, Network Address Translation'ı (NAT) içermektedir. Bu bölüm, benin son CCNA kitabıma bir güncelleme olarak, birkaç yıldır Sybex web sitesinde yer almaktadır. Fakat ben onu güncelledim ve bu baskıya ekledim. Yeni bilgi, komutlar, hata tespiti ve lab'lar, NAT CCNA konularını halletmenize yardımcı olacaktır.
- Bölüm 12, Wireless teknolojilerini kapsar. Bu, wireless teknolojilerini, Cisco'nun onları anladığı şekilde anlatan tanıtıcı bir bölümdür. Bununla beraber, Cisco'nun en yeni cihazlarının içerdiği gelişmiş wireless konularını da ekledim. Gelişmiş wireless, Cisco CCNA konularında yer almadı. Fakat bu durum bir gün değişebilir. Hem access point'ler gibi basit wireless teknolojilerini hem de 802.11a, b ve g arasındaki farkları anladığınıza emin olun.

- Bölüm 13, IPv6'yı içermektedir. Bu eğlenceli bir bölümdür ve bazı önemli bilgilere sahiptir. IPv6, birçok insanın düşündüğü gibi, büyük, kötü, ürkütücü bir canavar değildir. IPv6, bu yeni sınavın bir konusudur, o nedenle bu bölümü dikkatli çalışın. Son güncellemeler için www.lammle.com web sitesine bakın.
- Bölüm14, Cisco wide area network (WAN) protokollerine odaklanmaktadır. Bu bölüm, hem HDLC, PPP ve Frame Relay'i hem de Kablo, DSL ve PPPoE gibi günlük kullandığımız diğer protokolleri detaylı şekilde işlemektedir. CCNA sınavında başarılı olmak için, tüm bu protokollerde uzman olmanız gerekmektedir. Bu bölümdeki, Yazılı Lab, Gözden Geçirme Soruları ve Pratik Lab'ları sakın atlamayın.

## Bu Kitap Nasıl Kullanılmalı?

Cisco Certified Network Associate (CCNA Composite) 640-802 sınavına sıkı bir şekilde hazırlanırken, güçlü bir temel istiyorsanız, bu sizin için yeterlidir. CCNA sınavını geçmeniz ve Cisco router ve switch'leri nasıl yapılandıracağını öğrenmenize yardımcı olmak niyetiyle, bu kitabı derlemek için yüzlerce saat harcadım.

Bu kitap önemli bilgilerle donatılmıştır ve kitabı nasıl oluşturduğumu anlarsanız, çalışma zamanınızdan daha çok verim alabilirsiniz.

Bu kitaptan en iyi şekilde faydalanmak için aşağıdaki çalışma yöntemlerini öneriyorum:

1. Bu girişin devamındaki değerlendirme testine bakın. (cevaplar, testin sonundadır) Cevapların hiçbirini bilmiyorsanız, önemli değil, bu nedenle bu kitabı aldınız. Yanlış yaptığınız soruların cevaplarının açıklamalarını dikkatlice okuyun ve materyalin işlendiği bölümleri not alın. Bu bilgi, sizin çalışma stratejinizi planlayacaktır. Açıkçası ben, Modül 1'den okumaya başlamanızı ve Bölüm 14'e kadar durmamanızı tavsiye ederim.
2. Her birinin başlangıcında listelenen test konularını ve bilgileri tamamıyla anladığınıza emin olarak, her bölümü dikkatle çalışın. Değerlendirme testinde cevaplayamadığınız materyalleri kapsayan bölümlereekstra özen gösterin.
3. Her bölümün sonundaki Yazılı Lab'ları tamamlayın. CCNA Composite konuları ile direk ilgili olmalarından dolayı, bu Yazılı Lab alıştırmalarını sakın atlamayın. Bu lab'ları hızlıca geçmeyin. Her cevabın sebebini tamamıyla anladığınıza emin olun.
4. Bölümdeki Pratik Lab'ları tamamlayın (her bölümde Pratik Lab'lar yoktur). Bölümdeki yazılanlara dönerek, altığınız her adımın sebebini anlarsınız. Şayet uygun Cisco ekipmanlarına sahip değilseniz, Cisco sertifikanızın gerektirdiği tüm pratik lab'ları içerecek bir router simülatörü için www.routersim.com web sitesine gidin.
5. Her bölümle ilgili Gözden Geçirme Sorularının tamamını cevaplandırın.(Bu cevaplar, bölümlerin sonunda görünmektedir). Kafanızı karıştıran soruları not alın ve kitabın bu bölümlerini tekrar çalışın. Bu soruları hızlıca geçmeyin. Her cevabın nedenlerini tamamıyla anladığınıza emin olun. Bunların, sınavda karşılaşacaklarınızla aynı sorular olmayacağını hatırlayın. Bu sorular, bölüm materyalini anlamanıza yardımcı olması için hazırlanmıştır.

## Sınavları Nereden Alırız?

CCNA Composite sınavını, herhangi bir Pearson VUE yetkili test merkezlerinden (www.vue.com) alabilir veya 877-404-EXAM (3926) arayabilirsiniz.

Bir Cisco Certified Network Associate sınavına kayıt yaptırmak için:

1. Almak istediğiniz sınavın numarasını belirtin (CCNA Composit sınav numarası, 640-802'dir)
2. En yakın Pearson VUE test merkezine kayıt yaptırın. Bu noktada, sınav için peşin ödeme yapmanız istenecektir. Sınavınızı, 6 hafta öncesine kadar veya almak istediğiniz aynı güne kadar programlayabilirsiniz. Şayet sınavda başarısız olursanız, tekrar almadan önce beş gün

beklemek zorundasınız. Bir şey çıkar veya sınavınızı iptal etmeniz ya da tarihini değiştirmeniz gerekirse, Prometric veya Pearson VUE ile en az 24 saat önceden irtibata geçin.

3. Sınavınızı planladığınızda, randevu ve iptal prosedürleri, ID gereklilikleri ve test merkezi lokasyonuyla ilgili bilgiler alacaksınız.

## CCNA Composite Sınavı Almanın İpuçları

CCNA Composit sınav testi, 55-60 arası soru içerir ve 75-90 arası veya daha az sürede tamamlanması gerekir. Bu bilgi, her sınav için değişebilir. Sınavı geçmek için %85 puan almanız gerekir, fakat her sınav için farklı olabilir.

Sınavdaki birçok soru, ilk bakışta benzer gözüken (özellikle de syntax soruları), çoktan seçmelidir. Yakın olması kabul edilmeyeceği için, şıkları dikkatle okumayı unutmayın. Komutları yanlış sırayla kullanırsanız veya tek karakteri unutursanız, cevabınız yanlış olacaktır. Bu nedenle, sizde alışkanlık yapana kadar, her bölümün sonundaki pratik sınavları yapın.

Ayrıca, doğru cevabın, Cisco'nun cevabı olduğunu unutmayın. Çoğu durumda, birden fazla doğru cevap mevcuttur, fakat doğru cevap, Cisco'nun önerdiğidir. Sınavda, bir, iki veya üç şık seçmeniz istenir, tamamını seçmeniz asla istenmez. CCNA Composit 640-802 sınavı, aşağıdaki test formatlarını içerir:

- Tek cevaplı çoktan seçmeli
- Çok cevaplı çoktan seçmeli
- Sürükle-bırak
- Router simülasyonu

Çoktan seçmelilere ve doldurmalı sorulara ilave olarak, Cisco kariyer sertifikasyonları, performans simülasyon sınav öğeleri içerebilir.

Çoklu routing protokolleri çalışan, sınırsız sayıda Cisco router ve switch'i tasarlamanıza ve yapılandırmanıza izin verecek yazılımı www.routersim.com adresinden kontrol edin.

CD'deki yazılım ve RouterSim.com, Cisco router ve switch'lerin nasıl yapılandırıldığını adım-adım anlatan komutlar sağlar. Bununla beraber, Cisco sınavlarındaki router simülasyonları, bir router interface yapılandırmasının tamamlanması için izlenecek adımları göstermeyecektir. Onlar, kısmi komut yanıtlarına izin verecektir. Örnek olarak, show config veya `sho config` ya da `sh conf`, kabul edilecektir. `Router#show ip protocol` veya `router#show ip prot` kabul edilecektir.

Sınavda başarı için bazı genel ipuçları aşağıdadır:

- Sınav merkezine erkenden ulaşın. Böylece, rahatlayabilir ve sınav materyallerine göz atabilirsiniz.
- Soruları dikkatli şekilde okuyun. Yeterince düşünmeden karar vermeyin. Her sorunun tam olarak neyi sorduğunu anladığınızdan emin olun.
- Çok emin olmadığınız çoktan seçmeli soruları cevaplandırdığınızda, ilk olarak açık şekilde yanlış olan cevapların elenmesi prosesini izleyin. Bunu yapmak, iyi bir tahmin yapmanız gerektiğinde, olasılıklarınızı çok arttırır.
- Cisco sınavı boyunca, ileri veya geri gidemezsiniz. Kararınızı değiştiremeyeceğinizden, Next butonuna tıklamadan, cevabınızı iki defa kontrol edin.

Bir sınavı tamamladıktan sonra, sizin sınavı geçme veya kalma durumunuzu ve bölümlere göre sınav sonuçlarını belirten, yazılı bir Sınav Sonuç Raporu'nu hemen alacaksınız. (Test sorumlusu size yazılı sonuç raporu verecektir.) Test sonucu, testi aldıktan sonraki beş gün içinde, otomatik olarak Cisco'ya gönderilecektir. Şayet sınavı geçtiyseniz, tipik olarak iki ile dört hafta, bazen de daha uzun sürede, Cisco'dan doğrulama alacaksınız.

## Yazar ile Bağlantıya Geçmek

Todd Lammle'ye, www.lammle.com web sitesindeki forumundan ulaşabilirsiniz.

# Değerlendirme Testi

1. Hangi protokol, Network katman protokolünü belirlemek için PPP kullanır?
   A. NCP
   B. ISDN
   C. HDLC
   D. LCP
2. 10Mbps half-duplex çalışan bir hub'a bağlı 10 kullanıcınız vardır. Ayrıca, 10Mbps half-duplex çalışan bir switch'e bağlı bir sunucunuz var. Her kullanıcı, sunucuya ne kadar bant genişliğine sahiptir?
   A. 100kbps
   B. 1Mbps
   C. 2Mbps
   D. 10Mbps
3. Düzinelerce switch'e sahip bir ağda, kaç tane root bridge'e sahip olursunuz?
   A. 1
   B. 2
   C. 5
   D. 12
4. `routerA(config)#line cons 0` komutu, sonrasında ne yapmanıza izin verir?
   A. Telnet şifresini ayarlamak.
   B. Router'ı kapatmak.
   C. Konsol şifresini ayarlamak.
   D. Konsol bağlantılarını disable etmek.
5. IPv6 adresinin boyutu ne kadardır?
   A. 32 bit
   B. 128 byte
   C. 64 bit
   D. 128 bit
6. Hangi PPP protokolü, dinamik adresleme, authentication ve multilink sağlar?
   A. NCP
   B. HDLC
   C. LCP
   D. X.25
7. Hangi komut bir interface'in, line, protokol, DLCI ve LMI bilgisini gösterecektir?
   A. sh pvc
   B. show interface
   C. show frame-relay pvc
   D. sho runn

8. Aşağıdakilerden hangisi, 192.168.168.188 255.255.255.192 subneti için geçerli host aralığıdır?
   A. 192.168.168.129–190
   B. 192.168.168.129–191
   C. 192.168.168.128–190
   D. 192.168.168.128–192
9. `Passive` komutu, dinamik routing protokollerine ne sağlar?
   A. Bir interface'in periyodik güncelleme gönderip almasını durdurur.
   B. Bir interface'in periyodik güncelleme almasını durdurur, fakat göndermeye devam eder.
   C. Router'ın dinamik bir güncelleme almasını durdurur.
   D. Router'ın dinamik bir güncelleme göndermesini durdurur.
10. Hangi protokol ping'i kullanılır yapar?
   A. TCP
   B. ARP
   C. ICMP
   D. BootP
11. Bir ağı, 12 port switch ile segmentlerine ayırdığınızda, kaç tane collision domain oluşturulur?
   A. 1
   B. 2
   C. 5
   D. 12
12. Bir Cisco router'da Telnet şifrenizi oluşturmanızı aşağıdaki komutlardan hangisi sağlayacaktır?
   A. line telnet 0 4
   B. line aux 0 4
   C. line vty 0 4
   D. line con 0
13. Hangi router komutu, tüm access list'lerin içeriğinin tamamını görmenize izin verir?
   A. show all access-lists
   B. show access-lists
   C. show ip interface
   D. show interface
14. Bir VLAN ne yapar?
   A. Tüm sunuculara, en hızlı port gibi davranır.
   B. Bir switch port'unda çoklu collision domain'ler sağlar.
   C. Katman2 bir switch ağ topluluğundaki broadcast domainleri ayırır.
   D. Tek bir collision domain'inde çoklu broadcast domain'ler sağlar.

15. NVRAM'de tutulan yapılandırmayı silmek isterseniz, ne yazarsınız?
    A. erase startup
    B. erase nvram
    C. delete nvram
    D. erase running
16. Oluşturan host'lara, geri, hedef ağın bilinmediği mesajını göndermek için hangi protokol kullanılır?
    A. TCP
    B. ARP
    C. ICMP
    D. BootP
17. Hangi IP adres klasında, varsayılan olarak en fazla host adresi kullanılabilir?
    A. A
    B. B
    C. C
    D. A ve B
18. Bir katman2 cihazdan hangi sıklıkta BPDU'lar gönderilir?
    A. Hiçbir zaman
    B. Her 2 saniyede
    C. Her 10 dakikada
    D. Her 30 saniyede
19. VLAN'lar hakkında aşağıdakilerden hangisi doğrudur?
    A. Tüm Cisco switch'lerde, varsayılan olarak iki VLAN yapılandırılır.
    B. VLAN'lar sadece, tamamıyla Cisco switch'lere sahip olduğunuz bir ağ topluluğunda çalışırlar. Bilinmeyen marka switch'ler kabul edilmeyecektir.
    C. Aynı VTP domain'inde, 10'dan fazla switch'e sahip olmamalısınız.
    D. VTP, yapılandırılmış bir VTP domain'indeki switch'lere VLAN bilgisi göndermek için kullanılmaktadır.
20. Hangi WLAN IEEE düzenlemesi, 2.4Ghz'de 54Mbps'a izin verir?
    A. A
    B. B
    C. G
    D. N
21. Bir ağ 12 portlu switch ile segmentlerine ayrıldığında, kaç tane broadcast domain oluşturulur?
    A. 1
    B. 2
    C. 5
    D. 12

22. Tek IP adresi ile çok sayıda kullanıcıyı, global İnternet'e bağlamak için hangi Network Address Translation çeşidi kullanılabilmektedir?
    A. NAT
    B. Statik
    C. Dinamik
    D. PAT
23. Bir switch'te trunking yapılandırmak için hangi protokoller kullanılmaktadır? (İki şık seçin)
    A. VLAN Trunking Protocol
    B. VLAN
    C. 802.1Q
    D. ISL
24. Bir stub ağ nedir?
    A. Birden fazla çıkış noktası olan bir network.
    B. Birden fazla çıkış ve giriş noktası olan bir network.
    C. Yalnız bir girişi olan ve çıkış noktası olmayan bir network.
    D. Sadece tek bir giriş ve çıkış noktasına sahip bir network.
25. OSI modelinde bir hub nerede belirtilir?
    A. Session katmanı
    B. Physical katmanı
    C. Data Link katmanı
    D. Application katmanı
26. Access Control List'lerin (ACL) iki ana türü nedir?
    A. Standard
    B. IEEE
    C. Extended
    D. Specialized
27. Bir IOS'un yedeğini almak için, hangi komutu kullanırsınız?
    A. backup IOS disk
    B. copy ios tftp
    C. copy tftp flash
    D. copy flash tftp
28. Bir yedek yapılandırma oluşturmak için hangi komut kullanılmaktadır?
    A. copy running backup
    B. copy running-config startup-config
    C. config mem
    D. wr mem

29. OSI modelin oluşturulmasının ana sebebi nedir?
    A. DoD modelinden daha geniş katmanlı bir model oluşturmak için.
    B. Böylece uygulama geliştiriciler, tek seferde sadece bir katmanın protokolünü değiştirebilirler.
    C. Böylece farklı ağlat haberleşebilir.
    D. Böylece Cisco modeli kullanabilir.
30. DHCP, Transport katmanında hangi protokolü kullanır?
    A. IP
    B. TCP
    C. UDP
    D. ARP
31. Router'ınız, bir CSU/DSU'ya yardım ediyorsa, router'a 64000bps seri link sağlamak için kullanmanız gereken komut aşağıdakilerden hangisidir?
    A. RouterA(config)#bandwidth 64
    B. RouterA(config-if)#bandwidth 64000
    C. RouterA(config)#clockrate 64000
    D. RouterA(config-if)#clock rate 64
    E. RouterA(config-if)#clock rate 64000
32. Belirli bir interface'de bir IP access list'in etkin olup olmadığını anlamak için hangi komut kullanılmaktadır?
    A. show access-lists
    B. show interface
    C. show ip interface
    D. show interface access-lists
33. Bir Cisco router'daki IOS'u güncellemek için hangi komut kullanılmaktadır?
    A. copy tftp run
    B. copy tftp start
    C. config net
    D. copy tftp flash

# Değerlendirme Testinin Cevapları

1. A Pakette kullanılan Network katman protokolünü belirlemeye yardım etmesi için Network Control Protocol kullanılır. Daha fazla bilgi için Bölüm 14'e bakın.
2. D her cihaz sunucuya 10Mbps'a sahiptir. Daha fazla bilgi için Bölüm 8'e bakın.
3. A Network başına sadece bir root bridge'e sahip olmalısınız. Daha fazla bilgi için Bölüm 8'e bakın.
4. `line console 0` komutu sizi, sonra konsol user-mode şifresini ayarlayacağınız bir istemciye yerleştirilir. Daha fazla bilgi için Bölüm 4'e bakın.
5. D Bir IPv6 adresi, 32 bit olan IPv4 adresi ile karşılaştırıldığında, 128 bit uzunluğundadır. Daha fazla bilgi için Bölüm 13'e bakın.
6. C PPP yığınındaki Link Control Protocol'ü, dinamik adresleme, authentication ve multilink sağlar. Daha fazla bilgi için Bölüm 14'e bakın.
7. B `show interface` komutu, bir interface'in line, protokol, DLCI ve LMI bilgisini gösterir. Daha fazla bilgi için Bölüm 14'e bakın.
8. 256 – 192 = 64. 64 + 64 = 128. 128 + 64 = 192. Subnet 128, broadcast adresi, 191 ve geçerli host aralığı, 129-190'dır. Daha fazla bilgi için, Modül 3'e bakın.
9. B `passive-interface` komutunun kısa şekli olarak `passive` komutu, bir interface'den düzenli güncellemelerin gönderilmesini durdurur. Bununla beraber, interface hala güncellemeleri alabilecektir. Daha fazla bilgi için Bölüm 6'ya bakın.
10. ICMP, echo request ve reply'lar göndermek için kullanılan, Network katmanındaki protokoldür. Daha fazla bilgi için, Modül 2'ye bakın.
11. D Katman2 switching, ayrı collision domain'ler oluşturur. Daha fazla bilgi için Bölüm 1'e bakın.
12. C `line vty 0 4` komutu sizi, Telnet şifrenizi ayarlayıp değiştirmenize izin verecek bir istemciye götürür. Daha fazla bilgi için Bölüm 4'e bakın.
13. Tüm access list'lerin içeriğini görmek için, `show access-lists` komutunu kullanın. Daha fazla bilgi için Bölüm 10'a bakın.
14. C VLAN'lar, katman2'deki broadcast domain'lerini ayırırlar. Daha fazla bilgi için Bölüm 9'a bakın.
15. A `erase startup-config` komutu, NVRAM'de tutulan yapılandırmayı siler. Daha fazla bilgi için Bölüm 4'e bakın.
16. C ICMP, oluşturan router'a geri, mesajlar göndermek için kullanılan, Network katmanındaki protokoldür. Daha fazla bilgi için Bölüm 2'ye bakın.
17. A KlassA adreslemesi, host adreslemesi için 24 bit sağlar.
18. B Her 2 saniyede, BPDU'lar, varsayılan olarak, tüm aktif bridge portlarından gönderilmektedir. Daha fazla bilgi için Bölüm 7'ye bakın.
19. D Switch'ler varsayılan olarak VLAN bilgisini göndermezler. VTP domain'i oluşturmanız gerekir. VLAN Trunking Protocol (VTP), bir trunk link boyunca, VLAN bilgisini göndermek için kullanılır. Daha fazla bilgi için Bölüm 9'a bakın.
20. C IEEE 802.11B, 2.4GHz'dir, fakat maksimum sadece 11Mbps'dır. IEEE 802.11G, 54Mbps en yüksek hız değeri ile 2.4GHz'dedir. Daha fazla bilgi için Bölüm 12'ye bakın.
21. A Varsayılan olarak, switch'ler, collision domain'lerini ayırırlar, fakat geniş bir broadcast domain'indedir. Daha fazla bilgi için Bölüm 1'e bakın.

22. D Port Address Translation (PAT), network adres çevrimi için one-to-many yaklaşımına izin verir. Daha fazla bilgi için Bölüm 11'e bakın.
23. C,D VTP, VLAN bilgisinin, trunk bir link boyunca gönderilmesi dışında bir şey yapmadığından, doğru değildir. Bir portta trunk yapılandırmak için, 802.1Q ve ISL kullanılmaktadır. Daha fazla bilgi için Bölüm 9'a bakın.
24. D Stub ağlar, ağ topluluğuna sadece bir bağlantıya sahiptir. Sadece default route'lar, stub bir ağda ayarlanabilir, yoksa network döngüleri oluşabilir. Daha fazla bilgi için Bölüm 7'ye bakın.
25. B Physical katmanda belirtilen hub'lar, elektrik sinyallerini tekrar üretirler. Daha fazla bilgi için Bölüm 1'e bakın.
26. A,C Standart ve extended access list'ler (ACL), bir router'da güvenlik sağlamak için kullanılmaktadır. Daha fazla bilgi için Bölüm 10'a bakın.
27. D `copy flash tftp` komutu, flash'taki mevcut IOS'u, bir TFTP host'una yedek almayı sağlamanıza çalışacaktır. Daha fazla bilgi için Bölüm 5'e bakın.
28. B Bir router'daki konfigürasyonun yedeğini almak için kullanılacak komut, `copy running-config startup-config`'dir. Daha fazla bilgi için Bölüm 5'e bakın.
29. C IOS modelinin geliştirilmesinin birinci sebebi, farklı ağların birlikte çalışabilmesidir. Daha fazla bilgi için Bölüm 1'e bakın.
30. C User Datagram Protocol, Transport katmanındaki bir bağlantı ağ servisidir ve DHCP, bu connectionless servisi kullanır. Daha fazla bilgi için Bölüm 2'ye bakın.
31. E `clock rate` komutu, iki kelimedir ve hattın hızı, bps (bit per second) olarak verilir. Daha fazla bilgi için Bölüm 4'e bakın.
32. C `show ip interface` komutu size, herhangi bir gidiş veya geliş interface'inin, bir access list'e sahip olup olmadığını gösterecektir. Daha fazla bilgi için, Modül 10'a bakın.
33. D `copy tftp flash` komutu, Cisco router'lardaki Cisco IOS için varsayılan lokasyon olan flash bellekte yeni bir dosya yerleştirir. Daha fazla bilgi için, Modül 9'a bakın.

# 1 Ağlar Arası İletişim

# 1 Ağlar Arası İletişim

- Ağlar Arası İletişimin Temelleri
- Ağlar Arası İletişim Modelleri
- OSI Referans Modeli
- Ethernet Ağ Kurulumu
- Ethernet Kablolama
- Data Enkapsülasyonu
- Cisco Üç-Katmanlı Hiyerarşik Modeli
- Özet
- Sınav Gereklilikleri
- Yazılı Lab 1
- Gözden Geçirme Soruları
- Gözden Geçirme Sorularının Cevapları
- Yazılı Lab 1 Cevapları
- Yazılı Lab 1.2 Cevabı
- Yazılı Lab 1.3 Cevaplar
- Yazılı Lab 1.4 Cevaplar

# Ağlar Arası İletişim

Ağlar arası iletişimin heyecan verici dünyasına hoş geldiniz. Bu ilk bölüm, Cisco router ve switch'leri kullanıp, ağların birbirine nasıl bağlandığına odaklanarak, ağlar arası iletişimin temellerini anlamanıza yardımcı olacaktır. Öncelikle, ağlar topluluğunun ne olduğunu tam olarak bilmek zorundasınız, değil mi? Bir router aracılığıyla iki ya da daha fazla LAN ya da WAN'ı bağladığınızda ve IP gibi bir protokol ile mantıksal bir ağ adresleme planı yapılandırdığınızda, bir ağ topluluğu oluşturursunuz.

Bu bölümde şu dört konuyu işleyeceğim:

- Ağlar arası iletişimin temelleri.
- Ağın segmentlere ayrılması.
- Bir ağı fiziksel segmentlere ayırmada, bridge, switch ve router'lar nasıl kullanılmaktadır.
- Bir ağ topluluğu oluşturmak için router'lar nasıl kullanılmaktadır.

Aynı zamanda Open Systems Interconnection (OSI) modelini dikkatle inceleyeceğim ve üzerine kuracağınız ağ bilgisi için güçlü bir temeli gerçekten iyi kavramanız gerektiğinden, her bir bölümünü detaylı bir şekilde size açıklayacağım. OSI modeli; tamamıyla farklı sistemlerin farklı ağlarda güvenli bir şekilde iletişimine imkan vermek için geliştirilen 7 hiyerarşik katmana sahiptir. Bu kitap, CCNA ile ilgili her konuya odaklandığı için sizin OSI modelini Cisco'nun anladığı şekilde anlamanız çok önemlidir. Bu sebeple 7 katmanı size o şekilde sunacağım.

OSI modelinde farklı katmanlar için farklı cihaz türleri olduğundan, bir ağa bu cihazları bağlamak için kullanılacak kablo ve konnektör tiplerini bilmeniz de çok önemlidir. Cisco cihazlarının kablolamasını, (Ethernet teknolojileri ile) router ya da switch'e bağlanmasını, hatta konsol bağlantısıyla bir router ya da switch'e nasıl bağlantı kurulacağını da ele alacağım.

Bölümü ağ topluluklarını tasarlamak, kurmak ve arızalarını gidermek için Cisco tarafından geliştirilmiş üç katmanlı hiyerarşik modeli işleyerek tamamlayacağız.

Bu bölümü okuduktan sonra, 20 adet gözden geçirme sorusu ve dört yazılı laboratuvar uygulamasıyla karşılaşacaksınız. Bunlar size, bu bölümün bilgilerinin hafızanıza yerleşmesi için hazırlanmıştır. Bu nedenle bu bölümleri sakın atlamayın.

NOT *Bu bölüm ile ilgili güncellemeleri bulmak için www.lammle.com ve/veya www.sybex.com adreslerine bakınız.*

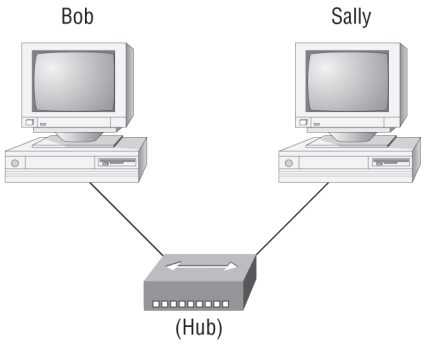
## Ağlar Arası İletişimin Temelleri

Ağlar arası iletişim modellerini ve OSI referans modelinin özelliklerini incelemeden önce, büyük resmi ve anahtar sorunun cevabını anlamamız gerekir. Cisco ağlar arası iletişimi öğrenmek neden bu kadar önemlidir?

Ağlar ve ağ kurma işlemleri, son 15 yıldır çok hızlı bir şekilde büyümektedir. Bunlar, veri ve yazıcı paylaşımı gibi temel vazgeçilmez kullanıcı ihtiyaçlarındaki büyük artışla beraber, video konferans gibi daha ileri talepleri yakalayabilmek için ışık hızında gelişme göstermişlerdir. Ağ kaynaklarını paylaşmaya ihtiyaç duyan herkes aynı ofis ortamında bulunmadığı sürece (gittikçe artan bir durum) yapılması gereken, tüm kullanıcıların bu ağların kaynaklarını kullanabilmeleri için birçok ağı birbirine bağlamaktır.

Şekil 1.1'de bir hub kullanılarak bağlanan basit bir LAN ağının resmini görebilirsiniz. Bu ağ, gerçekte bir collision domain ve bir broadcast domain'dir. Bunların anlamını bilmiyorsanız kaygılanmaya gerek yok, tüm bölüm boyunca birçok kez collision ve broadcast domain'lerden bahsedeceğim. Siz muhtemelen onları rüyanızda bile göreceksiniz.

Şekil 1.1 ile ilgili olarak, Bob isimli PC'nin, Sally isimli PC ile görüştüğünü nasıl söylersiniz? Her ikisi de çok portlu bir repeater (bir hub) ile bağlı aynı LAN'dadır. Bu durumda, Bob "Hey Sally orada mısın?" şeklinde bir veri mesajı gönderebilir mi? Ya da Bob, Sally'nin IP adresini kullanabilir mi? "Hey 192.168.0.3, orada mısın?" gibi veri mesajları kullanabilir mi? Umarım ki, siz IP adresli seçeneği seçtiniz. Fakat böyle olsa dahi, hala haberler kötü, her iki cevap da yanlış! Çünkü Bob gerçekte, ona ulaşmak için Sally'nin PC'sinin ağ kartına yazılı olan MAC adresini (donanım adresi olarak bilinir) kullanacaktır.




Temel ağ, cihazların bilgiyi paylaşmalarına izin verir. Bilgisayar dili terimi, binary koda işaret eder (0'lar ve 1'ler). Yukarıdaki iki kullanıcı, donanım (MAC) adreslerini kullanarak iletişim kurar.

***Şekil 1.1:*** *Temel ağ.*

Güzel, fakat Bob sadece Sally'nin adını biliyor ve onun IP adresini dahi bilmiyorken, Sally'nin MAC adresine nasıl sahip olacak? Bob, genelde Domain Name Service (DNS) kullanılarak gerçekleştirilen isim çözümlemesi (isimden IP adresine çözümleme) ile işe başlayacaktır. Şayet her ikisi de aynı LAN'da ise Bob sadece bu bilgi için Sally'e broadcast gönderebilir (DNS'e gerek yok). (Vista dahil) Microsoft Windows'a hoş geldiniz!

Aşağıda, Bob'dan Sally'e gönderilen basit bir isim çözümlemesinin, bir ağ analizöründen çıktısı var:

```
Time         Source        Destination    Protocol Info
53.892794    192.168.0.2   192.168.0.255  NBNS Name query NB SALLY<00>
```

Daha önce bahsettiğim gibi 2 kullanıcı lokal bir LAN'da bulunduğu sürece Windows (Bob), Sally ismini çözümlemek için bir broadcast gönderecektir (hedef 192.168.0.255, bir broadcast adresidir). Gelin verinin geri kalan kısmına bir bakalım:

```
EthernetII, Src:192.168.0.2 (00:14:22:be:18:3b),
Dst:Broadcast (ff.ff:ff:ff:ff:ff)
```

Bu çıktı gösteriyor ki, Bob kendi MAC adresini ve source (kaynak) IP adresini biliyor ama Sally'nin IP ve MAC adresini bilmiyor. Bu nedenle Bob, tümü ***F***'lerden oluşan bir MAC adresi broadcast'i (bu bir Data Link katmanı yayınıdır) ve 192.168.0.255 olarak bir IP LAN broadcast'i gönderiyor. Tekrar ediyorum, "Subnetting, Variable Lenght Subnet Mask'lar (VLSM) ve TCP/IP Arıza Gidermek" başlıklı bölüm 3'de broadcast'ler hakkında daha ayrıntılı bilgi öğreneceksiniz.

İsim çözümlemesinden önce, Bob'un yapması gereken ilk şey Sally'nin MAC adresini elde etmek için LAN'a broadcast göndermesidir, böylece Sally'nin PC'si ile iletişime geçebilir ve onun ismini IP adresine çevirebilir:

```
Time         Source         Destination   Protocol Info
53.153054    192.168.0.2    Broadcast     ARP Who has 192.168.0.3? Tell
                                                      192.168.0.2
```

Sally'nin cevabını inceleyelim:

```
Time        Source       Destination Protocol Info
5.153403    192.168.0.3  192.168.0.2  ARP 192.168.0.3 is at 00:0b:
                                                    db:99:d3:5e

5.53.89317 192.168.0.3   192.168.0.2  NBNS Name query response NB
                                                    192.168.0.3
```

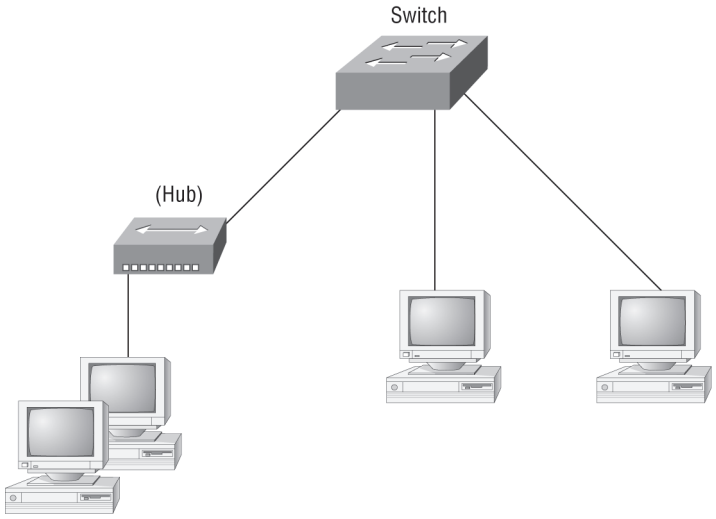
Tamam, Bob şimdi Sally'nin hem IP hem de MAC adresine sahip! Bu noktada her ikisi de source (kaynak) adresi olarak listelenmiştir, çünkü bu bilgi Bob'a Sally tarafından gönderilmiştir. Son olarak, Bob, Sally ile bağlantıya geçmek için ihtiyacı olan her şeye sahiptir. Size ARP'dan söz edeceğim ve "IP Routing" başlıklı bölüm 6'da, Sally'nin IP adresinin bir MAC adresine nasıl çözümlendiğini tam olarak göstereceğim.

Bu arada, Sally'nin Bob'la tekrar iletişime geçmek için aynı çözümleme işlemini tekrarlamak zorunda olduğunu anlamanızı istiyorum, çılgınca geliyor değil mi? Bunu, Windows ile basit ağ kurma ve IPv4'e giriş olarak düşünün (henüz router eklemedik bile).

İşleri biraz daha zorlaştırmak için bazı durumlarda geniş bir ağı, kullanıcı tepki süresini azaltmak için daha küçük birkaç ağa bölmek zorunda kalacaksınız. Çünkü ağ büyüdükçe ağırlaşacaktır. Bütün bu büyümeyle beraber, LAN'daki tıkanıklık çok yüksek seviyelere çıkacaktır. Bunun çözümü, network segmentation denen gerçekten büyük olan ağların daha küçük ağlara bölünmesidir. Bunu router, switch ve bridge gibi cihazlar kullanarak yaparsınız. Şekil 1.2 switch ile bölünmüş bir ağı göstermektedir. Böylece switch'e bağlı her bir ağ şimdi ayrı bir collision domain'idir. Fakat dikkat edilmesi gerekir ki bu ağ hala bir broadcast domain'den oluşmaktadır.

Şekil 1.2'de kullanılan hub'ın, switch porttaki bir collision domain'i genişlettiğini unutmayın. LAN'daki trafik tıkanıklığının genel sebepleri aşağıda listelenmiştir:

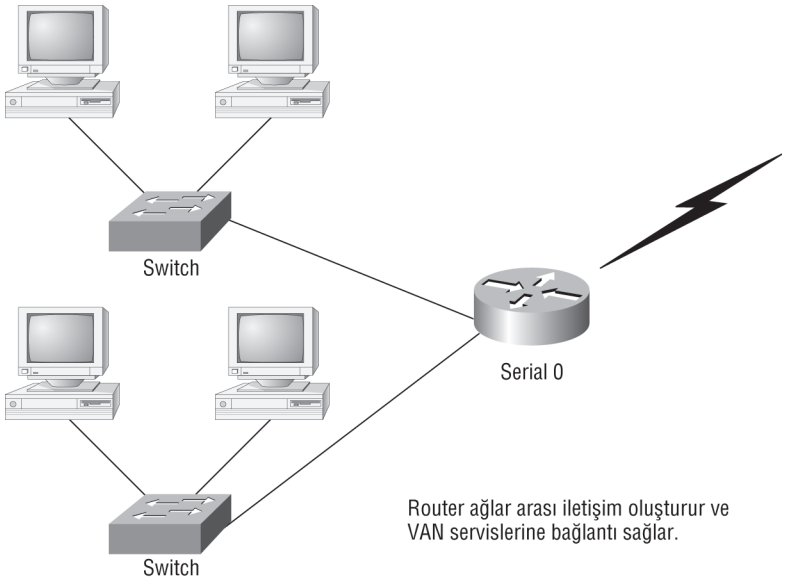
- Bir broadcast domain'de çok sayıda kullanıcı olması.
- Broadcast fırtınası.
- Multicasting.
- Düşük bant genişliği.
- Ağa bağlantı için hub eklenmesi.
- Yoğun bir ARP veya IPX trafiği (IPX, IP'ye benzer bir Novell protokolüdür. Günümüz ağlarında kullanılmaz.)




*Şekil 1.2: Collision domain'leri ayırmak için hub yerine switch kullanılabilir.*

Şekil 1.2'ye tekrar bakalım. Şekil 1.1'deki ana hub'ın yerine bir switch yerleştirdiğimi fark ettiniz mi? Fark ettiniz ya da etmediniz, bunu yapmamım sebebi; hub'ların ağları segmentlere ayırmaması, network segmentlerini sadece bağlamasıdır. Yani basit olarak, ev kullanımı ve arıza gidermek ve birkaç PC'yi birbirine bağlamak için kolay ve pahalı olmayan bir yoldur, ama hepsi bu kadar.

Günümüzde router'lar, ağları birbirine bağlamak için ve veri paketlerinin bir ağdan diğerine yönlendirilmesinde kullanılır. Cisco kaliteli router ürünleri ve mükemmel servisi ile router'lar için fiili standart olmuştur. Varsayılan olarak router'lar, segmente gönderilen tüm broadcast'leri işiten bir ağ segmentindeki tüm cihazların bütünü olan broadcast domain'lerini ayırır. Şekil 1.3, küçük ağımızda ağlar topluluğu oluşturan ve broadcast domain'leri ayıran bir router göstermektedir.




***Şekil 1.3:*** *Router'lar bir ağ topluluğu oluşturur.*

Şekil 1.3'teki ağ oldukça güzeldir. Her kullanıcı, kendi collision domain'ine bağlıdır ve router, iki broadcast domain oluşturur. Router'ların, WAN servislerine bağlantı sağladıklarını da unutmayın. Router'lar, Cisco cihazlarındaki V.35 gibi WAN bağlantıları için serial interface'leri kullanır.

Broadcast domain'i ayırmak önemlidir, çünkü bir kullanıcı ve sunucu, ağa bir broadcast gönderdiğinde eğer router yoksa ağdaki tüm cihazlar, bunu almak ve işleme koymak zorundadır. Router'ın interface'i bir broadcast aldığında bunu diğer ağlara iletmeksizin, "Teşekkür ederim, fakat almayayım." diyerek reddeder. Router'lar, varsayılan olarak, broadcast domain'lerini ayıran cihazlar olarak bilindiği halde, collision domain'lerini de ayırdıklarını hatırlamak önemlidir.

Ağınızda router kullanmanın iki avantajı vardır:

- Varsayılan olarak, broadcast'leri yönlendirmezler.
- IP adresi gibi 3. katman (ağ katmanı) bilgilerinin olduğu paketleri filtreleyebilirler.

Ağınızdaki dört router fonksiyonu aşağıdaki gibidir:

- Paket switching
- Paket filtreleme
- Ağlar arası iletişim
- Yol seçimi

Router'ların gerçekte switch olduklarını hatırlayın: Onlar gerçekten 3. katman switch'leridir (katmanlar hakkında bu bölümde bahsedeceğiz). Frame'leri ileten ya da filtreleyen 2. katman switch'lerin aksine router'lar (3.katman switch'ler) mantıksal adresleme kullanır ve paket switching sağlarlar. Router'lar, aynı zamanda access list'leri kullanarak, paket filtreleme sağlayabilir. İki ya da daha fazla ağı birbirine bağladıklarında mantıksal adresleme (IP veya IPv6) kullanırlar. Bu oluşumun adı ağlar topluluğudur. Son olarak router'lar, yol seçimi ve uzak ağlara paketleri yönlendirmek için bir routing tablosu (ağlar topluluğunun haritasını) kullanır.

Tam tersine switch'ler, ağlar topluluğu oluşturmak için kullanılmazlar (varsayılan olarak broadcast domain'leri oluşturmazlar); bir ağa işlevsellik katmak için görevlendirilirler. Bir switch'in ana görevi, LAN kullanıcılarına daha yüksek bant genişliği sağlayıp performanslarını artırarak, bir LAN'ın daha iyi çalışmasını sağlamaktır. Switch'ler, router'ların yaptığı gibi paketleri diğer ağlara yönlendirmez. Onun yerine, paketleri bir porttan diğerine anahtarlama yaparak iletirler. Burada

paket ve frame'in ne olduğunu sorabilirsiniz. Onlar hakkında, bu bölümde size daha sonra bilgi vereceğim.

Varsayılan olarak, switch'ler, collision domain'leri ayırır. Bu; belirli bir cihazın, bir ağ segmentine paket gönderdiği ve aynı segmentteki diğer tüm cihazların dikkatini ona çekmeye zorladığı bir ağ senaryosunu açıklamak için kullanılan bir Ethernet terimidir. Ağdaki iki cihazın aynı anda paket yollaması, collision'a sebep olur. Her seferinde birisi olmak koşuluyla, her iki cihaz da tekrar aktarım yapmak zorundadır. Bu pek randımanlı değildir. Bu durum, tipik olarak tek bir collision ve broadcast domain'inin olduğu, her kullanıcı segmentinin hub'a bağlı olduğu ortamda gerçekleşir. Hub'ın tersine, switch'in her bir portu, kendi collision domain'ini oluşturur.

*Switch'ler ayrı collision ve tek bir broadcast domain'i oluşturur. Router'lar, her bir interface için ayrı bir broadcast domain oluşturur.*

**NOT**

Bridging terimi, router ve hub'ların geliştirilmesinden önce ortaya çıkmıştır. Bu sebeple, switch'ler kadar bridge'leri de anlatmak için sıkça kullanılır. Bunun sebebi, bridge ve switch'lerin bir ağdaki collision domain'lerini ayırmak gibi, temelde aynı işi yapıyor olmalarıdır. (Gerçekte, günümüzde bir bridge cihazı alamazsınız, sadece LAN switch'leri alabilirsiniz. Switch'ler, bridging teknolojisi kullandığından Cisco hala onları çok portlu bridge olarak tarif eder.)

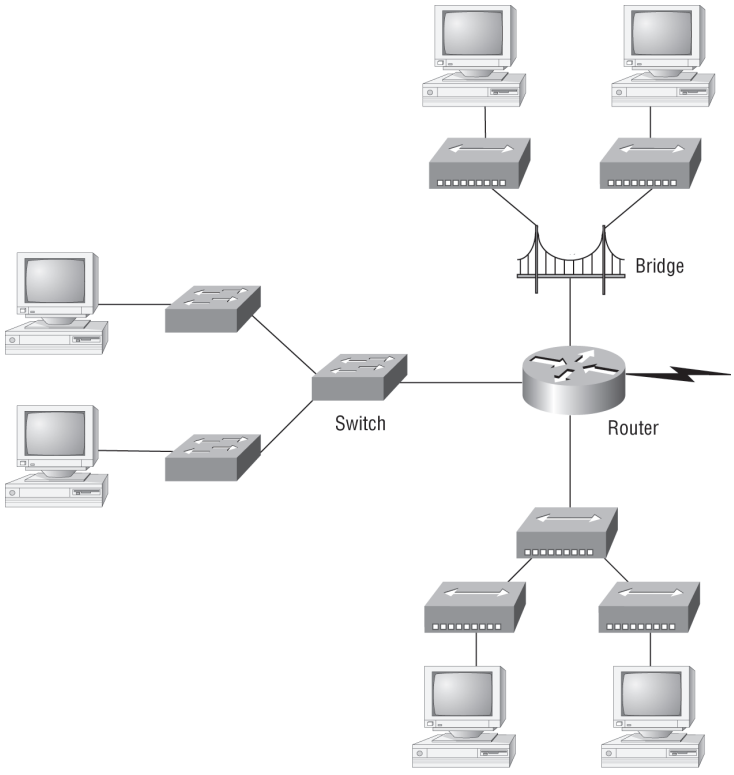
Yani basit olarak, switch için daha akıllı ve çok portlu bir bridge diyebiliriz. Bu tanım doğrudur, fakat bazı farklılıklar vardır. Switch'ler bu fonksiyonu, çok gelişmiş yönetim kabiliyeti ve özellikleri ile sağlar. Artı, çoğu zaman bridge'ler 2 ya da 4 portludur. Tamam, onları 16 porta kadar artırabilirsiniz, fakat bazı switch'lerdeki yüzlerce portla mukayese bile edilemezler.

*Bir bridge'i broadcast domain'deki collision'ı azaltmak ve ağınızdaki collision domain sayısını artırmak için kullanabilirsiniz. Ayrıca, hub'ın ethernet ağında, tıkanıklığa neden olabileceği aklınızda bulunsun. Ağ tasarımı yaparken daima dikkatli olun.*

**NOT**

Şekil 1.4, tüm ağ topluluğu cihazlarıyla, bir ağı göstermektedir. Router'ın, her bir LAN interface'i için sadece broadcast domain'lerini değil, collision domain'lerini de ayırdığını anımsayın.

Şekil 1.4'e baktığınızda router'ın merkezi bir konumda olduğunu ve tüm fiziksel ağlara bağlandığını fark ettiniz mi? Bu yerleşim planını, bridge ve hub gibi eski teknolojileri de içerdiği için kullanmak zorundayız.




***Şekil 1.4:*** *Ağlar arası iletişim cihazları.*

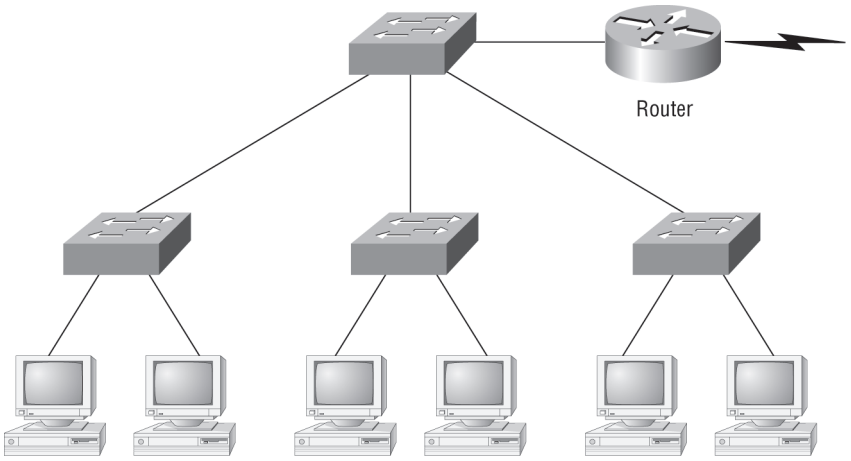
Şekil 1.4'ün üst tarafındaki ağda, hub'ları router'a bağlamak için bir bridge kullanıldığına dikkat edin. Bridge, collision domain'lerini ayırmaktadır, fakat her iki hub'a bağlı kullanıcılar aynı broadcast domain'ine sıkışmıştır. Aynı zamanda, bridge sadece iki collision domain oluşturmuştur, bu nedenle hub'a bağlı her bir cihaz, aynı hub'a bağlı diğer tüm cihazlar gibi aynı collision domain'indedir. Gerçekte bu oldukça sakıncalı bir durumdur, fakat hala bütün kullanıcıların aynı collision domain'de olmasından daha iyidir.

NOT *Bridge/Switch'ler ağları segmentlere bölmek için kullanılmasına rağmen, broadcast veya multicast paketlerini ayıramaz.*

Başka bir konuya dikkat edelim: Alttaki, birbirine ve aynı zamanda router'a bağlı 3 hub, bir collision ve broadcast domain oluşturmaktadır. Bu durum, bridge ile bağlanmış ağın daha iyi olduğunu gösterir!

Router'a bağlı en iyi ağ, soldaki switch kullanılan LAN ağıdır. Neden? Çünkü switch'teki her port collision domain'leri oluşturur. Fakat tüm cihazlar hala aynı broadcast domain'indedir. Bunun gerçekten neden kötü bir şey olduğunu hatırladınız mı? Çünkü tüm cihazlar, yayınlanan tüm broadcast'leri dinlemek zorundadır. Ve broadcast domain'iniz fazla genişse, kullanıcılar daha az bant genişliğine sahip olacak, daha fazla broadcast için daha çok işlem gücü gerekecek ve ağın tepkime süresi, ofiste karmaşa oluşturacak seviyeye düşecektir.

Ağımızda sadece switch'ler bulunduğu zaman, her şey çok değişecektir. Şekil 1.5, tipik olarak günümüzde kullanılan ağı göstermektedir.




***Şekil 1.5:*** *Bir ağlar topluluğu oluşturmak için switch kullanılmış ağ.*

Tamam, LAN switch'leri, ağların merkezine yerleştirdim, şimdi router'lar sadece mantıksal ağları birbirine bağlamaktadır. Bu tür bir kurulum yaparsam, Virtual LAN'lar (VLAN'ler) oluşturmuş olurum, bu konu "Virtual LAN'lar (VLAN'lar)" başlıklı bölüm 9'da ele alınacaktır. Endişe etmenize gerek yok, fakat bir switch kullanılmış ağa sahip olsanız bile, VLAN'lar arası ya da ağlar arası iletişim sağlamak için hala bir router'a ihtiyacınız olduğunu anlamanız gerçekten çok önemlidir. Bunu sakın unutmayın!

Belli ki en iyi ağ, hizmet ettiği firmanın iş gereksinimlerini karşılayacak şekilde doğru tasarlanmış olan ağdır. Ağda doğru yerleştirilmiş router'larla LAN switch'ler, en iyi ağ tasarımını sağlar. Bu kitap, router ve switch'lerin temellerini anlamanıza yardımcı olacaktır, böylece olaylara göre güçlü ve bilgili kararlar verebileceksiniz.

Şekil 1.4'e tekrar dönelim. Bu ağlar topluluğunda kaç adet collision ve broadcast domain'i vardır? Umarım, dokuz collision ve üç broadcast domain'i olarak cevap verdiniz. Broadcast domain'lerini görmek kesinlikle en kolay olanıdır, çünkü varsayılan olarak router'lar, broadcast domain'leri oluşturur. Üç bağlantı olduğundan, bu size üç broadcast domain verir. Fakat dokuz collision domain'ini görebildiniz mi? Hayır cevabına karşın açıklayacağım. Tüm hub ağları bir collision domain'idir, bridge ağı üç collision domain'ine eşittir. Switch'in her bir portu için bir tane olmak üzere, switch ağındaki beş collision domain'ini de eklersek, toplam dokuz adet vardır.

Şimdi, Şekil 1.5'de, switch'teki her port, ayrı bir collision domain'i ve her VLAN ayrı bir broadcast domain'idir. Fakat VLAN'lar arasında routing için hala bir router'a ihtiyacınız var. Burada kaç tane collision domain'i görüyorsunuz? Ben 10 tane sayıyorum. Switch'ler arasındaki bağlantıların da bir collision domain olarak kabul edildiğini hatırlayın!

### Gerçek Dünya Senaryosu

### Tüm Hub'larımı Switch'lerle değiştirmeli miyim?

*San Jose'daki büyük bir şirkette ağ yöneticisisiniz. Patron size geliyor ve bir switch alma isteğiniz için ne kadar bir gideri onaylayacağını bilmediğini ve gerçekten ihtiyaç olup olmadığını soruyor.*

*Switch'ler, gerçekten, hub'ların sahip olmadığı birçok özelliği ağa eklemektedir. Fakat hiçbirimiz sınırsız bir bütçeye sahip değiliz. Hub'lar, hala güzel bir ağ oluşturabilmektedir. Bu, tabi ki, ağı doğru bir şekilde tasarlayıp kurmanıza bağlıdır.*

*40 kullanıcınızın, dört hub'a bağlandığını ve her birinde 10 kişi olduğunu düşünelim. Bu noktada, herkes hub'larla bağlandığından, geniş bir collision domain ve geniş bir broadcast domain'e sahipsiniz. Şayet sadece bir tane switch alabiliyorsanız ve hem hub'ları hem de sunucuları switch portlarına bağlarsanız, o zaman dört collision domain'e ve bir broadcast domain'e sahip olursunuz. Mükemmel değil, fakat bir switch'in maliyetiyle, ağınız daha iyi olacaktır. O halde, siz yine de tamamen yeni switch'ler almak konusunda talepte bulunun. Ne kaybedersiniz?*

Ağlar arası iletişim ve bir ağ topluluğunda bulunan çeşitli cihazlarla tanıştınız. Ağlar arası iletişim modellerine başlama zamanı geldi.

# Ağlar Arası İletişim Modelleri

Ağ çalışmaya başladığında bilgisayarlar tipik olarak aynı üreticiden bilgisayarlarla iletişim kurar. Örneğin firmalar, ya komple bir DECnet çözümü ya da bir IBM çözümü çalıştırır, ikisi bir arada olmaz. 1970'lerin sonlarında Open Systems Interconnection (OSI) referans modeli, bu sınırlamayı kaldırmak için International Organization for Standardization (ISO) tarafından oluşturuldu.

OSI modeli, farklı üretici ağlarının, diğerleriyle çalışabilmesi için üreticilerin, ortak çalışan ağ cihazı ve protokollere benzer yazılım oluşturmasına yardım etmesi anlamına gelmekteydi. Aynı dünya barışı gibi, muhtemelen asla tamamıyla gerçekleşmeyecektir, fakat hala çok büyük bir hedeftir.

OSI modeli, ağlar için öncelikli bir mimari modeldir. OSI, bir bilgisayardaki bir uygulamadan, ağ ortamı aracı boyunca, başka bilgisayardaki bir uygulamaya veri ve ağ bilgisinin nasıl aktarılacağını açıklar. OSI referans modeli, bu yaklaşımı katmanlara ayırır.

Şimdiki bölümde, katmanlı yaklaşımı ve bu yaklaşımın, ağ topluluklarımızda hata tespitinde bize yardımcı olması için nasıl kullanabileceğini açıklayacağım.

## Katmanlı Yaklaşım

Bir referans modeli, haberleşmenin nasıl olması gerektiğinin kavramsal bir tasarımıdır. Verimli haberleşme için gerekli tüm işlemleri adresler ve bu işlemleri, katman denilen mantıksal gruplara böler. Bir iletişim sistemi, bu yolla tasarlandığında, katmanlı yapı olarak bilinir.

Şu şekilde düşünün: Siz ve bazı arkadaşlarınız bir firmada işe başlamak istiyorsunuz. Yapacağınız ilk işlerden biri, oturmak ve hangi işlemlerin yapılmak zorunda olduğunu, onları kimin yapacağını, yapılacağı sırayı ve birbirleriyle nasıl ilişkilendirileceğini düşünmektir. Eninde sonunda, bu işleri bölümlere ayırabilirsiniz. Diyelim ki, bir sipariş-alma ve bir stok ile bir nakliye bölümüne sahip olmaya karar verdiniz. Departmanlarınızın hepsinin, kendi çalışanlarını meşgul eden ve sadece kendi görevlerine odaklanmalarını gerektiren, işleri vardır.

Bu senaryoda bölümleri, bir iletişim sistemindeki katmanlara benzeteceğim. İşlerin ilerlemesini kolaylaştırmak için her bölümün elemanı diğerlerine güvenmeli, işlerini yapmalarını sağlamak için ağırlıklarını koymalı ve kendi sorumluluklarını profesyonelce ele almalıdır. Planlama toplantılarınızda notlar alırsınız, iş planınız veya referans modeliniz gibi operasyon standartları hakkında sonraki tartışmalara yardım etmesi için tüm işleyişi kaydedersiniz.

İşe başladığınızda, kendileri ile ilgili projelere sahip bölüm müdürleri, tanımlı görevleri yerine getirmek için pratik yöntemler geliştirmeye ihtiyaç duyacaktır. Bu pratik yöntem veya protokollerin, standart bir çalışma prosedür kılavuzu olarak birleştirilmesine ve yakından takip edilmesine gerek duyulacaktır. El kitabınızdaki çeşitli prosedürlerin her biri, farklı gerekçeler içerecek, farklı önem derecelerine ve kurulumlara sahip olacaktır. Şayet bir ortaklık oluşturacak ya da başka bir firmayı alacaksanız, iş protokollerinin (iş planlarının) sizinkilerle uygun (en azından uyumlu) olması gerekecektir.

Benzer şekilde, yazılım geliştiriciler, bilgisayar iletişim prosedürlerini anlamak için bile, referans modeli kullanabilirler ve herhangi bir katmanda hangi tür fonksiyonlara ihtiyaç duyulacağını görebilirler. Şayet belirli bir katman için protokol geliştiriyorlarsa, ilgilenmeleri gereken diğer katmanlar değil, bu belirli katmandır. Diğer katman ve protokoller, diğer fonksiyonları ele alacaktır. Bu fikir için kullanılan teknik terim, binding'dir. Birbirleriyle ilişkili haberleşme protokolleri, belirli bir katmanda birleştirilmiş veya gruplanmıştır.

## Referans Modellerinin Avantajları

OSI modeli hiyerarşiktir. Aynı fayda ve avantajlar, herhangi bir katmanlı modele uygulanabilir. Bu tür modellerin hepsinin, özellikle de OSI modelinin asıl amacı, farklı üretici ağlarının birlikte çalışmalarına izin vermektir.

OSI katmanlı modeli kullanmanın avantajları, aşağıda belirtilmiştir fakat bunlarla sınırlı değildir:

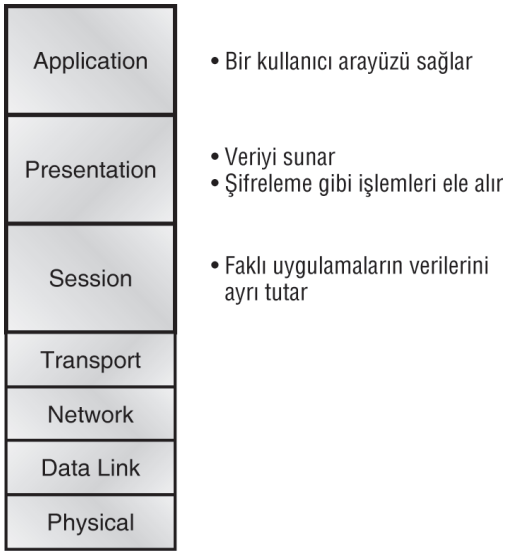
- Ağ iletişim işlemlerini daha küçük ve basit bileşenlere böler, böylece geliştirme, planlama ve hata gidermeye yardımcı olur.
- Ağ bileşenlerinin standartlaştırılmasıyla, çoklu-üretici gelişimine izin verir.
- Modelin her katmanında hangi fonksiyonların olduğunu açıklayarak, endüstri standartlaşmasına cesaret verir.
- İletişim için farklı ağ donanım ve yazılım çeşitlerine izin verir.
- Bir katmanın diğer katmanı etkilemesini engeller, böylece gelişimi engellemez.

# OSI Referans Modeli

OSI düzenlemelerinin en iyi fonksiyonlarından biri, tamamen farklı kullanıcı makineleri arasında veri transferine yardımcı olmasıdır. Yani, örneğin bize bir Unix host'u ve bir PC veya bir Mac arasında veri transferi yapmamıza izin verirler.

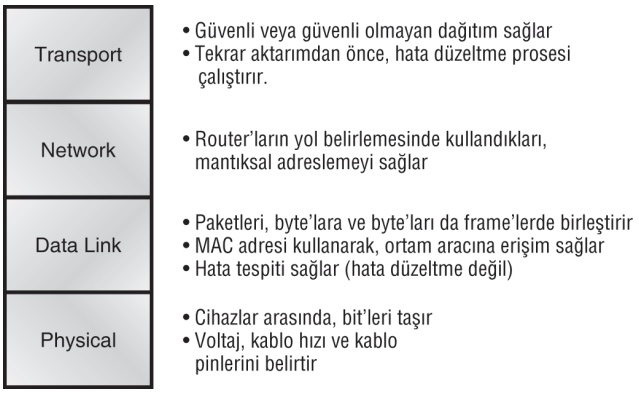
Buna rağmen, OSI fiziksel bir model değildir. Daha çok uygulama geliştiricilerin, bir ağda çalışan uygulamaları oluşturmak ve tamamlamak için kullanabildikleri bir kurallar bütünüdür. Ayrıca, ağ kurulumu standartları, cihazlar ve ağlar arası iletişim planları oluşturmak ve tamamlamak için bir iskelet oluşturur.

OSI, iki gruba ayrılmış, yedi katmana sahiptir. Üstteki üç katman, uç istasyonlardaki uygulamaların birbirleri ve kullanıcılar ile nasıl iletişim kuracaklarını açıklar. Alttaki dört katman, verinin uçtan uca nasıl aktarılacağını açıklar. Şekil 1.6, üstteki üç katmanı ve fonksiyonlarını gösterir. Şekil 1.7 dört alt katman ve fonksiyonlarını gösterir.




***Şekil 1.6:*** *Üst katmanlar.*

Şekil 1.6'yı çalıştığınızda, Application katmanında bilgisayarlar ile kullanıcı arayüzlerini ve ayrıca, üst katmanların, kullanıcı makineleri arasında uygulamaların iletişiminden sorumlu olduğunu anlarsınız. Üst katmanların hiçbirinin, ağ kurulumu ve ağ adresleri hakkında bir şey bilmediklerini hatırlayın. Bunlar alttaki dört katmanın sorumluluğundadır.




***Şekil 1.7:*** *Alt katmanlar.*

Şekil 1.7'de, switch ve router'lar yardımıyla veya fiziksel bir kablo üzerinden verinin nasıl transfer edildiğini açıklayan, alttaki dört katmanın çalışmasını görebilirsiniz. Bu alt katmanlar ayrıca, kaynak host'tan bir veri akışının, hedef host'un uygulamasında tekrar nasıl oluşturulacağını belirler.

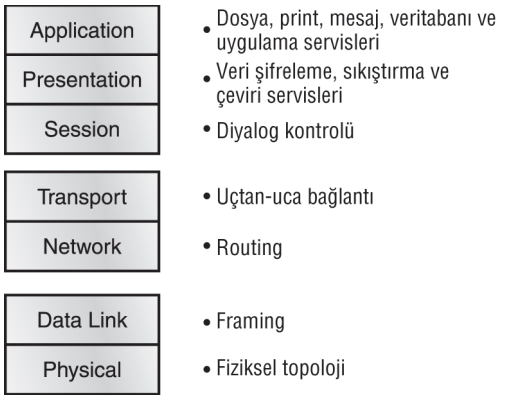
Aşağıdaki ağ cihazları, OSI modelinin tüm 7 katmanını da kullanır:

- Network management station (NMS) 'ler
- Web ve uygulama sunucuları
- Gateway'ler (varsayılan ağ geçitleri değil)
- Ağ host'ları

Basit olarak OSI, ağ protokol dünyasının Emly POST'udur. Tam Bayan Post'un, insan sosyal etkileşimi için standartlar ya da protokoller oluşumu için yazdığı gibi OSI, emsalleri ve açık ağ protokolü serisi için kılavuz ve emsal olarak, OSI referans modelini geliştirdi. İletişim modellerinin etiğini açıklayarak, bugün protokol aileleri için, en popüler kıyaslama yöntemi olarak kalmıştır.

OSI referans modeli yedi katmana sahiptir:

- Application katmanı (katman 7)
- Presentation katmanı (katman 6)
- Session katmanı (katman 5)
- Transport katmanı (katman 4)
- Network katmanı (katman 3)
- Data Link katmanı (katman 2)
- Physical katmanı (katman 1)




***Şekil 1.8:*** *Katman fonksiyonları.*

Şekil 1.8, OSI modelinin her katmanında tanımlanan fonksiyonların bir özetini gösterir. Bu bilgilerle, her katmanın fonksiyonlarını detaylı olarak inceleyebiliriz.

## Application Katmanı

OSI'nin application katmanı, kullanıcıların gerçekte bilgisayarla iletişime geçtiği yeri belirtir. Bu katman sadece kısa bir süre içinde ağa erişimin olması gerektiğinde devreye girer. TCP/IP, NIC kartı gibidir. Sistemden ağ kurulumu bileşenlerinin tümünü kaldırabilirsiniz ve hala yerel HTML dokümanlarına göz atmak için IE kullanabilirsiniz. Fakat HTTP kullanarak alınması gereken bir HTML dokümanına göz atmak veya FTP ya da TFTP ile bir dosyayı indirmek gibi şeyler yapmaya çalışırsanız, işler karmaşıklaşacaktır. Bu sebeple IE, application katmanına erişime çalışarak bu gibi isteklere cevap verecektir. Application katmanı, katmanlı yapının her bölümünde olmayan, gerçek uygulama programları ile protokol yığını yardımıyla bilgi göndermek için uygulamalara

yollar sağlayan alttaki katman arasında bir arayüz gibi davranmaktadır. Başka bir deyişle, IE gerçekte application katmanında yer almaz, uzak kaynaklarla temasa geçmeye ihtiyacı olduğunda Application katmanıyla çalışır.

Application katmanı ayrıca, uygun iletişim partnerinin belirlenmesi ve kurulumu ile uygun iletişim için yeterli kaynağının mevcut olup olmadığını saptamaktan da sorumludur.

Bazen bilgisayar uygulamalarının sadece masaüstü kaynaklarından daha fazlasını gerektirmesinden dolayı, bu görevler önemlidir. Sık sık bir ağ uygulamasından daha fazla iletişim bileşenini bir araya getirecektir. Önemli örnekler, dosya transferleri ve e-posta'dır. Bu uygulamalar uzak erişimi, ağ yönetim aktivitelerini ve istemci/sunucu işlemlerini gerektirecektir. Birçok ağ uygulaması şirket ağları üzerindeki iletişim için servisler sağlar, fakat şimdiki ve gelecekteki ağlar arası iletişim için ihtiyaç, mevcut fiziksel ağ kurulumu sınırlarının ötesine ulaşmak için hızla gelişmektedir.

NOT *Application katmanının, gerçek uygulama programları arasında bir arayüz gibi davrandığını hatırlamak önemlidir. Yani, örneğin Microsoft Word, Application katmanında olmayacaktır, fakat Application katman protokolleri ile çalışacaktır. Bölüm 2, FTP ve TFTP gibi Application katmanında bulunan bazı programları gösterecektir.*

## Presentation Katmanı

Presentation katmanı, ismini amacından alır. Application katmanına veri sunar ve veri çevirisi ile kod formatlamaktan sorumludur.

Bu katman aslında bir çeviricidir ve kodlama ile çevirme fonksiyonları sağlar. Başarılı bir veri transfer tekniği, aktarımdan önce veriyi standart bir şekle sokar. Bilgisayarlar, bu formatlanmış veriyi almak ve sonra da doğru okuma (örneğin EBCDIC'ı ASCII'ye) için veriyi kendi orijinal formatına tekrar çevirmek için yapılandırılırlar. Çeviri servisi sağlayarak, Presentation katmanı, bir sistemin Application katmanından aktarılan verinin, karşı tarafın Application katmanı tarafından okunabilmesini garanti eder.

OSI, standart verinin nasıl formatlanması gerektiğini tanımlayan protokol standartlarına sahiptir. Veri sıkıştırma, kriptolama gibi görevler bu katmanla ilgilidir. Bazı presentation katman standartları, multimedia uygulamalarına da karışır.

## Session Katmanı

Session katmanı, Presentation katmanı tarafları arasındaki oturumları kurmak, yönetmek ve sonlandırmaktan sorumludur. Bu katman, cihazlar veya düğümler arasında diyalog kontrolü de sağlar. Üç farklı mod (simplex, half duplex ve full duplex) önererek, sistem ve hizmetler arasındaki iletişimi koordine ederek, düzenler. Özet olarak, Session katmanı basitçe, farklı uygulama verilerinin diğer uygulamaların verilerinden ayrılmasını sağlar.

## Transport Katmanı

Transport katmanı, veriyi, bir data akışına segmentler ve tekrar bir araya getirir. Transport katmanında bulunan servisler, üst-katman uygulamalarından gelen verileri böler, tekrar bir araya getirir ve onu aynı veri akışında birleştirir. Uçtan-uca veri aktarım servisleri sağlar ve bir ağ topluluğunda gönderici ve hedef arasında mantıksal bir bağlantı kurabilir.

Bazılarınıza TCP ve UDP tanıdık gelebilir (gelmiyorsa, bölüm 2'de bundan bahsedeceğim). Şayet öyleyse ikisinin Transport katmanında çalıştığını, TCP'nin güvenilir, UDP'nin ise güvenilir olmayan bir servis olduğunu bilirsiniz. Yani, uygulama geliştiriciler TCP/IP ile çalışırken, iki protokol arasında bir tercihe sahiptir.

Transport katmanı, üst-katman uygulamalarının çoklanması (multiplexing), oturumların oluşturulması ve sanal devrelerin kapatılması için mekanizmalar sağlamaktan sorumludur. Ayrıca, transparan veri trafiği sağlayarak network ile ilgili bilgilerin detaylarını üst katmanlardan gizler.

Transport katmanı, connectionless ya da connection-oriented olabilir. Bununla beraber, Cisco Transport katmanının connection-oriented bölümünü anlamanızla ilgilenir. Sonraki bölümler, Transport katmanının connection-oriented (güvenli) kısmında kolaylık sağlayacaktır.

*Güvenli ağ deyimi, transport katmanında kullanılabilir. Bu, acknowledgment, sequencing ve akış kontrolü kullanılacak anlamına gelir.* **NOT**

## Akış Kontrolü

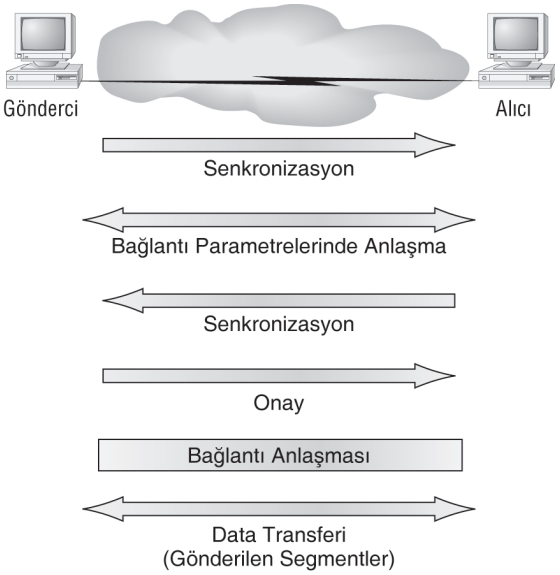
Veri bütünlüğünden, Transport katmanında, akış kontrolü yapılarak ve kullanıcılara, sistemler arasında güvenli veri aktarımı istemelerine izin vererek emin olunur. Akış kontrolü, bağlantının bir tarafındaki göndericiyi, alıcıdaki arabelleklerin aşırı yüklenmesinden korur (veri kaybına yol açabilecek bir olaydan). Güvenli veri aktarımı, sistemler arasında, connection-oriented iletişim oturumları çalıştırır ve ilgili protokoller, aşağıdakilerin yapılldığından emin olurlar:

- Taşınan segmentler, alındıklarında, gönderici onaylanır.
- Onaylanmayan bir segment tekrar aktarılmaz.
- Segmentler, hedeflerine varır varmaz tekrar düzgün sırasına dizilirler.
- Tıkanıklık, aşırı yükleme ve veri kaybından kaçınmak için, yönetilebilir bir veri aktarımı sağlanmaktadır.

*Akış kontrolünün amacı, gönderici tarafından yollanan veri miktarını yönetmek için alıcıya imkan sağlamaktır.* **NOT**

## Connection-Oriented İletişim

Güvenli bir aktarım işleminde gönderim yapmak isteyen bir makine, bir oturum oluşturarak, uzak bir makineyle connection-oriented bir iletişim kurar. Aktarıcı cihaz, ilk olarak karşı sistemle, bir call setup veya three-way handshake olarak bilinen bir connection-oriented oturum kurar. Bundan sonra veri aktarılır, aktarım tamamlandığında,sanal devre sonlandırılır.




***Şekil 1.9:*** *Connection-oriented oturum kurulması.*

Şekil 1.9, alıcı ve gönderici sistemler arasında olan tipik bir güvenli oturumu göstermektedir. Şekle bakarak, her iki host'un uygulama programlarının, kendi işletim sistemlerini, bir bağlantının nerdeyse başlayacağı konusunda uyarmaya başladığını görebilirsiniz. İki işletim sistemi, aktarımın onaylandığı ve iki tarafın hazır olduğunun doğrulandığını ağ üzerinden mesajlar göndererek, iletişim kurarlar. Tüm bu gerekli senkronizasyon gerçekleştikten sonra, bir bağlantı tamamıyla kurulmuş ve veri transferi başlamış olur. (Bu sanal devre kurulumu, ek yük olarak belirtilir!)

Kullanıcılar arasında bilgi aktarılıyorken, iki makine de her şeyin iyi gittiğinden ve verinin düzgün şekilde alındığından emin olmak için protokol yazılımları yardımıyla iletişime geçerek, periyodik olarak birbirlerini kontrol eder.

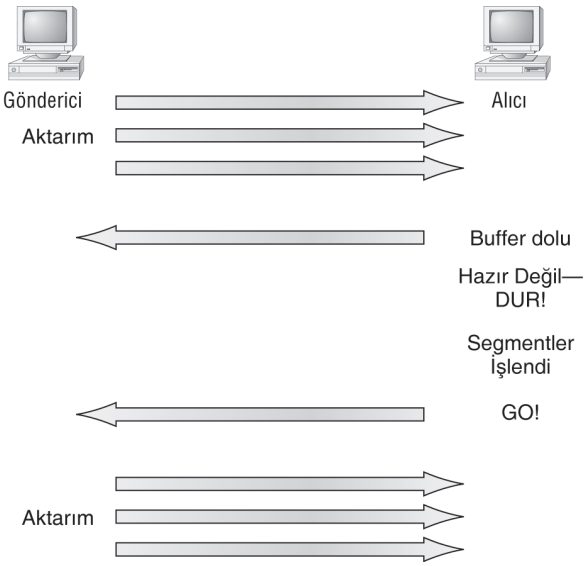
Şekil 1.9'da resimlenen, connection-oriented oturumun, Three-way handshake evrelerini özetleyeyim.

- İlk bağlantı anlaşması segmenti, senkronizasyon için bir istektir.
- İkinci ve üçüncü segmentler, isteği onaylar ve kullanıcılar arasında bağlantı parametreleri (kuralları) belirler. Bu segmentler, hem alıcı sıralamasını burada senkronize etmek hem de çift yönlü bir bağlantı şekillendirmek ister.
- Son segment, yine bir onaylamadır. Hedef host'u, bağlantı anlaşmasının kabul edildiği ve gerçek bağlantının kurulduğu konusunda uyarır. Veri transferi artık başlayabilir.

Her şey basit gibi görünüyor, fakat her zaman böyle problemsiz gerçekleşmez. Bazen hızlı bir bilgisayarın ağın aktarımında daha hızlı veri trafiği oluşturması sebebiyle, transfer sırasında tıkanıklık olur. Tek bir gateway veya hedef boyunca eşzamanlı datagram gönderen bir grup bilgisayar, her şeyi kolayca berbat edebilir. Bir diğer durum, tek bir kaynak probleme sebep olmadığı halde, bir gateway ya da hedef aşırı yoğun olabilir. Her iki durumda da, problem basit olarak, çok küçük bir kapasite için çok fazla trafiğin olduğu otoyoldaki sıkışıklığa benzemektedir. Genelde trafiğin sebebi tek bir araba değildir. Basit olarak bu otoyolda çok sayıda araba vardır.

Bir makine, işlemden geçirebileceğinden daha hızlı bir şekilde yoğun datagram akışı alırsa ne olur? Onları buffer denilen bir hafıza bölümünde saklar. Fakat bu bellekleme işlemi, sadece datagramların, küçük bir burst'ın parçası olması durumunda problemi çözebilir. Şayet değilse ve yoğun datagram akışı devam ederse, cihazın hafızası, sonunda yorgun düşecek, kapasitesinin üstüne çıkmış olacak ve ulaşan ilave datagramları atarak tepki verecektir.

Yine de çok endişelenecek bir durum yoktur. Aktarım fonksiyonu nedeniyle, ağ akış kontrol sistemleri gerçekten iyi çalışmaktadır. Kaynakları atmak veya verinin kaybolmasına izin vermek yerine, aktarım (şekil 1.10'da gösterildiği gibi), yoğun akışın göndericisine veya kaynağına bir "hazır değil" uyarısı gönderebilir. Bu mekanizma, bunalmış host'a, segment trafiğini aktarmayı durdurması için, gönderici makineye sinyal gönderen, bir trafik lambasına benzer şekilde çalışır. İlgili alıcı, zaten hafızasında saklanan segmentleri işlemden geçirdikten sonra, aktarım hazır uyarısı gönderir. Datagramlarının kalanını aktarmayı bekleyen makine, bu "devam" uyarısını aldığında, aktarımına devam eder.




***Şekil 1.10:*** *Akış kontrolüyle segmentlerin aktarılması.*

Esaslı ve güvenli connection-oriented veri transferinde, datagramlar aktarıldıkları aynı sırayla alıcıya taşınır. Bu sıra bozulursa aktarım başarısız olur. Şayet herhangi bir segmenti kaybolmuş, tekrarlanmış veya hasar görmüşse, hata aktarılacaktır. Bu problem, alıcı host'un aldığı tüm veri segmentini onaylanmasıyla çözülür.

Bir servis, aşağıdaki özelliklere sahip olursa connection-oriented sayılmaktadır:

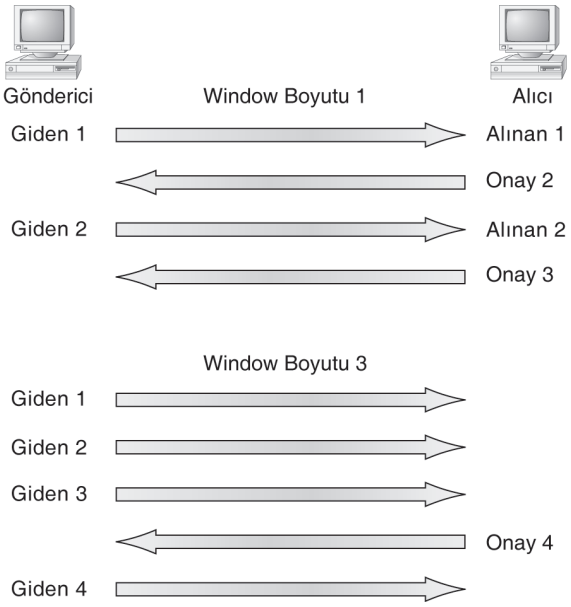
- Sanal bir devre kurulur (bir üç-yönlü anlaşma vs.).
- Sequencing kullanır.
- Onaylama kullanır.
- Akış kontrolü kullanır.

**NOT** *Akış kontrolü türleri, buffering, windowing ve tıkanıklıktan kaçınmadır.*

## Windowing

İdeal olarak, veri throughput'u, çabuk ve verimli gerçekleşir. Hayal edebileceğiniz gibi, her segment gönderildikten sonra, kaynak makine bir onay beklemek zorunda olduğundan, yavaşlık olacaktır. Fakat gönderici veri segmentini aktardıktan *sonra* ve alıcı makineden onaylama işlemini tamamlanmadan önce, uygun bir zaman aralığı olduğundan, gönderici bu arayı, daha fazla veri aktarmak için bir fırsat olarak kullanır. Bir onay almadan göndermek için aktarıcı makinesi tarafından kabul edilen (byte'larla ölçülen) veri segmentlerinin miktarı, window olarak belirtilir.

**NOT** *Window'lar, önemli, onaylanmamış veri segmentlerinin miktarını kontrol etmek için kullanılır.*




***Şekil 1.11:*** *Windowing.*

Böylece, window boyutu, bir uçtan diğerine ne kadar bilginin aktarıldığını kontrol eder. Bazı protokoller, paket numaralarını gözlemleyerek bilginin miktarını ölçerken, TCP/IP onu, byte sayılarını sayarak ölçer.

Şekil 1.11'de gördüğünüz gibi biri 1'e, diğeri 3'e ayarlı iki window boyutu vardır.

1 boyutunda bir window oluşturduğunuzda, gönderen makine, diğerini göndermeden önce, her veri segmenti için bir onay bekler. Şayet 3 boyutunda bir window oluşturduysanız, bir onay alınmadan önce, üç veri segmenti aktarmasına izin verilecektir.

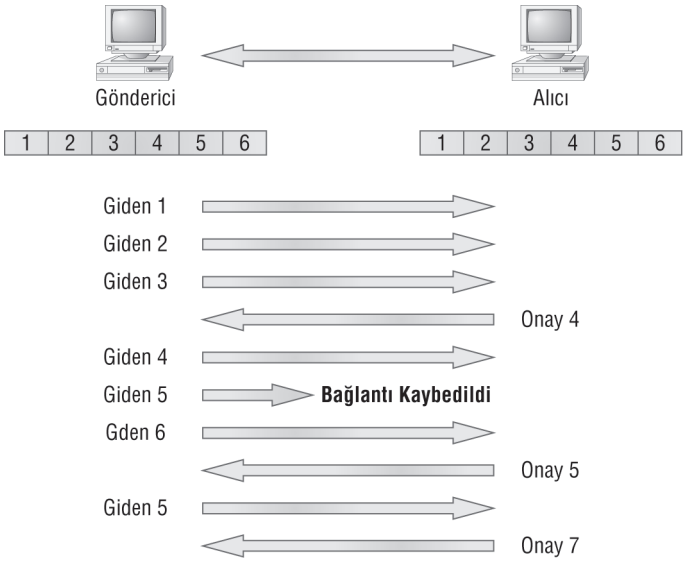
Basitleştirdiğimiz örneğimizde, gönderici ve alıcı makinelerin her ikisi de workstation'dır. Gerçekte bu az sayılarda olmaz, gönderilebilen byte miktarındadır.

*Şayet alıcı bir makine, onaylaması gereken tüm segmentleri almakta başarısız olursa host, window boyutunu küçülterek iletişim oturumunu düzeltebilir.*

**NOT**

## Acknowledgment (Onaylar)

Güvenli veri teslimi, tamamıyla fonksiyonel bir veri linki boyunca, bir makineden diğerine gönderilen veri akış bütünlüğünü garantiye alır. Bu, verinin, tekrarlanmamasını ve kaybolmamasını garanti eder. Bu, positive acknowledgement with retransmition olarak bilinen, veriyi aldığında, tekrar göndericiye bir onay mesajı göndererek aktarıcı kaynakla, haberleşmek için bir alıcı makine gerektiren bir teknik yardımıyla başarılmaktadır. Sonraki segmenti göndermeden önce, gönderici, gönderdiği ve onay için beklediği her segmenti belgeler. Bir segment gönderdiğinde, aktarıcı makine, bir timer başlatır ve alıcı uçtan bir onay dönmeden önce, süresi dolarsa tekrar aktarır.




***Şekil 1.12:*** *Transport katmanı güvenli teslimi.*

Şekil 1.12'de, gönderici makine, 1, 2 ve 3 segmentlerini aktarır. Onu onaylayan alıcı düğüm, segment 4'ü isteyerek onları alır. Ondan sonra gönderici, segment 4, 5 ve 6'yı aktarır. Şayet segment 5 hedefe gönderilemezse, alıcı düğüm, segmentin tekrar gönderilmesi isteğini onaylar. Sonra gönderici makine, kayıp segmenti tekrar gönderir ve segment 7'nin aktarımına devam etmek için, bir onay bekler.

## Network Katmanı

Network katmanı (ayrıca katman 3 olarak bilinir), cihaz adreslemelerini yönetir, ağdaki cihazların lokasyonunu izler ve verinin taşınması için en iyi yolu belirler. Yani Network katmanı, yerel olarak bağlı olmayan cihazlar arasındaki trafiği aktarmak zorundadır. Router'lar (katman3 cihazlar) Network katmanında belirtilir ve bir ağ topluluğundaki routing servislerini sağlar.

Bu şu şekilde olur: İlk olarak, router interface'inden bir paket alındığında hedef IP adresi kontrol edilir. Şayet paket bu belirli router'a gönderilmediyse, routing tablosundaki hedef network adresine bakacaktır. Router bir çıkış interface'i seçtiğinde paket, frame'lenmek ve yerel ağa gönderilmek için bu interface'e yollanacaktır. Şayet, router, routing tablosunda, paketin hedef ağı için bir kayıt bulamazsa, router paketi iptal eder.

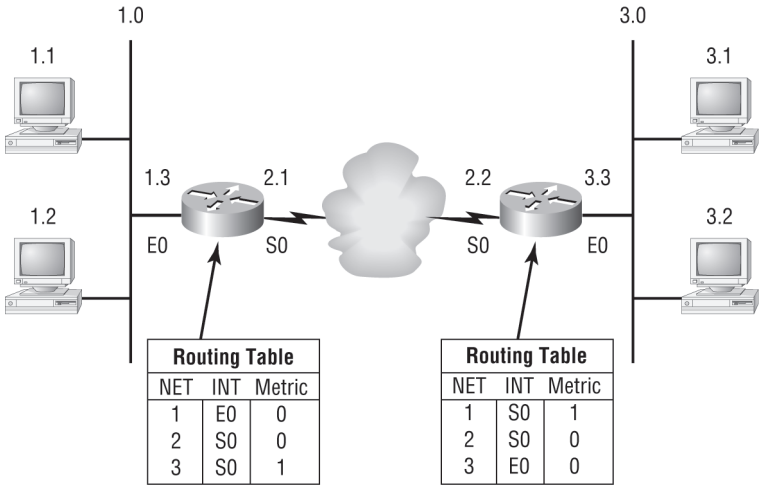
Network katmanında iki paket türü kullanılmaktadır: Veri ve route güncellemeleri.

**Veri paketleri:** Ağ topluluğu boyunca kullanıcı verisini aktarmak için kullanılır. Veri trafiğini desteklemek için kullanılan protokoller, routed protokoller olarak belirtilir; IP ve IPv6, routed protokol örnekleridir. Bölüm 2 ve 3'te IP adreslemesi, bölüm 13'te IPv6 hakkında bilgi alacaksınız.

**Route güncelleme paketleri:** Ağ topluluğundaki tüm router'lara bağlı ağlar hakkında komşu router'ları güncellemek için kullanılır. Route güncelleme paketi gönderen protokoller, routing protokolleri olarak belirtilir; RIP, RIPv2, EIGRP ve OSPF yaygın olarak kullanılan routing protokollerdir. Route güncelleme paketleri, her router'da routing tablolarını oluşturmaya ve devam ettirmeye yardım etmek için kullanılır.

Şekil 1.13'de, size bir routing tablosu örneği verdim. Bir router'da kullanılan routing tablosu aşağıdaki bilgileri içermektedir:

**Network adresleri:** Protokole özgü network adresleridir. Her routing protokolü, ağı farklı adresleme tasarımı (örneğin IP, IPv6 ve IPX) ile izlediğinden, bir router, ayrı routing protokolleri için bir routing tablosu oluşturmalıdır. Onu, belirli bir caddede yaşayan sakinlerin konuştuğu farklı dillerde bir trafik işaretçisi gibi düşünün. Şayet, Cat ismindeki bir caddede Amerikan, İspanyol ve Fransız halkı varsa, ışık, Cat/Gatro/Chat'i gösterir.

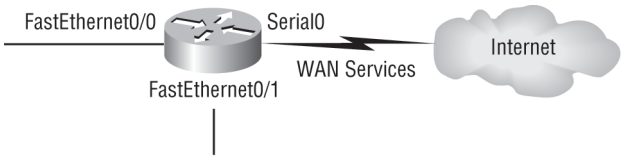



***Şekil 1.13:*** *Bir router'da kullanılan routing tablosu.*

**Interface:** Belirli bir ağı hedeflediğinde, bir paketin bulunacağı çıkış interface'idir.

**Metrik:** Uzak ağa uzaklıktır. Farklı routing protokolleri bu mesafenin hesaplanmasında farklı yollar kullanırlar. Routing protokollerini bölüm 6 ve 7'de ele alacağım, fakat şimdilik, diğerleri bant genişliği, hattın gecikmesi ve hatta tick count (saniyenin 1/18'i) bile kullanırken, bazı routing protokollerinin (RIP) hop count (bir paketin, bir route boyunca uzak bir ağa giderken uğradığı router sayısı) olarak belirtilen bir mekanizma kullandığını bilin.

Daha önce de belirttiğim gibi, router'lar broadcast domain'lerini ayırır. Yani, varsayılan olarak broadcast'ler bir router tarafından geçirilmezler. Bunun neden iyi bir şey olduğunu hatırladınız mı? Router'lar, ayrıca collision domain'lerini de ayırır. Fakat bunu katman2 switch'lerle de yapabilirsiniz. Router'daki her interface ayrı bir ağı gösterdiğinden, onlara ayrı network kimlik numarası verilmelidir ve bu router'a bağlı ağdaki her host, aynı network numarasını kullanmalıdır. Şekil 1.4, bir ağ topluluğundaki bir router'ın nasıl çalıştığını göstermektedir.




***Şekil 1.14:*** *Bir ağ topluluğundaki router.*

Burada router'lar hakkında ezberlemeniz gereken bazı noktalar vardır:

- Varsayılan olarak, router'lar broadcast ve multicast paketlerini geçirmeyecektir.
- Router'lar, paketi göndereceği next-hop router'ı belirlemek için network katmanı başlığındaki mantıksal adresi kullanır.
- Router'lar, bir interface'e giriş ve çıkışı kabul edilen paket tiplerindeki güvenliği kontrol etmek için bir yönetici tarafından oluşturulan access list'leri kullanabilir.
- Router'lar, ihtiyaç duyulduğunda katman2 bridging fonksiyonu sağlayabilir ve aynı interface üzerinden eşzamanlı route edebilirler.
- Katman3 cihazlar (burada router'lar), sanal LAN'lar (VLAN'lar) arasında bağlantı sağlar.
- Router'lar, belirli bir network trafik çeşidi için QoS (quality of services) sağlayabilir.

*Anahtarlama ve VLAN'lar bölüm 8, "LAN switching ve STP" ve bölüm 9, "Sanal LAN'larda (VLAN'lar) ele alınmaktadır.*

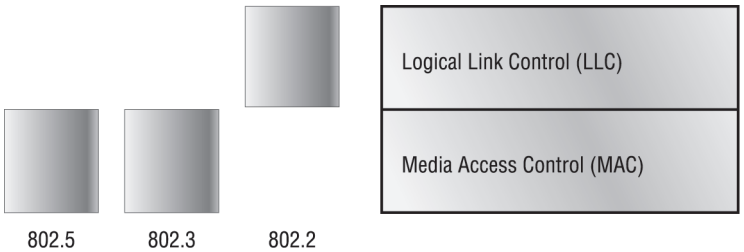
NOT

## Data Link Katmanı

Data link katmanı, verinin fiziksel aktarımını sağlar. Aynı zamanda hata bildirme, ağ topolojisi ve akış kontrolünü ele alır. Yani Data Link katmanı, donanım adresi kullanan bir LAN'daki belirli bir cihaza mesajların taşınmasından emin olacaktır ve aktarılması için Network katmanından gelen mesajları, Physical katman için bit'lere dönüştürecektir.

Data link katmanı, mesajları her biri veri frame'i olarak belirtilen parçalar olarak biçimlendirir ve donanım hedefi ve kaynak adresi içeren özel bir başlık ekler. Bu ilave bilgi bir bakıma, Apollo uzay projesinde ay modülüne eklenmiş, farklı ekipmanların bir kapsül ile çevrelenmesi gibi, orijinal mesajları çevreler. Bu farklı ekipman çeşitleri, sadece uzay uçuşunun belirli evrelerinde kullanışlıdır ve onlar için tanımlanmış evre tamamlandığında, modülden ayrılırlar ve atılırlar. Network boyunca verinin dolaşması prosesi de benzerdir.

Şekil 1.15, Ethernet ve IEEE düzenlemeleri ile Data Link katmanını göstermektedir. Onu incelediğinizde, IEEE 802.2 standardının, diğer IEEE standartlarıyla birlikte kullanıldığını ve onlara işlevsellik kazandırdığını fark edersiniz.




***Şekil 1.15:*** *Data link katmanı.*

Network katmanında çalışan router'ların, belirli bir host'un lokasyonuyla tamamıyla ilgilenmediklerini anlamanız önemlidir. Onlar sadece ağın nerede oluşturulduğu ve onlara erişmenin en iyi yoluyla ilgilenirler. Router'lar, ağa katıldıklarında, tamamıyla zorlayıcıdırlar. Ve ilk sefer için bu iyi bir şeydir. Data link katmanı, yerel bir ağda bulunan her cihazın, benzersiz tanımlanmasından sorumludur.

Hem bir kullanıcının, yerel ağdaki ayrı makinelere paketlerini göndermek hem de router'lar arasında paketleri aktarmak için Data Link katmanı, donanım adreslemesi kullanır. Router'lar arasında bir paket gönderildiğinde, paket Data link katmanındaki kontrol bilgisiyle frame'lenir, fakat bu bilgi alıcı router'da atılır ve sadece orijinal paket tamamıyla sağlam kalır. Bu paket frame işlemi, her hop için paketin sonunda doğru alıcıya ulaştırılmasına kadar devam eder. Paketin kendisinin bir route boyunca asla değişmediğini anlamak gerçekten önemlidir; o sadece, farklı medya türlerine uygun şekilde aktarılması için gerekli kontrol bilgisi tipiyle enkapsüle (kapsüllenmek) edilmektedir.

IEEE Ethernet Data link katmanı iki alt katmana sahiptir:

**Media Access Control (MAC) 802.3:** Paketlerin, ortam araçlarına nasıl yerleştirildiğini açıklar. Çekişmeli ortam erişimi, herkesin aynı bant genişliğini paylaştığı, "ilk gelen/ilk servis alır" erişimidir. Fiziksel adresleme de, mantıksal topolojiler de burada tanımlanmaktadır. Mantıksal topoloji nedir? Fiziksel bir topoloji boyunca sinyal yoludur. Hat disiplini, hata uyarısı(düzeltmesi değil), frame'lerin sıralı teslimi ve opsiyonel akış kontrolü de yine bu alt katmanda kullanılabilir.

**Logical Link Control (LLC) 802.2:** Network katmanı protokollerinin belirlenmesinden ve sonra onların enkapsüle edilmesinden sorumludur. Bir LLC başlığı, data link katmanına, frame alınınca, bir paketi ne yapacağını söyler. Şu şekilde çalışır: Bir host, bir frame alacak ve paketin nereye hedefleneceğini (örneğin, Network katmanındaki IP protokolüne) belirlemek için LLC başlığına bakacaktır.

Bölümün başında bahsettiğim switch ve bridge'lerin her ikisi de, Data link katmanında çalışır ve donanım (MAC) adreslerini kullanarak ağı filtreler. Buna şimdiki bölümde bakacağız.

## Data Link Katmanında Switch ve Bridge'ler

Katman2 switching, bir ASIC (application-specific integrated circuit) olarak bilinen özel donanım kullandığı için donanım-tabanlı bridging olarak kabul edilir. ASIC'ler çok küçük gecikme değerleriyle, gigabit hızına ulaşabilir.

Bridge ve switch'ler, her frame'i ağ boyunca dolaşıyor gibi okur. Bundan sonra katman2 cihazı, kaynak donanım adresini bir filtre tablosuna koyar ve frame'in alındığı port için bir kayıt ekler. Bu bilgi (bridge veya switch'in filtre tablosunda tutulan) makinenin belirli bir gönderici cihazın lokasyonunu belirlemesine yardım eder. Şekil 1.16, ağ topluluğundaki bir switch'i göstermektedir.

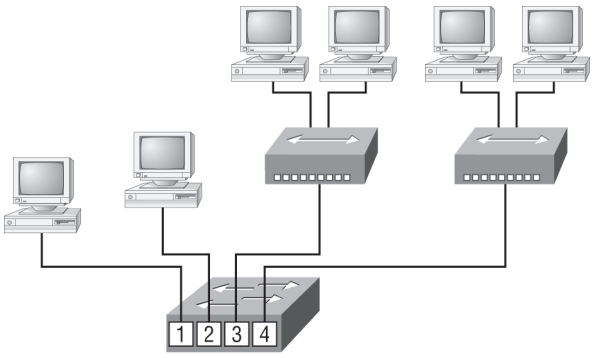
NOT *Latency (gecikme süresi) bir frame'in bir porta girişinden diğer portun çıkışına kadar ölçülen zamandır.*

Emlak işi, tamamıyla lokasyonla ilgilidir. Bu, katman2 ile katman3 cihazları için de aynıdır. Her ikisinin de ağı düzenleyebilmesi gerektiği halde, ağın farklı kısımlarıyla ilgilendiklerini hatırlamak önemlidir. Öncelikle, katman2 makineleri (switch ve bridge'ler), belirli cihazların yerini belirlemeye ihtiyaç duyarlarken, katman3 cihazlar (router gibi) belirli ağların yerini belirlemeye gereksinim duyar. Öyleyse ağlar, router'lar, bireysel cihazlar, switch'ler ve bridge'ler içindir. Ayrıca bireysel cihazların haritasını çıkaran filtre tablolarının switch ve bridge'ler için olduğu gibi, ağların haritasını çıkaran routing tabloları da router'lar içindir.

Katman2 cihazında bir filtreleme tablosu oluşturulduktan sonra, frame'leri sadece hedef adresin yerleştirildiği segmente gönderecektir. Şayet, hedef cihaz, frame ile aynı segmentteyse, katman2 cihaz, diğer segmentlere giden frame'leri bloklayacaktır. Şayet hedef farklı bir segmentte ise, frame, sadece bu segmente aktarılabilecektir. Bu *t*ransparent bridging olarak belirtilir.

Bir switch interface'i, cihazın filtre tablosunda olmayan bir hedef adresine sahip bir frame aldığında, frame'i bağlı tüm segmentelere gönderecektir. Şayet bilinmeyen frame gönderilen bilinmeyen cihaz, gönderme işlemini cevaplarsa, switch bu cihazın lokasyonu ile ilgili filtre tablosunu günceller. Fakat gönderilen frame'in hedef adresinin, bir broadcast adresi olması durumunda switch, varsayılan olarak broadcast'leri bağlı tüm segmentlere gönderecektir.

Broadcast'in gönderildiği tüm cihazların, aynı broadcast domain'inde oldukları farz edilir. Bu bir problem olabilir: Katman2 cihazlar, boğucu seviyede katman2 broadcast fırtınası yayar ve bir ağ topluluğundan yayılan broadcast fırtınasını durdurmanın tek yolu, router gibi bir katman3 cihazıdır.




***Şekil 1.16:*** *Ağ topluluğundaki bir switch.*

Ağ topluluğunuzda hub yerine bir switch kullanmanızın en büyük kazancı, her switch portunun aslında kendi collision domain'ine sahip olmasıdır (tersine bir hub geniş bir collision domain'i oluşturur). Fakat switch'iniz olsa dahi, hala broadcast domain'lerini ayıramazsınız. Ne switch ne de bridge'ler bunu yapacaktır. Tipik olarak tamamen tüm broadcast'i geçireceklerdir.

Hub-merkezli kurulumlara karşı LAN switching'in diğer avantajı, switch'e bağlı her segmentteki tüm cihazların, eş zamanlı aktarım yapabilmesidir. En azından, her portta bir host varsa ve bir switch portuna hub bağlı değilse yapabilirler. Tahmin edebileceğiniz gibi hub'lar, bir seferinde, network segmenti başına sadece bir cihaza izin verir.

## Binary'den Decimal ve Hexadecimal Dönüştürme

Bu bölümü tamamlamadan ve bölüm 2'deki TCP/IP protokol yığını ve IP adreslemesini tartışmaya geçmeden önce binary, decimal ve hexadecimal numaralar arasındaki farklılıkları ve birinden diğerine nasıl dönüştürüleceğini bilmeniz çok önemlidir.

Binary numaralamadan başlayacağız. Gerçekten çok basittir. Kullanılan rakamlar, 1 veya 0 ile sınırlıdır ve her rakam, 1 bit'tir (binary digit için kısaltmadır). Tipik olarak, birlikte 4 veya 8 bit sayarsınız, yarım byte (nibble) veya byte olarak belirtilir.

Binary numaralamada bizi ilgilendiren, anaokulundan beri kullandığımız 10-tabanlı numara düzenlemesiyle tipik ondalık formatta gösterilen bir değerdir. Binary numaralar, bir değer alanına yerleştirilir: sağdan başlar ve sola doğru devam eder. Her spot, önceki spot değerinin iki katına eşit bir değere sahiptir.

Şekil 1.1 nibble ve bir byte'taki her bit lokasyonunun ondalık değerlerini göstermektedir Nibble'ın 4 bit, bir byte'ın 8 bit olduğunu hatırlayın.

**Tablo 1.1:** Binary Değerleri

| Nibble Değerleri | Byte Değerleri |
|---|---|
| 8 4 2 1 | 128 64 32 16 8 4 2 1 |

Bütün bunların anlamı şudur; eğer 1, bir değer alanına yerleştirildiğinde nibble ya da byte bu ondalık değerle ilgilenir ve onu, diğer 1'e sahip değer alanlarına ekler. Şayet bit alanına bir 0 konulursa, bu değeri saymazsınız.

Bazı şeylere açıklık getirmeme izin verin. Şayet, nibble'ımızın her alanına yerleştirilmiş 1'e sahip olursak, bize maksimum 15 değerini vermesi için, 8+4+2+1'i toplarız. Nibble değerimiz için başka bir örnek, 1010 olabilir; bunun anlamı, 8 bit ve 2 bit kullanılmaktadır ve ondalık olarak 10 sayısına eşittir. Şayet 0110 binary nibble değerine sahip olursak, 4 ve 2 kullanıldığından, ondalık değerimiz 6 olur.

Fakat byte değerleri, 15'ten büyük bir değere ulaşabilir. Şayet her bit'i (1) olarak sayarsak, byte binary değeri şu şekilde olurdu (bir byte'ın 8 bit'e eşit olduğunu hatırlayın):

**11111111**

Hepsi kullanıldığından, tüm bit alanlarını sayarız. Bir byte'ın en yüksek değeri olarak şu şekilde görülürdü:

**128+64+32+16+8+4+2+1 = 25**

Bir binary numaranın eşit olabileceği çok sayıda ondalık değer vardır. Bazı örneklere bakalım:

**10010110**

Hangi bit'ler kullanılmaktadır? 128, 16, 4 ve 2 bit'leri kullanılmaktadır, bu nedenle sadece onları toplarız: 128+16+4+2 = 150

**01101100**

Hangi bit'ler kullanılmaktadır? 64, 32, 8 ve 4 bit'leri kullanılmaktadır, bu nedenle sadece onları toplarız: 64+32+8+4 = 108.

**11101000**

Hangi bit'ler kullanılmaktadır? 128, 64, 32 ve 8 bit'leri kullanılmaktadır, bu nedenle sadece onları toplarız: 128+64+32+8 = 232.

Tablo 1.2, bölüm 2 ve 3'deki IP bölümlerinden önce ezberlemeniz gereken bir tablodur.

**Tablo 1.2:** Binary'den Ondalık Sayıya Ezber Tablosu

| Binary Değer | Decimal Değer |
|---|---|
| 10000000 | 128 |
| 11000000 | 192 |
| 11100000 | 224 |
| 11110000 | 240 |
| 11111000 | 248 |
| 11111100 | 252 |
| 11111110 | 254 |
| 11111111 | 255 |

Hexadecimal adresleme, binary ve ondalık'tan tamamen farklıdır. Onlar, byte''ların değil de, nibble'ların (yarım byte'ların) okunmasıyla çevrilmektedir. Nibble kullanılarak, bit'leri kolayca hex'e çevirebiliriz. İlk olarak, hexadecimal adreslemenin sadece 0'dan 9'a kadar numaraları kullandığını anlamalısınız. 10, 11, 12 ve devamındakiler kullanılmaz (çünkü çift rakamlı numaralardır), 10, 11, 12, 13, 14 ve 15'i göstermek için A, B, C, D, E ve F kullanılmaktadır.

NOT *Hex, ondalık sistemdeki kullanılabilir 10 sayıyı genişletmek için alfabenin ilk altı harfini (A'dan F'ye kadar) kullanan bir numaralama sistemidir. Hexadecimal, toplam 16 karakter içermektedir.*

Tablo 1.3, her hexadecimal değeri için, binary ve ondalık değerleri göstermektedir.

**Tablo 1.3:** Hex'i, Binary ve Ondalığa Dönüştürme Tablosu

| Hexadecimal Değer | Binary Değer | Decimal Değer |
|---|---|---|
| 0 | 0000 | 0 |
| 1 | 0001 | 1 |
| 2 | 0010 | 2 |
| 3 | 0011 | 3 |
| 4 | 0100 | 4 |
| 5 | 0101 | 5 |
| 6 | 0110 | 6 |
| 7 | 0111 | 7 |
| 8 | 1000 | 8 |
| 9 | 1001 | 9 |
| A | 1010 | 10 |
| B | 1011 | 11 |
| C | 1100 | 12 |
| D | 1101 | 13 |
| E | 1110 | 14 |
| F | 1111 | 15 |

İlk 10 hexadecimal rakam (0-9) ile ondalık değerlerin aynı olduğunu fark ettiniz mi? Şayet hayırsa, tekrar bakın. Bu faktör, bu değerlerin çok kolay çevrilmesini sağlar.

Şu şekilde yazılış tarzı görebilirsiniz: 0x6A. (Bazen Cisco karakterlerin önüne 0x koyar, böylece bir hex değeri olduğunu bilirsiniz. Başka özel bir anlamı yoktur.) Binary ve ondalık değerler nedir? Hatırlamanız gereken, her hex karakterinin nibble olduğu ve iki karakterin beraber bir byte olduğudur. Binary değerini anlamak için, hex karakterlerini bir nibble'a ve sonra bunları beraber bir byte'a koymaya ihtiyacımız vardır. 6=0110 ve A (hex'de 10 dur)= 1010, böylece tüm byte 01101010 olur.

Binary'den hex'e çevirmek için, byte'ı alın ve onu iki nibble'a ayırın.

01010101 binary numaranız var. İlk olarak,1 ve 4 kullanılmakta olduğundan, 0101 ve 0101 şeklinde, her biri 5 olan nibble'lara ayırın. Bu, hex karşılığını 0x55 yapar. Ve ondalık formatta, binary numarası 01010101'ı 64+16+4+1=85 olarak çevrilir.

Başka bir binary numara:

**11001100**

Cevabınız 110=12 ve 1100=12 (hex'de CC olarak çevrilir). Ondalık çeviri cevabınız 128+64+8+4=204 olur.

Bir örnek daha verelim, ondan sonra Physical katmanı çalışalım. Farz edelim şu binary numaranız var:

**10110101**

NOT *Binary/hex/decimal ile ilgili daha fazla örnek için Yazılı Lab 1.4'e bakınız.*

Hex karşılığı, 1011, B'ye ve 0101, 5'e çevrildiğinden cevap 0xB5 olacaktır. Ondalık karşılığı 128+32+16+4+2=181'dir.

## Physical Katman

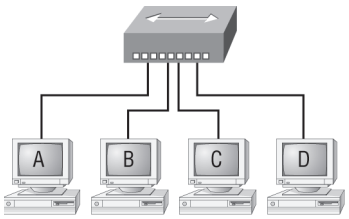
Son olarak en alt katmana geldiğimizde karşılaştığımız Physical katman iki iş yapar: Bit'leri gönderir ve alır. Bit'ler, sadece sayısal bir Mors koduyla, 1 ve 0 değerlerinde gelir. Physical katman, farklı tip iletişim ortam araçlarıyla direk olarak haberleşir. Farklı tür ortam araçları, bu bit değerlerini farklı yollarla gösterirler. Bazıları, yüksekten düşüğe ve düşükten yükseğe voltajdaki değişimlere göre durum geçişleri kullanırken, bazıları radyo ses tonları kullanır. Kullanılacak özel bit örneklerini açıklamak için her ortam tipi için özel protokole ihtiyaç vardır.

Physical katman, uç sistemler arasında fiziksel bir linki aktifleştirmek, sürdürmek ve pasif hale getirmek için elektriksel, mekanik ve fonksiyonel gereklilikler tanımlar. Bu katman ayrıca DTE (data terminal equipment) ve DCE (data communication equipment) arasındaki interface'i belirlediğiniz yerdir. (Bazı eski telefon firmaları hala DCE kullanır.) DTE, bağlı cihazken, DCE genellikle servis sağlayıcıda bulunur. DTE uyumlu servislere, genellikle bir modem ya da CSU/DSU (channel service unit/data service unit) yardımıyla erişilebilir.

Physical katman konnektörleri ve farklı fiziksel topolojiler farklı sistemlerin iletişimini kabul eden standartlar olarak, OSI tarafından tanımlanırlar. CCNA konuları sadece IEEE Ethernet standartlarıyla ilgilenir.

### Physical Katmandaki Hub'lar

Bir hub aslında çok portlu bir repeater'dır. Repeater, dijital bir sinyali alır, bu sinyali tekrar üretir veya tekrar güçlendirir. Sonra herhangi bir bilgiye bakmaksızın tüm aktif portlardan gönderir. Aktif bir hub da aynı şeyi yapar. Hub portundaki bir segmentten alınan bir dijital sinyal, tekrar üretilir veya tekrar güçlendirilir ve hub'ın tüm portlarından iletilir. Yani, hub'a bağlı tüm cihazlar hem aynı collision domain'inde hem de aynı broadcast domain'indedirler. Şekil 1.17 ağdaki bir hub'ı göstermektedir.




Tüm cihazlar aynı collision domain'indedir.
Tüm cihazlar aynı broadcast domain'indedir.
Cihazlar aynı bant genişliğini paylaşırlar.

***Şekil 1.17:*** *Ağdaki bir hub.*

Hub'lar, repeater'lar gibi fiziksel ortama girerken ve aktarılırken, herhangi bir trafiği incelemez. Hub ya da hub'lara bağlı her cihaz, bir cihazın aktarım yapıp yapmadığını dinlemek zorundadır. Hub'ın bir merkez cihaz olduğu ve kabloların ondan her yöne uzatıldığı bir fiziksel yıldız ağı, hub'ın oluşturduğu bir topoloji çeşididir. Sinyalin ağ boyunca uçtan uca gitmek zorunda olduğu Ethernet ağları, mantıksal bir bus topoloji çalışırken, hub tasarımı görsel olarak gerçekten bir yıldızı andırır.

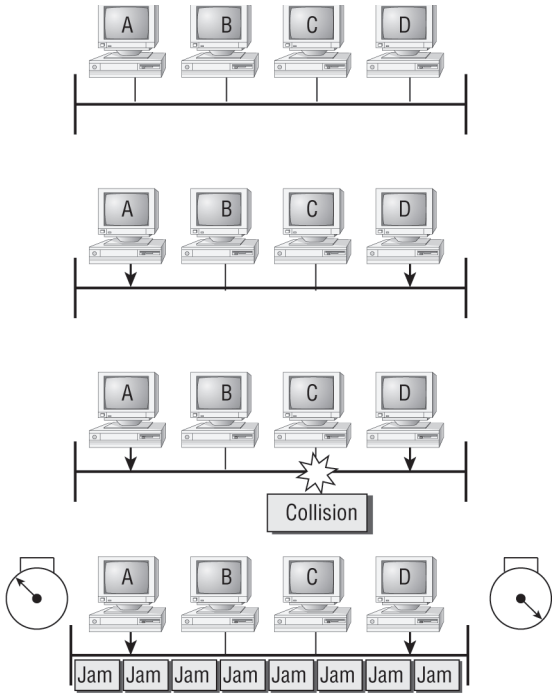
NOT *Bunu tavsiye etmesem de, hub ve repeater'lar, tek bir LAN segmentinin kapsadığı alanı genişletmek için kullanılabilmektedir. LAN switch'ler, nerdeyse her durum için finansal olarak uygundur.*

# Ethernet Ağ Kurulumu

Ethernet, ağdaki tüm kullanıcı makinelere, bir linkin aynı bant genişliğini paylaşmasına izin veren bir ortam erişim yöntemidir. Kolayca ölçeklenebilir olmasından dolayı, Ethernet popülerdir. Yani, bir Fast Ethernet ve Gigabit Ethernet gibi yeni teknolojiler mevcut ağ altyapılarıyla oldukça kolay bütünleşmektedir. Ayrıca, ilk yapılandırma çalışmaları nispeten daha kolaydır ve hata giderimleri oldukça basittir. Ethernet Data link ile Physical katman düzenlemelerini kullanır ve modülün bu bölümü, bir Ethernet ağını etkin olarak çalıştırmak, arıza gidermek ve sürdürmek için ihtiyacınız olan Data link ve Physical katman bilgilerini size verecektir.

Ethernet ağ kurulumu, cihazların, ağ ortamında aynı anda iki cihazın aktarım yapmadan, bant genişliğini paylaşmasına izin veren bir protokol olan CSMA/CD'yi (Carrier Sense Multiple Access

with Collision Detection) kullanır. CSMA/CD, paketlerin farklı noktalardan eşzamanlı aktarıldığında oluşan collision problemlerinin üstesinden gelinmesi için geliştirilmiştir. Bir düğüm, CSMA/CD ağında aktarım yaptığında, ağdaki diğer tüm düğümler bu aktarımı alıp incelediğinden, iyi bir collision yönetimi çok önemlidir. Sadece bridge ve router'lar, etkin olarak bir aktarımın tüm ağ boyunca yayılmasını engellemektedirler!




Carrier Sense Multiple Access İle Collision Detection (CSMA/CD)

***Şekil 1.18:*** *CSMA/CD.*

Peki, CSMA/CD protokolü nasıl çalışır? Gelin Şekil 1.18'e bakarak başlayalım..

Bir kullanıcı makinesi ağ üzerinde aktarım yapmak istediği zaman, ilk olarak kabloda dijital bir sinyalin varlığını kontrol eder. Eğer her şey temizse (diğer host'lar aktarım yapmıyorsa), o zaman kullanıcı aktarmaya devam eder. Aktarıcı makine, diğer kullanıcıların aktarıma başlamadığına emin olmak için, kabloyu sürekli izler. Şayet kullanıcı kabloda başka bir sinyal algılarsa, segmentteki tüm kullanıcıların veri gönderimini durdurmasına sebep olan genişletilmiş bir jam sinyali gönderir. Düğümler, tekrar aktarıma başlamadan önce bir süre bekleyerek bu sinyale karşılık verirler. Backoff algoritması, çarpışan istasyonların tekrar ne zaman aktarımda bulunabileceğine karar verir. Şayet 15 denemeden sonra, collision olmaya devam ederse, o zaman aktarımı deneyen düğümler, zaman aşımına uğrayacaktır.

Bir Ethernet LAN'ında collision olduğunda, şunlar meydana gelir:

- Bir jam sinyali, bir collision olduğu hakkında tüm cihazları bilgilendirir.
- Collision, rastgele bir backoff algoritması başlattırır.
- Ethernet segmentindeki her cihaz, timer sonlanıncaya kadar kısa bir süre aktarımı durdurur.
- Timer sonlandıktan sonra, tüm kullanıcılar, aktarım için eşit önceliğe sahiptir.

Yoğun collision'ları sürdüren bir CSMA/CD ağına sahip olmanın etkileri şunlardır:

- Gecikme
- Düşük throughput
- Tıkanıklık

*Bir 802.3 ağındaki backoff, bir collision olduğunda mecbur tutulan tekrar aktarım gecikmesidir. Bir collision olduğunda, bir kullanıcı, mecbur tutulan zaman gecikmesi sonlandıktan sonra aktarıma devam edecektir. Bu backoff gecikme periyotları dolduktan sonra, tüm istasyonlar veri aktarmak için eşit önceliğe sahip olurlar.*

NOT

Şimdiki bölümlerde, Data Link (katman2) ve Physical katmanda (katman1) Ethernet'i detaylı şekilde aktaracağım.

## Half ve Full-Duplex Ethernet

Half-duplex Ethernet, orijinal 802.3 Ethernet'te tanımlanmıştır. Cisco, onu, dijital sinyalin her iki yönde gidebildiği kabloda, sadece bir çifti kullanılması olarak belirtir. IEEE düzenlemeleri, half duplex işleyişinden biraz farklı olarak bahseder, Fakat Cisco'nun burada bahsettiği, Ethernet ile neler olduğunun genel bir algılamasıdır.

Half-duplex ayrıca, collision'dan korunmaya yardımcı olmak ve collission olduğunda tekrar aktarıma izin vermesi için CSMA/CD protokolünü kullanır. Şayet bir hub switch'e bağlıysa, uç istasyonların collision'ı algılayabilmelerinden dolayı, hub, half-duplex modda çalışmak zorundadır. Half-duplex Ethernet, yani tipik olarak 10BaseT, geniş bir 10BaseT ağının genelde sadece 3-4Mbps sağlamasından dolayı, Cisco'nun da belirttiği gibi, sadece yüzde 30-40 randımanda çalışır.

Fakat full-duplex Ethernet, half-duplex'deki tek çift yerine iki çifti kullanır. Ve full-duplex, verici ile alıcı cihaz arasında noktadan-noktaya bir bağlantı kullanır. Yani, full duplex veri transferiyle, half duplex'e kıyasla daha hızlı veri aktarımı sağlarsınız. Ve gönderilen veri ve alınan veri farklı kablo çiftinden aktarıldığından, collision olmaz.

Collision'dan endişe duymamamızın sebebi, half duplex'in sağladığı tek-şeritli yol yerine çok şeritli bir otoyola benzemesindendir. Full-duplex Ethernet'in, her iki yönde, %100 verimlilik önerdiği farz edilir. Örneğin, full-duplex çalışan bir 10Mbps Ethernet'ten 20Mbps veya Fast Ethernet için 200Mbps sağlayabilirsiniz. Bu değer bazen, %100 verimlilik almanız varsayılmaktadır şeklinde çevrilen, bir aggregate değeri olarak bilinmektedir. Gerçek yaşamdaki ağ kurulumunda bunun garantisi yoktur.

Full-duplex Ethernet üç durumda kullanılabilmektedir:

NOT *Sadece iki düğüm olduğunda, Full-duplex Ethernet, noktadan-noktaya bir bağlantı gerektirir.*

- Switch'ten, bir host'a bir bağlantıyla
- Switch'ten, switch'e bir bağlantıyla
- Bir kullanıcıdan, çapraz kablo kullanan bir kullanıcıya bağlantıyla

Şayet tüm bu hız kapasitesine sahipse, neden bu kapasitede taşınmaz? Bir full-duplex Ethernet portu çalışmaya başladığında, ilk olarak uzak uca bağlanır ve sonra Fast Ethernet linkin diğer uçlarıyla görüşür. Bu bir otomatik-algılama mekanizmasıdır. Bu mekanizma ilk olarak alıp gönderme kapasitesine karar verir. Yani 10Mbps veya 100Mbps'ta mı çalışabileceğini kontrol eder. Bundan sonra full-duplex'de çalışabileceğini kontrol eder, eğer çalışamazsa, o zaman half-duplex'te çalışacaktır.

NOT *Half-duplex Ethernet'in bir collision domain'ini paylaştığını ve tipik olarak özel bir collision domain'le daha yüksek throughput değerine sahip full-duplex Ethernet'ten daha düşük randımanda bir throughput değerini sağladığını hatırlayın.*

Son olarak, şu önemli noktaları hatırlayın:

- Full-duplex modda collision yoktur.
- Her full-duplex düğüm için, atanmış bir switch portu gerekmektedir.
- Host network kartı ve switch portu mutlaka full-duplex modda çalışma kapasitesinde olmalıdır.

Şimdi Ethernet, Data Link katmanında nasıl çalışır.

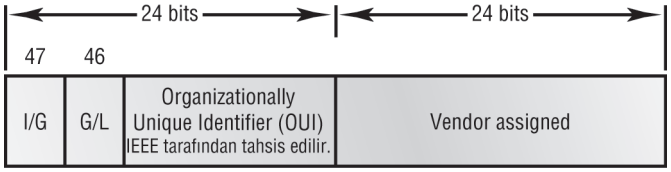
## Data Link Katmanında Ethernet

Data Link katmanında Ethernet, genelde donanım veya MAC adreslemesi olarak tanımlanan Ethernet adreslemesinden sorumludur. Ethernet aynı zamanda, Network katmanından alınan paketlerin frame'lenmesinden ve Ethernet ortam erişim yöntemi yardımıyla, yerel ağdaki bu paketlerin aktarım için hazırlanmasından da sorumludur.

### Ethernet Adreslemesi

Ethernet adreslemenin nasıl çalıştığına bakacağımız yer burasıdır. Adresleme, tüm Ethernet Network interface card'a (NIC) yazdırılan Media Access Control (MAC) adresini kullanır. MAC veya donanım adresi, hexadecimal formatta yazılı 48-bit (6-byte) bir adrestir.

Şekil 1.19, 48-bit MAC adresini ve bit'lerin nasıl bölündüğünü göstermektedir.




***Şekil 1.19:*** *MAC adres kullanarak Ethernet adreslemesi.*

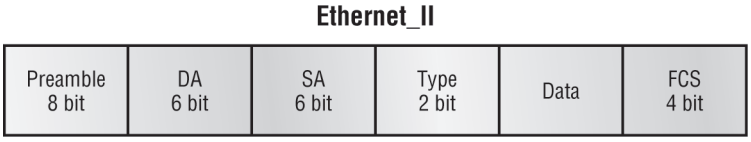
Organizationally unique identifier (OUI), IEEE tarafından bir kuruluşa tahsis edilmektedir. 24 bit ya da 3 byte'tan oluşmuştur. Kuruluş, sırayla, her adaptörün üretiminde benzersiz (varsayım olarak ve garantisi yok) olan global sağlanan bir (24 bit veya 3 byte) adres atar. Yakından şekle bakınız. Yüksek-öncelikli bit, Individual/Group (I/G) bit'idir. Sıfır değerine sahip olduğunda, bunun bir cihazın MAC adresi olduğunu ve MAC başlığının kaynak bölümünde görüneceğini varsayarız. Bu değer 1 ise, adresin, Ethernet'te bir broadcast ya da multicast adresi veya TR ve FDDI'da, bir broadcast ya da fonksiyonel adres olduğunu farz ederiz.

Sonraki bit, global/lokal bit ya da sadece G/L bit'idir (aynı zamanda, U'nun universal anlamına geldiği U/L olarak bilinir). Sıfıra ayarlandığında, bu bit, (IEEE'nin yaptığı gibi) bir global yönetim adresini gösterir. Bit 1 olduğunda, yerel bir yönetim adresini gösterir (DECnet'te kullanıldığı gibi). Ethernet adresinin düşük-değerli 24 bit'i bir yerel yönetimli ya da üretici-tahsisli kodu belirtmektedir. Bu bölüm yaygın olarak, üretilen ilk kart için 24 adet 0'dan (sıfır) başlar ve son (16,777,216'ıncı) üretilen kart için 24 adet 1'e kadar devam eder.

## Ethernet Frame'leri

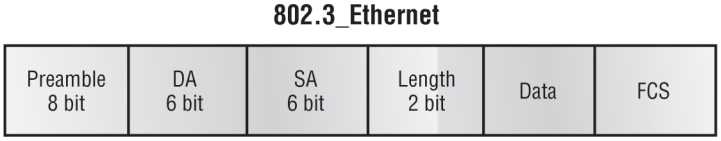
Data Link katmanı, bit'lerin byte'lara ve frame'lerin byte'lara birleştirilmelerinden sorumludur. Frame'ler, bir ortam aracı erişim türünde aktarmak amacıyla, Network katmanından gelen paketleri enkapsüle etmek için Data Link katmanında kullanılır.

Ethernet istasyonlarının fonksiyonu, MAC frame formatı olarak bilinen bir bit grubunu kullanarak, diğerleri arasında data frame'lerini geçirmektir. Bu, CRC (cyclic redundancy check) ile hata tespiti sağlar. Fakat bunun bir hata düzeltme değil de hata tespiti olduğunu hatırlayın. 802.3 ve Ethernet frame'leri şekil 1.20'de gösterilmektedir.




*Bir frame'i farklı bir frame türüne enkapsüle etmeye tunelling denilmektedir.*

NOT




***Şekil 1.20:*** *802.3 ve Ethernet frame formatları.*

Aşağıdakiler, 802.3 ve Ethernet frame çeşitlerindeki farklı alanlardaki detaylardır:

**Preamble:** Dalgalı bir 1,0 formu, her paketin başlangıcında bir 5MHz hız sağlar. Bu, alıcı cihazların, gelen bit akışını durdurmasına izin verir.

**Start Frame Delimiter (SFD)/Synch:** Preamble yedi oktet'tir ve SFD, bir oktet'tir(synch). SFD, son bir çiftinin, alıcının, ortada bir yerdeki 1,0 formunun değişimine, hala senkron durumunu korumasına ve verinin başlangıcını belirlemeye izin verdiği, 10101011'dır.

**Destination Address (DA):** Bu, (least significant bit) LSB-first kullanarak, 48-bit bir değer iletir. DA, gelen bir paketin belirli bir düğüme adreslenip adreslenmediğini tespit etmek için alıcı istasyonlar tarafında kullanılmaktadır. Hedef adresi, bireysel bir adres veya bir broadcast ya da multicast MAC adresi olabilir. Bir broadcast'in tamamıyla 1'lerden (veya hex olarak F'lerden) oluştuğunu veya tüm cihazlara gönderildiğini, fakat bir multicast'in sadece, bir ağdaki düğümlerin benzer alt gruplarına gönderildiğini hatırlayın.

**Source Address (SA):** SA, verici cihazları tespit etmek için kullanılan 48-bit bir MAC adresidir ve LSB-first kullanır. Broadcast ve multicast adres formatları SA alanında geçersizdir.

**Lenght veya Type:** 802.3, bir Lenght alanı kullanır, fakat Ethernet frame'i, Network katmanı protokolünü tespit etmek için bir Type alanı kullanır. 802.3, üst-katman protokollerini tanıyamaz ve (örneğin IPX gibi) tescilli bir LAN ile kullanılması gerekmektedir.

**Data:** Bu, Network katmanından, Data Link katmanına gönderilen bir pakettir. Boyutu, 64 ile 1,500 bit arasında değişebilir.

**Frame Check Sequence (FCS):** FCS, CRC'leri saklamak için kullanılan frame'in sonundaki bir alandır.

Burada bir ara verelim ve bizim güvenilir OmniPeek network analizörüne yakalanan bazı frame'lere bir bakalım. Aşağıda sadece, Destination, Source ve Type (bu analizörde Protocol Type olarak görünen) olarak sadece üç alana sahip frame görebilirsiniz:

```
Destination:        00:60:f5:00:1f:27
Source:             00:60:f5:00:1f:2c
Protocol Type:    08-00 IP
```

Bu bir Ethernet_II frame'idir. Type alanının IP veya hexadecimal'de 08-00 (çoğunlukla 0x800 olarak gösterilir) olduğuna dikkat ediniz.

Sonraki frame, aynı alanlara sahiptir. Bu nedenle buda bir Ethernet_II frame'idir:

```
Destination:      ff:ff:ff:ff:ff:ff Ethernet Broadcast
Source:           02:07:01:22:de:a4
Protocol Type:    08-00 IP
```

Bu frame'in bir broadcast olduğunu fark ettiniz mi? Hedef donanım adresi, binary olarak 1'lerden, hexadecimal olarak F'lerden oluştuğundan bunu söyleyebilirsiniz.

Gelin, Ethernet_II frame'ine bir daha göz atalım. Bölüm 13'te IPv6 kullandığımızda, bu örnekten tekrar bahsedeceğim, fakat Ethernet frame'inin, IPv4 routed protokolü ile kullandığımız Ethernet_II frame'iyle aynı olduğunu görebiliyoruz. Fakat type alanı, IPv6 veri taşıdığımızda 0x86dd'dir ve IPv4 veri varken, protokol alanında 0x0800 kullanıyoruz.

```
Destination: IPv6-Neighbor-Discovery_00:01:00:03 (33:33:00:01:00:03)
Source:      Aopen_3e:7f:dd (00:01:80:3e:7f:dd)
Type:        IPv6 (0x86dd)
```

Bu Ethernet_II'nin güzelliğidir. Protokol alanı nedeniyle, herhangi bir Network katmanı routable protokolü kullanabiliyoruz ve Network katmanı protokolünü tespit edebildiğinden, o veriyi taşıyabilmektedir.

## Physical Katmanda Ethernet

Ethernet ilk olarak, DIX (Digital, Intel ve Xerox) olarak bilinen bir grup tarafından gerçekleştirilmiştir. Onlar, ilk Ethernet LAN düzenlemesi olan, IEEE 802.3 komitesini oluşturulması için kullanılan IEEE'yi oluşturmuş ve kurmuşlardır. Bu, koaksiyel kabloda ve daha sonra sarmal-çift ve fiber ortamında çalışan bir 10 Mbps ağ idi.

IEEE,802.3 Komitesini, 802.3u (FastEthernet) ve 802.3ab (kategori5'teki Gigabit Ethernet) olarak bilinen iki yeni komiteye ve son olarak 802.3ae'ye (fiber ve koaksiyel ürerinde 10Gbps) genişletmiştir.

Şekil 1.21, IEEE 802.3 ve orijinal Ethernet Physical katmanı düzenlemelerini göstermektedir.

LAN'ınızı tasarladığınızda, size uygun farklı türde Ethernet ortam aracı olduğunu anlamanız çok önemlidir. Tabi ki, her kullanıcıya Gigabit Ethernet ve switch'ler arasında 10Gbps çalışması mükemmel olurdu. Bu bir gün gerçekleşse de, böyle bir ağın maliyetiyle, bugün oldukça zor olacaktır. Fakat halen uygun olan farklı Ethernet ortam yöntemlerini karşılaştırır ve bir arada kullanırsanız, çok iyi çalışan, uygun maliyetli bir ağ çözümü geliştirebilirsiniz.


***Şekil 1.21:*** *Ethernet Physical katman düzenlemeleri.*

EIA/TIA (Electronic Industries Association and the newer Telecommunications Industry Alliance), Ethernet için Physical katman düzenlemeleri oluşturan standartlar kuruludur. EIA/TIA, Ethernet'in, UTP(unshielded twisted-pair) kablolamada bir 4 5 kablo bağlantı sıralaması ile RJ (registered jack) konnektörü kullandığını belirtir (RJ45). Bununla beraber endüstri artık, sadece 8-pin modüler konnektöre doğru ilerlemektedir.

EIA/TIA tarafından tanımlı her Ethernet kablo çeşidi, kablo boyunca seyahat ederken ve desibel(dB) olarak ölçülen, sinyal gücünün kaybı olarak belirtilen, kendi özelliklerine has sinyal zayıflamasına sahiptir. İşyeri ve evlerde kullanılan kablolama, kategorilerle ölçülmektedir. Daha yüksek-kaliteli kablo, daha yüksek oranda kategoriye ve düşük zayıflamaya sahip olacaktır. Örneğin, kategori5, metrede daha fazla kablo sarmalına sahiptir ve bu nedenle daha az crosstalk olacaktır. Crosstalk, kablodaki bitişik iletken çiftlerde, istenmeyen sinyal karışmasıdır.

Orijinal IEEE 802.3 standartları şunlardır:

**10Base2:** 185 metre uzunluğa kadar, 10Mbps, temel bant teknolojisidir. *Thinnet* olarak bilinir ve tek segmentte 30 iş istasyonunu destekleyebilir. AUI konnektörlerle fiziksel ve mantıksal bir bus(veri yolu) kullanır.10, 10Mbps anlamına gelir, Base, temel bant teknolojisi (ağdaki iletişim için bir sinyalleşme yöntemidir) anlamına gelir ve 2, yaklaşık 200 metreyi belirtir. 10Base2 Ethernet kartları, bir ağa bağlanmak için BNC (British Naval Connector, Bayonet Neill Concelman veya Bayonet Nut Connector) ve T-konnektörleri kullanır.

**10Base5:** 500 metreye kadar, 10Mbps, temel bant teknolojisi. Thicknet olarak bilinir. AUI konnektörlerle fiziksel ve mantıksal bir veri yolu kullanır. Repeater'larla 2,500 metreye kadar çıkabilir ve tüm segmentler için 1,024 kullanıcıyı destekler.

**10BaseT:** Kategori 3 UTP kablolama kullanan 10Mbps'dır. 10Base2 ve 10Base5 ağların tersine, her cihaz bir hub'a ya da switch'e bağlanır ve segment veya kablo başına sadece bir kullanıcıya sahip olabilirsiniz. Fiziksel star topoloji veya mantıksal bus ile RJ45 konnektör kullanır.

802.3 standartlarının her biri, Data Link ortam erişim yönteminden Physical katmana her seferde bir bit transfere izin veren bir Attachment Unit Interface (AUI) belirler. Bu, MAC'in sabit kalmasını sağlar, fakat Physical katman, mevcut ve yeni teknolojileri destekleyebilir anlamına gelir. Orijinal AUI interface'i, 15-pin sarmal-çifte çevrimi sağlayan bir transceiver'a (transmitter/receiver) izin veren 15-pin bir konnektördür.

AUI interface'i, yüksek frekansları içermesinden dolayı 100Mbps Ethernet'i destekleyemez. Bu nedenle, 100BaseT için yeni bir interface ihtiyacı doğdu ve 802.3u şartnamesi, 100Mbps throughput sağlayan Media Independent Interface (MII) olarak adlandırılan bir interface geliştirdi. MII, 4bit olarak tanımlanan nibble kullanır. Gigabit Ethernet, bir Gigabit Media Independent Interface (GMII) kullanır ve tek seferde 8 bit iletir.

802.3u (Fast Ethernet), 802.3 Ethernet ile uyumludur, çünkü aynı fiziksel özellikleri paylaşırlar. Fast Ethernet ile Ethernet, aynı MTU (maximum transmission unit), aynı MAC mekanizmasını kullanır ve 10BaseT Ethernet tarafından kullanılan frame formatını korur. Esasen, Fast Ethernet, 10BaseT'nin 10 katı fazla bir hız önermesi dışında, IEEE 802.3 düzenlemesine bir eklenti olarak geliştirilmiştir.

Genişletilmiş IEEE Ethernet 802.3 standartları şunlardır:

**100BaseTX (IEEE 802.3u):** EIA/TIA kategori 5, 6 veya 7 UTP iki-çift kablolama. Segment başına bir kullanıcı:100 metre mesafe. Fiziksel star topoloji ve mantıksal bir bus ile RJ45 konnektör kullanır.

**100BaseFX (IEEE 802.3u):** 62.5/125-micron multimode fiber ile fiber kablolama kullanır. Noktadan-noktaya topoloji; 412 metreye kadar mesafe. Media-interface konnektörleri olan ST ve SC konnektör kullanır.

**1000BaseCX (IEEE 802.3z):** Sadece 25 metreye kadar çalışabilen twinax (dengelenmiş koaksiyel çift) olarak bilinen bakır sarmal-çifti.

**1000BaseT (IEEE 802.3ab):** Kategori 5, 100 metreye kadar, dört-çift UTP kablolama.

**1000BaseSX (IEEE 802.3z):** 62.5 ve 50-micron damar kullanan MMF; 850 nanometre lazer kullanır ve 62,5 micron ile 220 metreye, 50-micronla 550 metreye kadar erişebilir.

**1000BaseLX:** 9-micron damar, 1300 nano-metre lazer sağlayan ve 3km.'den 10 kilometreye kadar gidebilen single-mode fiber.

NOT *EMI'ye (electromagnetic interference) duyarlı olmayan bir network ortamı gerçekleştirmek istiyorsanız, fiber-optik kablo, yüksek-hızlarda EMI'ye duyarlı olmayan, daha güvenli, uzun-mesafe kablo sağlar.*

# Ethernet Kablolama

Ethernet kablolaması, özellikle de Cisco sınavları almayı planlıyorsanız, önemli bir konudur. Üç tip Ethernet kablosu vardır:

- Straight-through (düz) kablo
- Crossover (çapraz) kablo
- Rollover kablo

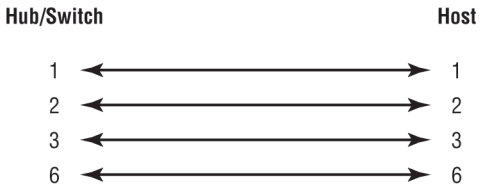
Şimdiki bölümlerde bunlara değineceğiz.

## Straight-Through (Düz) Kablo

Düz kablo, şunları bağlamak için kullanılmaktadır:

- Host'u switch veya hub'a
- Router'ı switch veya hub'a

Dört tel, Ethernet cihazlarını bağlamak için düz kabloda kullanılmaktadır. Bu kablo tipini oluşturmak nispeten kolaydır: Şekil 1.22 düz Ethernet kabloda kullanılan dört teli göstermektedir.



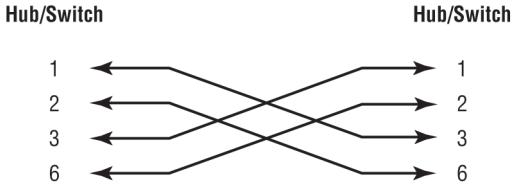

***Şekil 1.22:*** *Düz Ethernet kablo.*

Sadece 1, 2, 3 ve 6 no'lu pinlerin kullanıldığına dikkat edin. 1,1'e, 2, 2'ye, 3, 3'e ve 6, 6'ya bağlayın, anında çalışır ve ağa bağlanır olacaksınız. Ancak, bunun sadece Ethernet'e has kablo olduğunu, ses, Token Ring ISDN ve diğerlerinde çalışmayacağını unutmayın.

## Crossover (çapraz) Kablo

Çapraz kablo şunları bağlamak için kullanılmaktadır:

- Switch'i, switch'e
- Hub'ı, hub'a
- Host'u, host'a
- Hub'ı, switch'e
- Router'ı, direk host'a



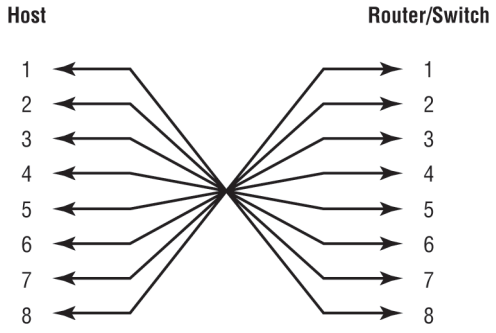

***Şekil 1.23:*** *Çapraz Ethernet kablo.*

Aynı dört tel, düz kabloda olduğu gibi, bu kabloda da kullanılır. Sadece farklı pinleri birbiriyle bağlarız. Şekil 1.23, çapraz bir Ethernet kablosundaki dört telin nasıl kullanıldığını göstermektedir. 1,1'e, 2, 2'ye vs. yerine burada kablonun her iki tarafında da 1, 3'e, 2, 6'ya bağladığımıza dikkat edin.

## Rollover Kablo

Her ne kadar rollover kablo herhangi bir Ethernet bağlantısıyla beraber kullanılmasa da, siz rollover kabloyu, bir host'u, router'ın seri iletişim (com) portuna bağlamak için kullanabilirsiniz.

Şayet Cisco router ya da switch'iniz varsa, Hyperterminal çalışan PC'nizi Cisco donanımına bağlamak için bu kabloyu kullanabilirsiniz. Aynı Ethernet ağ kurulumunda olduğu gibi, hepsi bilgi gönderilmesinde kullanılmasa da, seri cihazların bağlanması için bu kabloda, sekiz tel kullanılmaktadır. Şekil 1.24, bir rollover kablodaki sekiz teli göstermektedir.

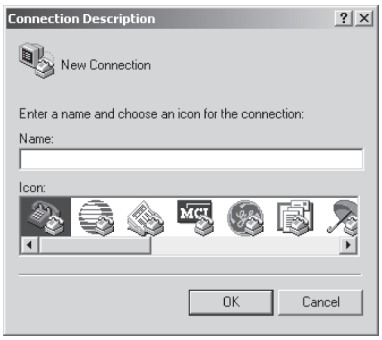



***Şekil 1.24:*** *Rollover Ethernet kablo.*

Bu, muhtemelen, yapması en kolay kablo tipidir. Çünkü sadece düz kablonun bir tarafının ucunu kesiyorsunuz, onu ters sıralıyorsunuz ve tekrar sonlandırıyorsunuz (tabii ki yeni bir konnektör ile).

PC'nizden, Cisco router ya da switch'e bağlı doğru kablonuz varsa, bir konsol bağlantısı oluşturmak ve cihazı yapılandırmak için HyperTerminal'i kullanabilirsiniz. Konfigürasyonu aşağıdaki gibi ayarlayın:

**1.** HyperTerminal'i açın ve bağlantı için bir isim girin. Ne isim verdiğinizin önemi yoktur, fakat ben daima Cisco'yu kullanırım. Sonra OK düğmesini tıklayın.

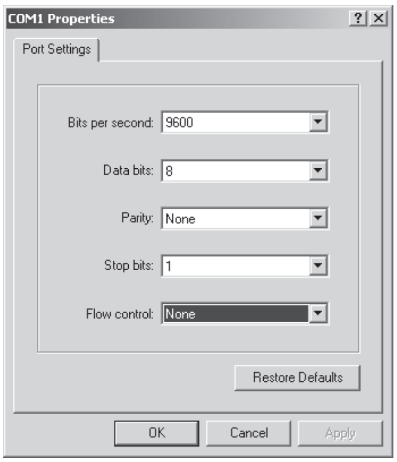



**2.** COM1 veya COM2 gibi, PC'nizde açık olan iletişim portlarını seçin.

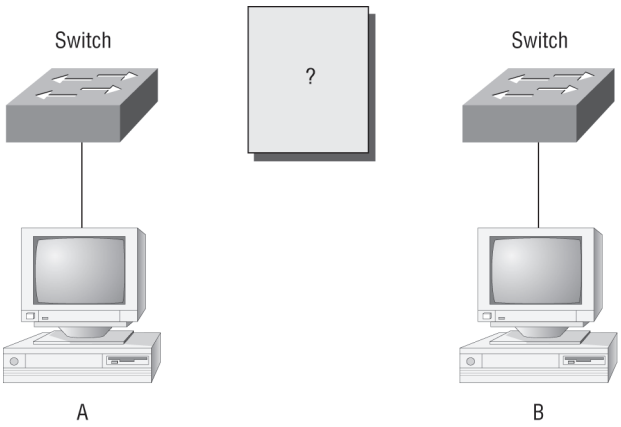

**3.** Şimdi port ayarlarını yapın. Varsayılan değerler (2400bps ve no flow control hardware) çalışmayacaktır. Port ayarlarını, Şekil 1.25'deki gibi yapmalısınız.

Bit hızının 9600'e ve flow kontrol'ün None olarak ayarlandığına dikkat edin. Bu noktada, OK butonuna tıklayabilir ve Enter tuşuna basabilirsiniz, Cisco cihazın konsol portuna bağlı olmalısınız.

Çeşitli RJ45 UTP (unshielded twisted pair) kablolarına göz attık. Bunları aklınızda tutarak, Şekil 1.26'daki switch'ler arasında hangi kablo kullanılmaktadır? Host A'nın host B'ye ping atması için, iki switch'i birbirine çapraz bir kablo ile bağlamanız gerekir. Peki, Şekil 1.27'de görülen ağda hangi tip kablo kullanılmaktadır?

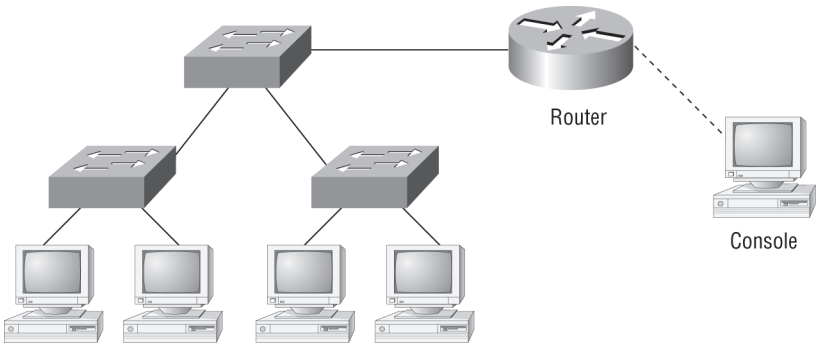



***Şekil 1.25:*** *Bir rollover kablo bağlantısı için port ayarları.*




***Şekil 1.26:*** *RJ45 UTP kablo sorusu.*

Şekil 1.27'de, kullanılan farklı tipte kablolar vardır. Switch'leri bağlamak için, Şekil 1.23'te gördüğümüz gibi, çapraz bir kablo kullanacağız. Sorun, rollover kablo kullanan bir konsol bağlantımız olmasıdır. Artı, host'ların switch'lere doğru iken, router'dan switch'e bağlantı, bir çapraz kablodur. Şayet seri bir bağlantıya sahipsek (ki değiliz), onun, bizi WAN'a bağlayacak bir V.35 olacağını aklınızdan çıkartmayın.




***Şekil 1.27:*** *RJ45 UTP kablo sorusu.*

# Data Enkapsülasyonu

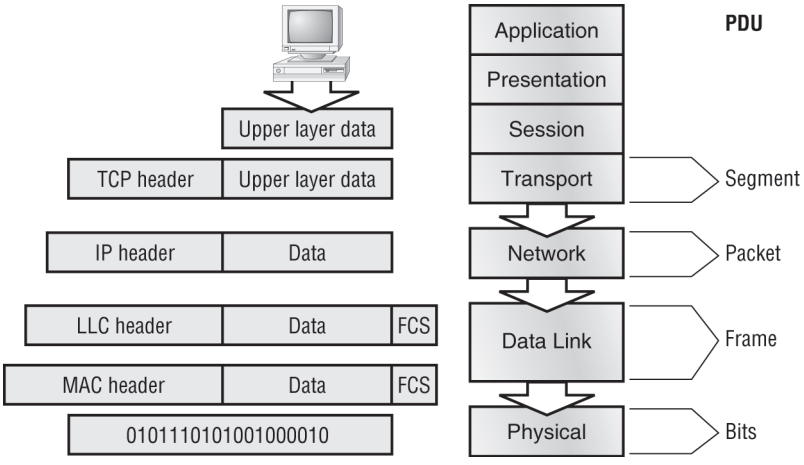
Bir host, bir ağ boyunca başka bir cihaza veri ilettiğinde, veri enkapsüle edilir: Veri, OSI modelinin her katmanındaki protokol bilgisi ile paketlenir. Her katman sadece, alıcı cihazda onun eş düzey katmanıyla iletişim kurar.

İletişim kurmak ve bilgileri değiş tokuş etmek için her katman PDU'lar (Protocol Data Units) kullanır. PDU'lar, OSI modelinin her katmanındaki veriye ekli kontrol bilgileri tutar. Genellikle, veri alanının önündeki başlığa eklenirler, fakat onun sonunda veya kuyruğunda da olabilirler.

Her PDU, OSI modelinin her katmanında enkapsüle ile eklenir ve her biri, her başlıkta sağlanan bilgiye bağlı olarak özel bir isme sahip olur. Bu PDU bilgileri, sadece alıcı cihazdaki eşdüzey katman tarafından okunurlar. Okunduktan sonra bu atılır ve bir sonraki üst katmana sunulur.

Şekil 1.28, PDU'ları ve onların kontrol bilgilerini katmanlara nasıl eklediğini göstermektedir. Bu şekil, üst-katman kullanıcı verisinin, ağda aktarım için nasıl dönüştürüldüğünü göstermektedir. Sonra veri akışı, bir senkron paketi göndererek, alıcı cihaza sanal bir devre oluşturan Transport katmanına gönderilir. Daha sonra, veri akışı, daha ufak parçalara bölünür, bir Transport katman başlığı (bir PDU) oluşturulur ve data alanının başlığına eklenir. Şimdi verinin parçaları bir segment olarak belirtilir. Her segment sıralıdır, böylece veri akışı, alıcı makine tarafında aktarıldığı şekilde tekrar bir araya getirilir.

Her segment bundan sonra, ağ adreslemesi ve ağ topluluğu boyunca route edilmesi için Network katmanına gönderilir. Mantıksal adresleme (örneğin IP) her segmenti doğru ağa koymak için kullanılmaktadır. Network katmanı protokolleri, Transport katmanından gönderilen segmente bir kontrol bilgisi ekler Şimdi sahip olduğumuz, bir *paket* ya da *datagram* olarak isimlendirilir. Transport ve Network katmanının, alıcı makinesinde veri akışını tekrar oluşturmak için beraber çalıştıklarını hatırlayın. Fakat yerel bir network segmentinde




***Şekil 1.28:*** *Data enkapsülasyonu.*

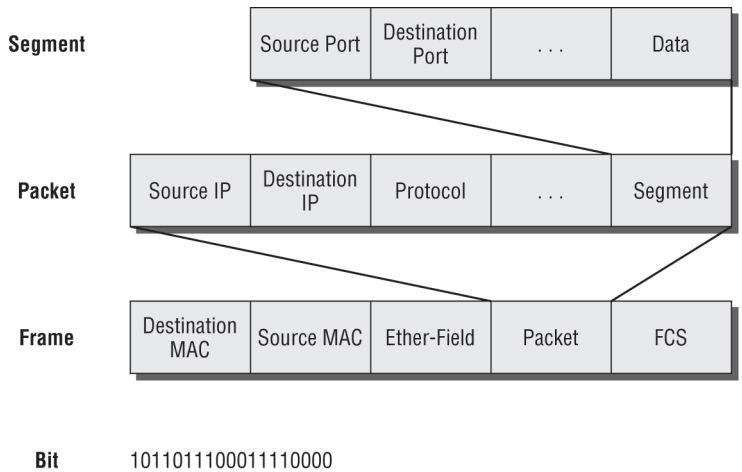
PDU'ları yerleştirmek onun işinin bir parçası değildir. Bu, bir router veya host'a bilgi sağlamanın tek yoludur.

Network katmanından paketleri alıp onları ağ ortamına (kablo veya wireless'e) yerleştirmekten, Data Link katmanı sorumludur. Data Link katmanı, bir frame'deki her paketi enkapsüle eder ve frame başlığı, kaynak ve hedef host'ların donanım adreslerini taşır. Şayet hedef cihaz uzak bir ağda ise, o zaman frame, ağ topluluğu boyunca route edilmesi için bir router'a gönderilir. Hedef bir ağa ulaştığında, paketi hedef makineye ulaştırmak için yeni bir frame kullanılmaktadır.

Bu frame'i ağa koymak için, önce dijital bir sinyale çevrilmesi gerekmektedir. Frame'in, aslında sıfır ve birlerin mantıksal bir grubu olmasından dolayı, Physical katman, bu rakamları, aynı yerel ağdaki cihazlar tarafından okunacak bir dijital sinyale kodlamaktan sorumludur. Alıcı cihazlar dijital sinyalleri senkronize edecek ve dijital sinyalden, sıfır ve birlerin kodunu çözecektir. Bu noktada, cihazlar frame'leri oluşturur, CRC çalıştırır ve sonra kendi cevaplarını frame'in FCS alanındaki cevapla karşılaştırır. Şayet eşleşirse, paket frame'den çekilir ve frame'den kalan kısım atılır. Bu proses, de-encapsulation olarak adlandırılır. Paket, adresin kontrol edildiği Network katmanına gönderilir. Şayet adres eşleşirse, segment paketten çekilir ve paketten geriye kalan kısmı atılır. Segment, veri akışının tekrar oluşturulduğu ve her parçasını alan verici istasyonuna onaylayan Transport katmanında işlemden geçirilir. Daha sonra veri akışını, üst-katman uygulamasına gönderir.

Bir gönderivci cihazda, veri enkapsülasyon şu şekilde çalışır:

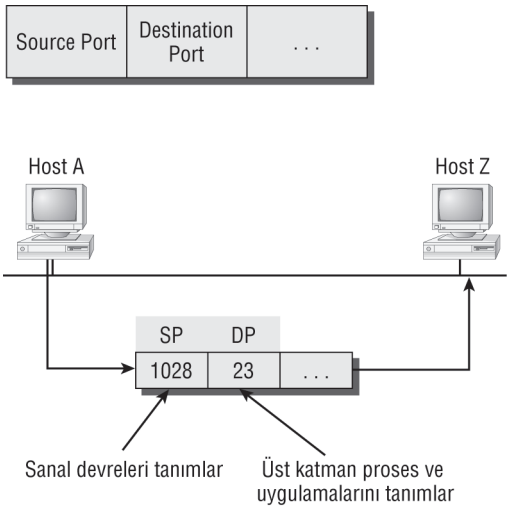
1. Kullanıcı bilgisi, ağda iletim için veriye dönüştürülür.
2. Veri, segment'lere dönüştürülür ve alıcı ile verici makineler arasında güvenli bir bağlantı kurulur.
3. Segmentler, paketlere veya datagram'lara dönüştürülür ve her paketin bir ağ topluluğu boyunca route edilmesi için mantıksal bir adres, başlığa yerleştirilir.
4. Paket ve datagramlar, yerel ağda iletilmek için frame'lere dönüştürülür. Donanım (Ethernet) adresleri, yerel bir network segmentindeki kullanıcı makinelerini eşsiz olarak tanımlamak için kullanılmaktadır.
5. Frame'ler bit'lere dönüştürülür ve dijital kablolama ve saat denetimi kullanılır.
6. Bunu, katman adreslemesi kullanarak daha detaylı açıklamak için Şekil 1.29'u kullanalım.




***Şekil 1.29:*** *PDU ve katman adreslemesi.*

Bir veri akışının, üst katmandan Transport katmanına gönderildiğini hatırlayın. Aslında programcıların problemi olduğundan, teknisyenler gibi veri akışının kimden geldiğiyle ilgilenmeyiz. Bizim işimiz, veri akışının güvenli bir şekilde tekrar oluşturulması ve onun alıcı cihazdaki üst katmana gönderilmesidir.

Şekil 1.29'daki tartışmaları biraz daha ileri götürmeden, port numaralarını ve onları anlayıp anlamadığımızdan emin olalım. Transport katmanı, Şekil 1.30'da görebileceğiniz gibi, sanal devre ve üst-katman proseslerini tanımlamak için port numaralarını kullanır.




***Şekil 1.30:*** *Transport katmanındaki port numaraları.*

Transport katmanı, veri akışını alır, onun segmentlerini seçer ve sanal bir devre kurarak güvenli bir oturum oluşturur. Sonra her segmenti sıraya dizer (numaralar) ve acknowledgment ve akış kontrolü kullanır. Şayet TCP kullanıyorsanız, sanal devre, kaynak port numarası ile tanımlanmaktadır. Host'un bunu, port numarası 1024'ten başlayarak düzenlediğini hatırlayın (0 ile 1023 arası, iyi-bilinen port numaraları için ayrılmıştır). Hedef port numarası, veri akışı, alıcı makinede tekrar oluşturulduğunda, gönderildiği üst-katman prosesini (uygulamayı) tanımlar.

Port numaralarını ve onların Transport katmanında nasıl kullanıldıklarını anladıysanız, Şekil 1.30'a geri dönelim. Transport katmanı başlık bilgisi, veri parçasına eklenince, bir segment olur ve hedef adresi ile birlikte Network katmanına gönderilir. (Hedef IP adresi, veri akışıyla üst katmandan Transport katmanına gönderilmişti ve üst katmandaki bir isim çözümlemesi (muhtemelen DNS) yardımı ile bulunmuştu.)

Network katmanı, bir başlık ilave eder ve her segmentin önüne mantıksal bir adresleme (IP adresleri) ekler. Başlık segmente eklenince, PDU bir paket olarak isimlendirilir. Paket, segmentin geldiği (ya UDP ya da TCP) yeri tanımlayan bir protokol alanına sahiptir. Böylece, verici makineye ulaştığında, segmenti transport katmanındaki doğru protokole gönderebilir.

Network katmanı, paketin yerel ağda nereye gönderileceğini belirleyen, hedef donanım adresinin bulunmasından sorumludur. Bunu, Address Resolution Protocol (ARP) kullanarak yapar (ARP hakkında bölüm 2'de daha detaylı bilgi alacağız). Network katmanındaki IP, hedef IP adresine ve subnet maskına bakar. Şayet, yerel bir ağ isteği çıkarsa, yerel makinenin donanım adresi, bir ARP isteği yoluyla istenir. Şayet paket, uzak bir kullanıcı makinesi için hedeflenirse, IP, varsayılan ağ geçidinin (router) IP adresine bakacaktır.

Yerel host ya da default gateway'in hedef donanım adresi ile birlikte paket, Data Link katmanına gönderilir. Data Link katmanı, paketin önüne bir başlık ekler ve artık veri parçası bir frame olur. (Hem bir başlık hem de bir kuyruğun pakete eklenmesinden dolayı, onu bir frame olarak belirteceğiz) Bu Şekil 1.29'da görülmektedir. Frame, paketin, Network katmanındaki hangi protokolden geldiğini tanımlamak için bir Ether-Type alanı kullanır. Şimdi frame'de bir CRC (cyclic redundancy check) çalışır ve CRC'ye cevap, frame'in kuyruğunda bulunan bir FCS (Frame Check Sequence) alanına yerleştirilir.

Frame şimdi, dijital bir sinyaldeki veriyi kodlamak için, bit zamanlama kuralı kullanacak olan Physical katmana, her seferinde bir bit olarak gönderilmeye hazırdır. Network segmentindeki her cihaz saat ile senkronize olacak ve dijital sinyalden 1 ve 0'ları seçerek, bir frame oluşturacaktır. Frame tekrar oluşturulduktan sonra, frame'in tamam olup olmadığından emin olmak için bir CRC çalıştırılacaktır. Şayet her şey tamamsa, kullanıcı makineleri, frame'in onlar için olup olmadığını anlamak için, hedef adreslerini kontrol edecektir.

Eğer bütün bunlar size karmaşık geliyorsa, Bölüm 6'da, verinin tam olarak nasıl enkapsüle edildiğini ve bir ağ topluluğunda route edildiğini göreceksiniz.

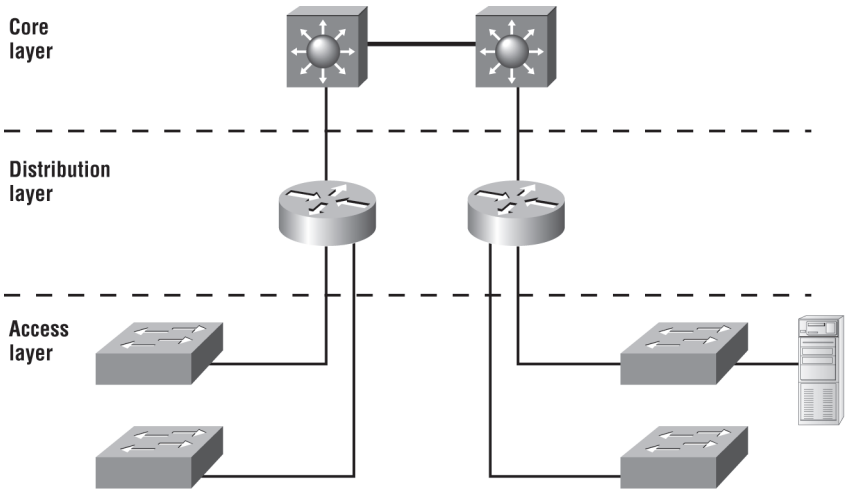
# Cisco Üç-Katmanlı Hiyerarşik Modeli

Çoğumuz hayatımızda daha evvel hiyerarşiye maruz kalmışızdır. Ağabey veya ablalarımızla, hiyerarşinin en altında olmanın ne demek olduğunu öğrendik. Hiyerarşiyi ilk olarak nerede keşfettiğinize bakmaksızın, bugün çoğumuz onu hayatımızın birçok safhasında tecrübe ediyoruz. Bir şeylerin nereye ait olduğunu, hangi fonksiyonların nereye gittiğini anlamamıza yardımcı olan hiyerarşidir. Karmaşık modellere düzen ve anlaşılabilirlik getirir. Örneğin, maaş artışı istiyorsanız, hiyerarşi, sizin altınızda çalışanınıza değil de patronunuza sormayı gerektirir. Bu, sizin isteğinizi onaylayacak ya da reddedecek kişidir. Esasında, hiyerarşiyi anlamak, ihtiyacımız olanı almak için nereye gideceğimizi ayırt etmemize yardımcı olur.

Hiyerarşi, hayatımızın diğer alanlarında olduğu gibi, network tasarımında da aynı avantajların çoğuna sahiptir. Doğru bir şekilde kullanıldığında, ağın daha öngörülebilir olmasını sağlar. Hangi alanların belirli fonksiyonlarla yapılacağını tanımlamakta bize yardımcı olur. Aynı şekilde, hiyerarşik ağlarda belirli seviyelerde access list'ler gibi araçlar kullanabilir ve diğerlerinde onlardan kaçınabilirsiniz.

Gelin onunla yüzleşelim: Geniş ağlar, çoklu protokoller, detaylı konfigürasyonlar ve farklı teknolojilerle oldukça karmaşık olabilir. Hiyerarşi, detayların karmaşık bütünlüğünü anlaşılabilir bir modele özetlememizde bize yardımcı olur. Sonra, belirli konfigürasyonlar gerektikçe, model onların uygulanacağı uygun yollar belirler.

Cisco hiyerarşik modeli, ölçeklenebilir, güvenli, uygun-maliyetli hiyerarşik bir ağ topluluğu tasarlamak, kurmak ve devam ettirmek için size yardımcı olabilir. Cisco, Şekil 1.31'de gösterilen, belirli fonksiyonlarıyla üç katmanlı hiyerarşiyi tanımlar.




***Şekil 1.31:*** *Cisco hiyerarşik modeli.*

Üç katman ve tipik fonksiyonları şunlardır:

- Core katman: omurga
- Distribution (dağıtım) katmanı: routing
- Access(erişim) katmanı: switching

Her katmanın belirli sorumlulukları vardır. Her katmanın, mantıksal olduğunu ve fiziksel cihazlar olmadıklarını hatırlayın. OSI modelindeki diğer mantıksal hiyerarşiyi düşünün. Yedi katman, fonksiyonları açıklar, protokolleri değil. Bazen, OSI modelinin birden fazla katmanına denk gelmektedir ve bazen, çoklu protokoller tek bir katman içinde iletişim kurmaktadır. Aynı şekilde, hiyerarşik ağların fiziksel kurulumunu oluşturduğumuzda, tek katmanda birçok cihaza sahip olabilir veya iki katmanda çalışan tek bir cihaza sahip olabiliriz. Katmanların tanımlaması mantıksaldır, fiziksel değildir.

Şimdi her katmanı daha detaylı inceleyelim.

## Core Katman

Core katman, gerçekten ağın özüdür. Hiyerarşinin en üstünde, core katmanı, çok miktarda trafiğin güvenli ve çabuk iletilmesinden sorumludur. Ağın core katmanının tek amacı, trafiği mümkün olduğu kadar hızlı bir şekilde anahtarlamaktır. Core üzerinden iletilen trafik, kullanıcıların çoğunluğuna mahsustur. Bununla beraber, kullanıcı verisinin, gerektiğinde istekleri core'a gönderen distribution katmanında işlemden geçtiğini hatırlayın.

Şayet core'da bir arıza varsa, her kullanıcı etkilenebilir. Bu nedenle, bu katmanda hata toleransı bir sorundur. Core'un, trafiğin büyük bölümünü görmesi olasıdır, bu nedenle hız ve gecikme burada önem kazanıyor. Core'un belirli fonksiyonlarıyla, şimdi bazı tasarım detaylarını dikkate alabiliriz. Yapmayı istemediğimiz bazı şeylerle başlayalım:

- Trafiği yavaşlatacak hiçbir şey yapmayın. Bu, access list'ler kullanmayı, VLAN'lar arasında routing yapmayı ve paket filtreleme uygulanmasını da içerir.
- Buraya workgroup erişiminine izin vermeyin.
- Ağ topluluğu büyüdüğünde, core'un genişlemesinden kaçının (router'lar eklemek v.s.). Şayet core'da performans problemi olursa, genişleme üzerine güncellemeleri tercih edin.

Core'u tasarlarken, yapmayı istediğimiz bazı şeyler de şunlardır:

- Core'u yüksek güvenilirlikle tasarlayın. FDDI, Fast Ethernet (yedek linklerle beraber) veya ATM gibi hız ve yedekliliği kolaylaştıran data-link teknolojilerini düşünün.
- Hıza önem vererek tasarım yapın. Core, düşük gecikmeye (latency) sahip olmalıdır.
- Düşük convergance zamanlı routing protokollerini seçin. Routing tablonuz kullanılmaz haldeyse, hızlı ve yedekli data-link bağlanırlığı size yardımcı olmaz.

## Distribution (Dağıtım) Katmanı

Distribution katmanı, bazen workgroup katmanı olarak belirtilir ve access ve core katmanları arasındaki iletişim noktasıdır. Distribution katmanının ana fonksiyonları; routing, filtreleme ile WAN erişimi sağlamak ve gerektiğinde, paketlerin core'a nasıl erişeceklerini belirlemektir. Distribution katmanı, örneğin bir dosya isteğinin sunucuya nasıl gönderileceği gibi, ağ servis isteklerinin en hızlı şekilde ele alınmasını belirlemek zorundadır. Distribution katmanının en iyi yolu belirlemesinden sonra, gerekirse, isteği core katmanına iletir. Sonra core katmanı hızlı bir şekilde isteği uygun doğru servise iletir.

Distribution katmanı, ağ için policy'lerin uygulandığı yerdir. Burada, network operasyonlarını tanımlamada oldukça esnek davranabilirsiniz. Genelde distribution katmanında yapılması gereken çeşitli işlemler vardır:

- Routing
- Paket filtreleme, access list'ler gibi araçlar veya kuyruklama uygulamak
- Adres çevrimi ve firewal'lar içeren, güvenlik ve network policy'leri uygulamak
- Statik routing içeren, routing protokolleri arasındaki redistribution
- VLAN'lar arasında routing ve diğer workgroup destek fonksiyonları
- Broadcast ve multicast domain'leri tanımlamak

Distribution katmanında sakınılacak şeyler, sadece diğer katmanlardan birine ait olan fonksiyonlarla sınırlıdır.

### Access (Erişim) Katmanı

Access katmanı, kullanıcı ve çalışma gruplarının ağ topluluğu kaynaklarına erişimini kontrol eder. Access katmanı bazen desktop katmanı olarak belirtilir. Birçok kullanıcının ihtiyaç duyduğu network kaynakları yerel olarak mevcuttur. Distribution katmanı, uzak servisler için tüm trafiği yönetir. Access katmanının içerdiği bazı özellikler şunlardır:

- (Distribution katmanından gelen) Erişim kontrolü ve policy'lerin kullanımını devam ettirmek
- Ayrı collision domain'ler oluşturmak (segmentation)
- Distribution katmanına workgroup bağlanırlığı

DDR ve Ethernet switching gibi teknolojiler, sık sık access katmanında görülür. Statik routing de (dinamik routing yerine) yine burada görülür.

Önceden de belirttiğimiz gibi üç ayrı katman, üç ayrı router demek değildir. Daha azda olabilir daha çokta. Bunun bir katmanlı yaklaşım olduğunu hatırlayın.

## Özet

Bu bölüm bitmeyecek gibi görünüyordu, fakat bitti. Şimdi, çok fazla temel bilgiye sahipsiniz; onun üzerine inşa etmeye hazırsınız ve sertifikasyona doğru ilerliyorsunuz.

Basit, temel network kurulumu ve collision ile broadcast domain'leri arasındaki farklılıklarından bahsederek başladık. Ayrıca, bir ağ topluluğunda kullanılan çeşitli cihazları da konuştuk.

Sonra, uygulama geliştiricilere, her tür sistem ve ağda çalışabilen uygulamaları tasarlamada yardımcı olan, yedi-katmanlı modeli kullanan OSI'yi gördük. Her katman, kendine özel görevlere sahiptir ve sağlam, etkin iletişimi temin etmek için, modeldeki sorumluluğunu seçer. Her katmanın detaylarını öğrendiniz ve Cisco'nun, OSI modeli düzenlemelerine nasıl baktığını gördünüz.

İlave olarak, OSI modelindeki her katman, farklı cihaz tiplerini belirtir. Her katmanda kullanılan farklı cihaz, kablo ve konnektörleri açıkladım. Hub'ların Physical katman cihazları olduğunu ve dijital sinyalleri, aldığı dışında tüm segmentlere gönderdiğini hatırlayın. Switch'ler, donanım adresi kullanarak ağı segmentlere böler ve collision domain'leri ayırır. Router'lar, broadcast domain'leri (ve collision domain'leri) oluşturur ve bir ağ topluluğu boyunca paketleri göndermek için mantıksal adresleme kullanır.

Son olarak, bu bölüm, Cisco üç-katman hiyerarşik modelini inceledi. Üç katmanın detaylarını ve her birinin bir Cisco ağ topluluğu tasarlamaya ve uygulamaya yardımcı olmak için nasıl kullanıldığını açıkladım. Bir sonraki bölümde IP adreslemesine bakacağız.

## Sınav Gereklilikleri

**LAN'da trafik tıkanıklığının olası sebeplerini hatırlamak:** Bir broadcast domain'inde çok sayıda host olması, broadcast fırtınası, multicasting ve düşük bant genişliği, LAN'daki trafik tıkanıklığının olası sebeplerindendir.

**Bir collision domain ile broadcast domain arasındaki farkı anlamak:** Collision domain, belirli bir cihazın paket gönderdiği ve aynı segmentteki diğer cihazların ona dikkat etmesinin zorunlu olduğu network segmentindeki cihazlardan oluşan bir ağ derlemesini açıklamakta kullanılan terimdir. Bir broadcast domain'inde, bir network segmentindeki cihazların tümü, bu segmentte gönderilen tüm broadcast'leri duyarlar.

**Bir hub, bridge, switch ve router arasındaki farkları anlamak:** Hub'lar, sadece bir collision ve bir broadcast domain oluşturur. Bridge'ler, collision domain'lerini ayırır fakat geniş bir broadcast domain'i oluştururlar. Ağı filtrelemek için donanım adresi kullanırlar. Switch'ler, aslında daha akıllı, çok portlu bridge'lerdir. Router'lar, broadcast domain'lerini (ve collision domain'lerini) ayırırlar ve ağı filtrelemek için mantıksal adresleme kullanır.

**Connection-oriented ve connectionless network servisleri arasındaki farkı hatırlamak:** Connection-oriented servisler, güvenli bir oturum oluşturmak için acknowledgment'lar ve akış kontrolü kullanır. Connectionless network servislerinden daha fazla ek yük kullanılmaktadır. Connectionless servisleri veriyi, acknowledgement ve akış kontrolü olmaksızın gönderir. Bu güvensiz sayılmaktadır.

**OSI katmanlarını hatırlamak:** OSI modelinin yedi katmanını ve her katmanın hangi fonksiyonları sağladığını hatırlamak zorundasınız. Application, Presentation ve Session katmanları üst katmanlardır ve kullanıcı arayüzünden bir uygulamaya haberleşmeden sorumludurlar. Transport katmanı, segmentation, sequencing (sıralama) ve sanal devreler sağlar. Network katmanı, bir ağ topluluğu boyunca mantıksal network adreslemesi ve routing sağlar. Data Link katmanı, framing ve verinin ağ ortamına yerleştirilmesini sağlar. Physical katman, 1 ve 0'ları almak ve onları network segmentinde iletim için dijital sinyale kodlamaktan sorumludur.

**Ethernet kablolamasını ve onları ne zaman kullanacağınızı hatırlamak:** Bir Ethernet kablodan oluşturulabilen üç tip kablo; straight-through (düz) (PC'nin ya da router'ın Ethernet interface'ini bir hub'a ya da switch'e bağlamak için), crossover (çapraz) (hub'ı hub'a, hub'ı switch'e, switch'i switch'e veya PC'yi PC'ye) ve rollover (PC'den bir router ya da switch'e konsol bağlantısı için).

**Bir PC'den router'a konsol kablosunun nasıl bağlandığını ve HyperTerminal'in nasıl başladığını hatırlamak:** Bir rollover kablo alın ve host'un COM portundan, router'ın konsol portuna bağlayın. HyperTerminal'i başlatın ve BPS'i 9600'e ve flow control'ü None olarak ayarlayın.

**Cisco üç-katman modelindeki üç katmanı hatırlamak:** Cisco hiyerarşik modelindeki üç katman; core, distribution ve access katmanlarıdır.

# Yazılı Lab 1

Bu kısımda, onları içeren kavram ve bilgilere sahip olduğunuzdan emin olmak için aşağıdaki lab'ları tamamlayacaksınız:

- Lab 1.1: OSI Soruları
- Lab 1.2: OSI Katman ve Cihazlarını tanımlamak
- Lab 1.3: Collision and Broadcast Domain'lerini tespit etmek
- Lab 1.4: Binary/Decimal/Hexadecimal Dönüşüm

(Yazılı lab'ların cevaplarını, bu bölüm için gözden geçirme sorularının cevaplarından sonra bulabilirsiniz)

## Yazılı Lab 1.1: OSI Soruları

OSI modeli ile ilgili aşağıdaki soruları cevaplayın:

1. Hangi katman, bağlantı kurmak için gerekli kaynaklarla birlikte partnerlerin iletişiminin uygunluğunu seçip belirler, partnerlik uygulamalarını koordine eder ve hata giderme ile veri bütünlüğünü kontrol için prosedürlerde bir görüş birliği oluşturur?
2. Data Link katmanından gelen veri paketlerinin, elektrik sinyallerine dönüştürülmesinden hangi katman sorumludur?
3. Hangi katmanda, iki uç sistem arasında yol seçimi ve bağlantıları mümkün kılarak, routing gerçekleşmektedir?
4. Verinin nasıl formatlandığını, sunulduğunu, kodlandığını ve dönüştürüldüğünü hangi katman tanımlar?
5. Uygulamalar arasındaki oturumların oluşturulması, yönetilmesi ve sonlandırılmasından hangi katman sorumludur?

6. Hangi katman, fiziksel bir link boyunca verinin güvenli iletimini garantiye alır ve fiziksel adresleme, hat disiplini, network topolojisi, hata uyarısı, frame'in sıralı teslimi ve akış kontrolünden öncelikle ilgilidir?
7. Hangi katman network boyunca uç düğümler arasında güvenli iletişim için kullanılır? Ve sanal devreleri kurmak, devam ettirmek ile sonlandırmak için mekanizmalar, transport-hata algılama, giderme ile bilgi akışının kontrolünü sağlar?
8. Hangi katman, path determination için router'ların kullandığı mantıksal adreslemeyi sağlar?
9. Hangi katman voltajı, kablo hızını, kabloların pinlerini ve cihazlar arasındaki bit'lerin hareketini belirler?
10. Hangi katman, bit'leri byte'lara, byte'ları frame'lere birleştirir, MAC adreslemesi kullanır ve hata tespiti sağlar?
11. Ağda, farklı uygulamalardan gelen veriyi ayrı tutmaktan hangi katman sorumludur?
12. Hangi katman frame'lerle ifade edilir?
13. Hangi katman segmentlerle ifade edilir?
14. Hangi katman paketlerle ifade edilir?
15. Hangi katman bit'lerle ifade edilir?
16. Aşağıdakileri enkapsülasyon sırasına koyun:
    - Paketler
    - Frame'ler
    - Bit'ler
    - Segmentler
17. Hangi katman veriyi segment haline dönüştürür?
18. Hangi katman, verinin fiziksel aktarımını sağlar ve hata uyarısı, network topolojisi ile akış kontrolünü ele alır?
19. Hangi katman, cihaz adreslemesini yönetir, ağdaki cihazların izlerini sürer ve veriyi taşımak için en iyi yöntemi belirler?
20. Bir MAC adresinin bit uzunluğu ve ifade şekli nedir?

## Yazılı Lab 1.2: OSI Katmanı ve Cihazlarını Tanımlamak

Boşlukları, uygun OSI katmanı, hub, switch ya da router cihazlarıyla doldurun.

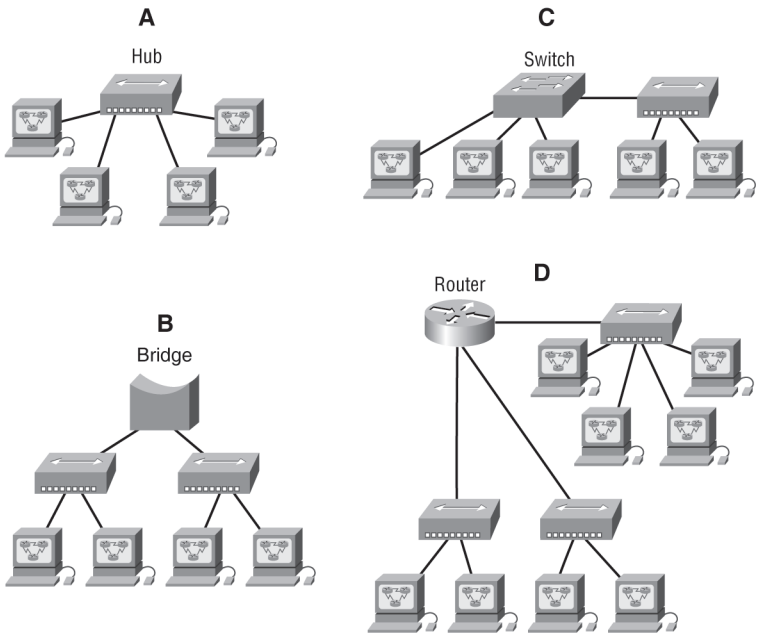
| Açıklama | Cihaz veya OSI katmanı |
|---|---|
| Bu cihaz, Network katmanı hakkında bilgileri alır ve gönderir. | |
| Bu katman, iki uç istasyon arasında aktarımdan önce sanal bir devre oluşturur. | |
| Bu katman, servis erişim noktaları kullanır. | |
| Bu cihaz, bir ağı filtrelemek için donanım adresi kullanır. | |
| Ethernet, bu katmanda tanımlanmıştır. | |

| Açıklama | Cihaz veya OSI katmanı |
|---|---|
| Bu katman, akış kontrolü ve sequencing'i destekler | |
| Bu cihaz, uzak bir ağa mesafeyi ölçebilir. | |
| Bu katmanda mantıksal adresleme kullanılır. | |
| Donanım adresleri bu katmanda tanımlanır. | |
| Bu cihaz, büyük bir collision ve büyük bir broadcast domain oluşturur. | |
| Bu cihaz, birçok küçük collision domain'i oluşturur, fakat ağ hala geniş bir broadcast domain'idir. | |
| Bu cihaz, asla full duplex çalışamaz. | |
| Bu cihaz, collision ve broadcast domain'lerini ayırır. | |

## Yazılı Lab: 1.3 Collision ve Broadcast Domain'lerinin Tespit Edilmesi

Aşağıdaki grafikte, tüm belirli cihazlardaki collision ve broadcast domain sayılarını tespit edin. Her cihaz bir harf ile belirtilmiştir:

A. Hub
B. Bridge
C. Switch
D. Router



## Yazılı Lab 1.4: Binary/Decimal/Hexadecimal Dönüşümü

1. Decimal adresleri binary formatına çevirin.

   192.168.10.15'i binary formatında ifade etmek için aşağıdaki tabloyu tamamlayın.

| 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 | Binary |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |

172.16.20.55'i binary formatında ifade etmek için aşağıdaki tabloyu tamamlayın.

| 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 | Binary |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |

10.11.1299'u binary formatında ifade etmek için aşağıdaki tabloyu tamamlayın.

| 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 | Binary |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |

2. Aşağıdakileri binary formattan decimal IP adresine çevirin.

   11001100.00110011.10101010.01010101'i decimal IP adres formatında ifade etmek için aşağıdaki tabloyu tamamlayın.

| 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 | Decimal |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |

11000110.11010011.00111001.11010001'i decimal IP adres formatında ifade etmek için aşağıdaki tabloyu tamamlayın.

| 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 | Decimal |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |

10000100.11010010.10111000.10100110'ı decimal IP adres formatında ifade etmek için aşağıdaki tabloyu tamamlayın.

| 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 | Decimal |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |

3. Aşağıdakileri binary formatından hexadecimal'e çevirin

11011000.00011011.00111101.01110110 hexadecimal'de ifade etmek için aşağıdaki tabloyu tamamlayın.

| 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 | Hexa-decimal |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |

11001010.11110101.10000011.11101011 hexadecimal'de ifade etmek için aşağıdaki tabloyu tamamlayın.

| 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 | Hexa-decimal |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |

10000100.11010010.01000011.10110011 hexadecimal'de ifade etmek için aşağıdaki tabloyu tamamlayın.

| 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 | Hexa-decimal |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |

# Gözden Geçirme Soruları

NOT *Aşağıdaki sorular bu modülün materyallerini anladığınızı test etmek için hazırlanmıştır. İlave sorulara nasıl sahip olacağınızla ilgili bilgi için bu kitabın Giriş bölümüne bakınız*

1. Alıcı bir makine, onaylanması gereken segmentlerin hepsini almakta başarısız olmuştur. Makine, bu iletişim oturumunun güvenliğini artırmak için ne yapabilir?
   A. Farklı bir kaynak port numarası gönderir.
   B. Sanal devreyi tekrar başlatır.
   C. Sequence (sıra) numarasını azaltır.
   D. Window boyutunu azaltır.
2. Hangi alanlar bir IEEE Ethernet frame başlığında bulunur? (İki şık seçin.)
   A. Source ve destination MAC adresi
   B. Source ve destination network adresi
   C. Source ve destination MAC adresi ve source ve destination network
   D. adresi
   E. FCS alanı
3. Hangi katman 1 cihazı, tek bir LAN segmenti ile çevrili bir alanı genişletmek için kullanılabilir?
   A. Switch
   B. NIC
   C. Hub
   D. Repeater
   E. RJ45 transceiver
4. Bir veri akışı segmentasyonu, OSI modelinin hangi katmanında olur?
   A. Fiziksel
   B. Data Link
   C. Network
   D. Transport

5. Aşağıdakilerden hangileri router fonksiyonlarını açıklamaktadır? (Dört şık seçin.)
   A. Paket switching
   B. Collision engelleme
   C. Paket filtreleme
   D. Broadcast domain genişletme
   E. Ağ toplulukları iletişimi
   F. Broadcast gönderme
   G. Yol seçimi
6. Router'lar, katman __ 'te çalışır. LAN switch'leri, katman __ 'te çalışır. Ethernet hub'ları, katman __'te çalışır. Kelime işlenmesi, katman __ 'te çalışır.
   A. 3, 3, 1, 7
   B. 3, 2, 1, hiçbiri
   C. 3, 2, 1, 7
   D. 2, 3, 1, 7
   E. 3, 3, 2, hiçbiri
7. Data ne zaman enkapsüle edilir, doğru sıralama hangisidir?
   A. Data, frame, paket, segment, bit
   B. Segment, data, paket, frame, bit
   C. Data, segment, paket, frame, bit
   D. Data, segment, frame, paket, bit
8. Veri iletişim endüstrisi neden OSI referans modelini kullanır? (İki şık seçin.)
   A. Network iletişim proseslerini daha küçük ve basit bileşenlere böler, böylece bileşenlerin gelişimi, tasarımı ve hata giderimine yardımcı olur.
   B. Farklı üretici ekipmanlarının aynı elektronik bileşenleri kullanmasını mümkün kılar, böylece kaynak ve araştırma geliştirme fonlarından kazanç sağlanır.
   C. Çoklu rekabet standartlarının gelişmesini destekler ve böylece ekipman üreticilerine iş olanakları sağlar.
   D. Modelin her katmanında hangi fonksiyonların olduğunu tanımlayarak endüstri standardizasyonunu teşvik eder.
   E. Bir katmandaki işlevsellikteki değişikliğin diğer katmanlarda değişiklikler gerektirdiği bir yapı sağlar.
9. Bir bridge'le segmentasyonun iki amacı nedir?
   A. Daha fazla broadcast domain'i eklemek.
   B. Daha fazla collision domain oluşturmak.
   C. Kullanıcılar için daha fazla bant genişliği sağlamak.
   D. Kullanıcılar için daha fazla broadcast'e izin vermek.
10. Aşağıdakilerden hangileri, full-duplex ile karşılaştırıldığında, half-duplex'in benzersiz özelliklerindendir?
    A. Half-duplex Ethernet, paylaşımlı bir collision domain'de çalışır.
    B. Half-duplex Ethernet, özel bir collision domain'de çalışır.

C. Half-duplex Ethernet, daha randımanlı throughput'a sahiptir.

D. Half-duplex Ethernet, daha düşük throughput'a sahiptir.

E. Half-duplex Ethernet, özel bir broadcast domain'de çalışır.

11. EMI'ye duyarlı olmayan bir network ortam aracı kullanmak istiyorsunuz. Hangi tip kablolama kullanmak zorundasınız?

A. Thicknet koaksiyel

B. Thinnet koaksiyel

C. Kategori 5 UTP kablo

D. Fiber-optik kablo

12. Acknowledgement, sequencing ve akış kontrolü hangi OSI katmanının özelliklerindendir?

A. Katman 2

B. Katman 3

C. Katman 4

D. Katman 7

13. Aşağıdakilerden hangileri akış kontrolü türlerindendir? (Doğru olan tüm seçenekleri işaretleyin.)

A. Buffering

B. Cut-through

C. Windowing

D. Tıkanıklık (congestion) engelleme

14. Aşağıdaki bağlantı tiplerinden hangisi full-duplex kullanabilir? (Üç şık seçin.)

A. Hub'tan hub'a

B. Switch'ten switch'e

C. Host'tan host'a

D. Switch'ten host'a

15. Akış kontrolünün amacı nedir?

A. Bir acknowledgment alınmadıysa, verinin tekrar aktarıldığından emin olmak.

B. Hedef cihazda segmentleri doğru sıra ile bir araya getirmek.

C. Verici tarafından gönderilen veri miktarını yönetmek için alıcıya bir yöntem sağlamak.

D. Her segmentin boyutunu düzenlemek.

16. Full-duplex bir network operasyonu ile ilgili hangi üç ifade doğrudur?

A. Full-duplex'te collision'lar yoktur.

B. Her full-duplex düğümü için tanımlı bir switch portu gerekmektedir.

C. Ethernet hub portları, full-duplex için yeniden yapılandırılırlar.

D. Bir full-duplex ortamda, host'un network kartı, aktarımdan önce network ortam aracının uygunluğunu kontrol etmelidir.

E. Host'un network kartı ve switch portu, full-duplex modda çalışma kapasitesinde olmalıdır.

17. Switch'ler arasında hangi tip RJ45 UTP kablo kullanılır?

    A. Düz

    B. Çapraz

    C. CSU/DSU ile çapraz kablo

    D. İki switch arasındaki bir router ile çapraz kablo

18. Ethernet LAN'ındaki bir kullanıcı, bir collision olduktan sonra, ne zaman ileteceğini nasıl bilir? (İki şık seçin.)

    A. Bir CSMA/CD collision domain'inde, birçok istasyon veriyi eşzamanlı olarak başarılı bir şekilde iletebilir.

    B. Bir CSMA/CD collision domain'inde, istasyonlar, aktarımdan önce, ortam aracının kullanılmadığı ana kadar beklemek zorundadırlar.

    C. Daha fazla hub ekleyerek CSMA/CD ağını geliştirebilirsiniz.

    D. Bir collision'dan sonra, collision'ı tespit eden istasyon, kayıp veriyi göndermek için ilk önceliğe sahiptir.

    E. Bir collision'dan sonra, tüm istasyonlar, rastgele bir backoff algoritması çalıştırır. Backoff gecikme periyodu dolduğunda, tüm istasyonlar veriyi iletmek için eşit önceliğe sahiptir.

    F. Bir collision'dan sonra, ilgili tüm istasyonlar aynı backoff algoritmasını çalıştırır ve sonra, veriyi aktarmadan önce birbirlerini senkronize ederler.

19. PC'nin COM portunu bir router ya da switch konsol portuna bağlamak için hangi tip RJ45 UTP kablo kullanırsınız?

    A. Düz

    B. Çapraz

    C. Bir CSU/DSU ile çapraz kablo

    D. Rollover kablo

20. Aşağıdaki binary numaraya sahipsiniz.

    **10110111**

    Ondalık (decimal) ve hexadecimal karşılıkları nelerdir?

    A. 69 / 0x2102

    B. 183 / B7

    C. 173 / A6

    D. 83 / 0xC5

# Gözden Geçirme Sorularının Cevapları

1. D Bir alıcı makinesi akış kontrolü kullanarak vericiyi kontrol edebilir (TCP, varsayılan olarak Windowing kullanır). Window boyutunu azaltarak, alıcı makinesi aktarımı yavaşlatabilir, böylece alıcı makinenin bellekleri taşmaz.
2. A,D Bir Ethernet frame'i, source ve destination MAC adresleri, Network katman protokolünü belirlemek için bir Ether-Type alanı, data ve CRC'ye cevabı tutan FCS alanına sahiptir.
3. C,D Tek bir collision domain'ini genişletmeyi gerçekten istemezsiniz, fakat bir hub (çok portlu repeater), sizin için bunu sağlayacaktır.
4. D Transport katmanı, üst katmandan data akımlarını alır ve bunları, segment denilen daha küçük parçalara ayırır.
5. A, C, E, G Router'lar, paket switching (anahtarlama), paket filtreleme, ağ topluluğu iletişimi ve yol seçimi sağlar.
6. B Router'lar katman 3 'te çalışırlar. LAN switch'ler katman2 'de çalışırlar. Ethernet hub'ları, katman 1 'de çalışır. Kelime işleme uygulamaları, application katman interfacelerine iletilir, fakat katman 7'de çalışmazlar, bu nedenle cevap Hiçbiri olmalıdır.
7. C Enkapsülasyon yöntemi, data, segment, paket, frame, bit'tir.
8. A, D Katmanlı modelin asıl avantajı, uygulama geliştiricilere, katman modeli düzenlemelerinin sadece bir katmanının bakış açılarını değiştirme olanağı sağlamasıdır. OSI katmanlı modeli kullanmanın avantajları şunları içerir: network iletişim proseslerini daha küçük ve basit bileşenlere böler, böylece bileşen gelişimi, tasarımı ve hata giderimine yardımcı olur; network bileşenlerinin standardizasyonuyla çoklu-üretici gelişimine izin verir; modelin her katmanında hangi fonksiyonların olduğunu tanımlayarak endüstri standardizasyonunu teşvik eder; iletişim için birçok network donanım ve yazılım türüne izin verir ve bir katmandaki değişikliklerin, diğer katmanlardakileri etkilemesini engeller, böylece gelişmeyi engellemez.
9. B, C Bridge'ler, collision domain'lerini ayırarak kullanıcılara daha fazla bant genişliği sağlar.
10. A, D Full-duplex'in tersine, half-duplex Ethernet, paylaşımlı bir collision domain'inde çalışır ve full-duplex'den daha düşük verimli bir throughput'a sahiptir.
11. D Fiber-optik kablo, yüksek hızlarda EMI karışmasına duyarlı olmayan, daha güvenli, uzak-mesafe kablo sağlar.
12. C Güvenli bir transport katmanı bağlantısı, tüm verinin aktarıldığından ve güvenli bir şekilde alındığından emin olmak için acknowledgment kullanır. Güvenli bir bağlantı, transport katmanının (katman4) özellikleri olan acknowledgment, sequencing ve akış kontrolü kullanan sanal bir devre ile tanımlanır.
13. A, C, D Yaygın akış kontrolü tipleri, buffering (bellekleme), windowing ve congestion (tıkanıklık) engellemedir.
14. B, C, E Hub'lar full-duplex Ethernet çalışamazlar. Full-duplex, full-duplex çalışma kapasitesindeki iki cihaz arasında noktadan-noktaya bir bağlantıda kullanılmalıdır. Switch'ler ve host'lar, birbirleri arasında full-dupplex çalışabilir, fakat hub asla full-duplex çalışamaz.
15. C Akış kontrolü, alıcı cihaza, aktarıcıyı kontrol etme izni verir, böylece alıcı cihazın belleği taşmaz.
16. A, B, E Full-duplex, kablo çiftlerini, veri göndermek ve almak için eşzamanlı olarak kullanıyorsunuz demektir. Collision olmayacağı anlamında, her düğüm için tanımlı bir switch portuna sahip olmalısınız. Hem host'un network kartı hem de switch, full-duplex'te çalışma kapasitesinde ve ayarlı olmalıdır.
17. B İki switch'i birbirine bağlamak için, RJ45 UTP çapraz kablosu kullanırsınız.

18. B, E Bir Ethernet segmentindeki verici istasyonlar, bir collision fark edince, tüm istasyonların collision'ın farkına vardıklarından emin olmak için bir jam sinyali yayarlar.
19. D Bir router ya da switch konsol portuna bağlamak için, bir RJ45 UTP rollover kablo kullanabilirsiniz.
20. B Bir binary numara alıp onu decimal (ondalık) ve hexadecimal'e çevirmek zorunda olabilirsiniz. Ondalık sayıya çevirmek için kendi değerlerini kullanarak 1'leri toplayın. 10110111 binary numaranın çevrilmiş değeri, 128+32+16+4+2+1=183'tür. Hexadecimal karşılığını bulmak için, sekiz binary sayıyı,1011 ve 0111 şeklinde nibble'a (4 bit'e) bölün. Bu değerleri toplayarak, 11 ve 7'e ulaşırsınız. Hexadecimal'de, 11, B olduğundan cevap 0xB7'dir.

# Yazılı Lab 1 Cevapları

1. Application katman, bir sunucudan broadcast edilen network kaynaklarını bulmaktan ve akış kontrolü ile hata kontrolünden sorumludur (şayet uygulama geliştiriciler seçerlerse).
2. Physical katman, Data Link katmanından frame'leri alır ve network ortam aracında aktarım için 1 ve 0'ları dijital bir sinyale kodlar.
3. Network katmanı, bir ağ topluluğu boyunca routing ve mantıksal adresleme sağlar.
4. Presentation katman, verinin, Application katman için okunabilir bir formatta olduğundan emin olur.
5. Session katman, uygulamalar arasındaki oturumları kurar, devam ettirir ve sonlandırır.
6. Data Link katmanındaki PDU'lar frame olarak bilinirler. Bir soruda frame görür görmez, cevabı biliyorsunuzdur.
7. Transport katmanı, iki host arasında güvenli bir bağlantı oluşturmak için sanal devreler kullanır.
8. Network katmanı, mantıksal adresleme (tipik olarak IP) ve routing sağlar.
9. Physical katman, cihazlar arasında elektriksel ve mekanik bağlantılardan sorumludur.
10. Data Link katmanı, data paketlerinin frame'lenmesinden sorumludur.
11. Session katman, farklı kullanıcı uygulamaları arasında oturumlar oluşturmaktan sorumludur.
12. Data Link katmanı, Network katmanından alınan paketleri frame'ler.
13. Transport katmanı, kullanıcı verilerini segmentlere ayırır.
14. Network katmanı, Transport katmanından gelen segmentlerden, paketler oluşturur.
15. Physical katman, bir dijital sinyalde 1 ve 0'ları taşımaktan sorumludur.
16. Segmentler, paketler, frame'ler, bit'ler.
17. Transport.
18. Data Link.
19. Network.
20. Hexadecimal olarak ifade edilen 48 bit (6 byte).

# Yazılı Lab 1.2 Cevabları

| Açıklama | Cihaz veya OSI katmanı |
|---|---|
| Bu cihaz, Network katmanı hakkında bilgileri alır ve gönderir. | Router |
| Bu katman, iki uç istasyon arasında aktarımdan önce sanal bir devre oluşturur. | Transport |
| Bu katman, servis erişim noktaları kullanır. | Data Link (LLC alt katmanı) |
| Bu cihaz, bir ağı filtrelemek için donanım adresi kullanır. | Bridge ve switch |
| Ethernet, bu katmanda tanımlanmıştır. | Data Link ve Fiziksel |
| Bu katman, akış kontrolü ve sequencing'i destekler. | Transport |
| Bu cihaz, uzak bir ağa mesafeyi ölçebilir. | Router |
| Bu katmanda mantıksal adresleme kullanılır. | Network |
| Donanım adresleri bu katmanda tanımlanır. | Data Link (MAC alt katmanı) |
| Bu cihaz, büyük bir collision ve büyük bir broadcast domain oluşturur. | Hub |
| Bu cihaz, birçok küçük collision domain'i oluşturur, fakat ağ hala geniş bir broadcast domain'idir. | Switch ve bridge |
| Bu cihaz, asla full duplex çalışamaz. | Hub |
| Bu cihaz, collision ve broadcast domain'lerini ayırır. | Router |

# Yazılı Lab 1.3 Cevaplar

1. Hub: Bir collision domain, bir broadcast domain.
2. Bridge: İki collision domain, bir broadcast domain.
3. Switch: Dört collision domain, bir broadcast domain.
4. Router: Üç collision domain, üç broadcast domain.

# Yazılı Lab 1.4 Cevaplar

1. Decimal adresleri binary formatına çevirin.

   192.168.10.15'i binary formatında ifade etmek için aşağıdaki tabloyu tamamlayın.

| Decimal | 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 | Binary |
|---|---|---|---|---|---|---|---|---|---|
| 192 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 11000000 |
| 168 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 10101000 |
| 10 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 00001010 |
| 15 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 00001111 |

172.16.20.55'i binary formatında ifade etmek için aşağıdaki tabloyu tamamlayın.

| Decimal | 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 | Binary |
|---|---|---|---|---|---|---|---|---|---|
| 172ˇ | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 10101100 |
| 16 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 00010000 |
| 20 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 00010100 |
| 55 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 00110111 |

10.11.12.99'u binary formatında ifade etmek için aşağıdaki tabloyu tamamlayın.

| Decimal | 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 | Binary |
|---|---|---|---|---|---|---|---|---|---|
| 10 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 00001010 |
| 11 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 00001011 |
| 12 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 00001100 |
| 99 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 01100011 |

2. Aşağıdakileri binary formattan decimal IP adresine çevirin.

11001100.00110011.10101010.01010101'i ondalık IP adres formatında ifade etmek için aşağıdaki tabloyu tamamlayın.

| Binary | 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 | Decimal |
|---|---|---|---|---|---|---|---|---|---|
| 11001100 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 204 |
| 00110011 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 51 |
| 10101010 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 170 |
| 01010101 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 85 |

11000110.11010011.00111001.11010001'i ondalık IP adres formatında ifade etmek için aşağıdaki tabloyu tamamlayın.

| Binary | 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 | Decimal |
|---|---|---|---|---|---|---|---|---|---|
| 11000110 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 198 |
| 11010011 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 211 |
| 00111001 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 57 |
| 11010001 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 209 |

10000100.11010010.10111000.10100110'i ondalık IP adres formatında ifade etmek için aşağıdaki tabloyu tamamlayın.

| Binary | 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 | Decimal |
|---|---|---|---|---|---|---|---|---|---|
| 10000100 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 132 |
| 11010010 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 210 |
| 10111000 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 184 |
| 10100110 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 166 |

3. Aşağıdakileri binary formatından hexadecimal'e çevirin

11011000.00011011.00111101.01110110'i hexadecimal'de ifade etmek için aşağıdaki tabloyu tamamlayın.

| Binary | 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 | Hexadecimal |
|---|---|---|---|---|---|---|---|---|---|
| 11011000 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | D8 |
| 00011011 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1B |
| 00111101 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 3D |
| 01110110 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 76 |

11001010.11110101.10000011.11101011'i hexadecimal'de ifade etmek için aşağıdaki tabloyu tamamlayın.

| Binary | 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 | Hexadecimal |
|---|---|---|---|---|---|---|---|---|---|
| 10000100 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 84 |
| 11010010 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | D2 |
| 01000011 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 43 |
| 10110011 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | B3 |

10000100.11010010.01000011.10110011'i hexadecimal'de ifade etmek için aşağıdaki tabloyu tamamlayın.

| Binary | 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 | Hexadecimal |
|---|---|---|---|---|---|---|---|---|---|
| 10000100 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 84 |
| 11010010 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | D2 |
| 01000011 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 43 |
| 10110011 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | B3 |

# 2 TCP/IP'ye Giriş

# 2 TCP/IP'ye Giriş

- TCP/IP ve DoD Modeli
- IP Adresleme
- Broadcast Adresleri
- Özet
- Sınav Temelleri
- Yazılı Lab 2
- Gözden Geçirme Soruları
- Gözden Geçirme Sorularının Cevapları
- Yazılı Lab 2 Cevapları

# TCP/IP'ye Giriş

Transmission Control Protocol/Internet Protocol (TCP/IP) ailesi, hem veri bütünlüğünden emin olmak ve korumak hem de bir savaş durumunda iletişimi devam ettirmek için Department of Defense (DoD) tarafından geliştirilmiştir. Bu nedenle, doğru şekilde tasarlanıp gerçekleştirildiyse, bir TCP/IP ağı gerçekten güvenilir ve esnek olabilir. Bu bölümde TCP/IP protokollerini işleyeceğim ve bu kitap boyunca, Cisco router'lar kullanarak, mükemmel bir TCP/IP ağının nasıl oluşturulacağını öğreteceğim.

DOD'un TCP/IP versiyonuna bir göz atarak başlayacağız ve sonra onun bu versiyonu ile "Ağlar Arası İletişim" başlıklı bölüm 1'de bahsedilen OSI referans modeli protokollerini karşılaştıracağız.

DoD modelinin farklı seviyelerinde kullanılan protokolleri kavradığınızda, IP adreslemesi ve bugünün ağlarında kullanılan farklı adres sınıflarını işleyeceğim.

*Subnetting "IP Subnetting, Variable Length Subnet Mask'lar (VLSM'ler) ve TCP/IP Sorun Gidermek" başlıklı bölüm 3'de işlenecektir.*

NOT

Broadcast adresleri, hem IP adreslemesini hem de subnetting'i ve VLSM'i anlamak için çok önemli olduğundan, broadcast adreslerini anlamak önemlidir. Bilmeniz gereken farklı tip broadcast adresleri ile bu bölümü tamamlayacağım.

*Bu bölüm güncellemeleri için www.lammle.com ve/veya www.sybex.com adreslerine bakınız.*

NOT

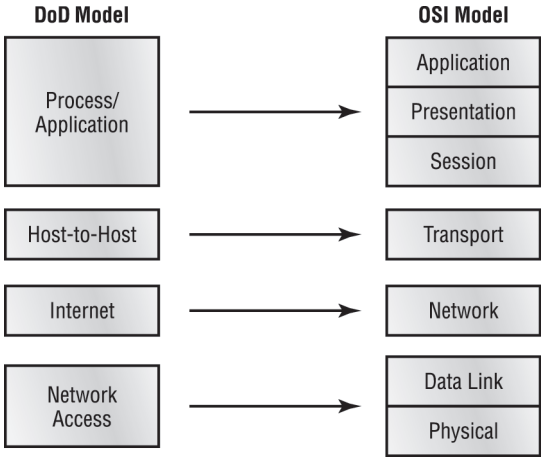
Internet Protocol version 6 (IPv6) bu bölümde işlenmeyecek; sadece IPv4'e odaklanacağız. IPv6, "IP Versiyon 6 (IPv6)" başlıklı bölüm 13'de işlenecektir. Ayrıca, Internet Protocol version 4 işlendiğinde, genel olarak IPv4 olarak değil de, sadece IP olarak yazıldığını göreceksiniz.

## TCP/IP ve DoD Modeli

DoD modeli, temel olarak, OSI modelinin yoğunlaştırılmış bir versiyonudur. Yedi katman yerine dört katmandan oluşur:

- Process/Application katmanı
- Host-to-Host katmanı
- Internet katmanı
- Network Access katmanı

Şekil 2.1, DoD modeli ile OSI referans modelinin bir kıyaslamasını göstermektedir. Görebileceğiniz gibi ikisi, kavramsal olarak benzerdir, fakat her ikisi de farklı isimlerle farklı katman sayılarına sahiptir.




***Şekil 2.1:*** *DoD ve OSI modelleri.*

*IP yığınındaki farklı protokollerden bahsedildiğinde, OSI ve DoD modellerinin katmanları, birbiriyle yer değiştirilebilir. Diğer bir deyişle, Host-to-host katmanının Transport katmanı ile aynı tanımlamalara sahip olması gibi, Internet katmanı da, Network katmanıyla aynı tanımlamaları açıklamaktadır.*

NOT

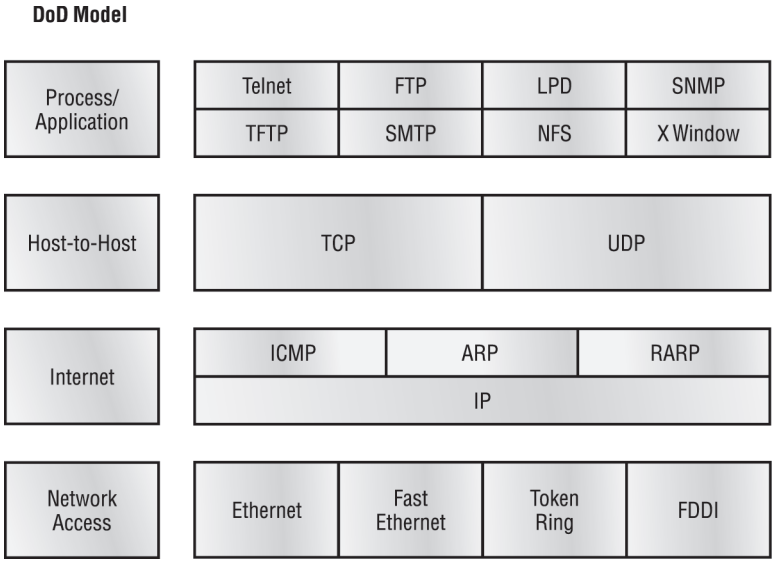
Birçok protokol serisi, OSI'nin üstteki üç katmanının (Application, Presentation ve Session) açıkladığı çeşitli aktivite ve görevleri tamamlamak için DoD modelinin Process/Application katmanında bir araya gelir. Bu bölümün sonlarına doğru bu protokollere detaylı bakacağız. Process/Application katmanı, node-to-node uygulama iletişimini tanımlar ve ayrıca kullanıcı-arayüzü düzenlemelerini kontrol eder.

Host-to-host katmanı, OSI'nin Transport katmanının fonksiyonlarıyla paraleldir, uygulamalar için iletim servis düzeyleri kurmak için protokoller tanımlar. Güvenli uçtan-uca iletişim oluşturmak ve verinin hatasız iletilmesi gibi konularla uğraşır. Paket sıralama kullanır ve veri bütünlüğünü devam ettirir.

Internet katmanı, OSI'nin Network katmanına denk gelir. Tüm ağ boyunca paketlerin mantıksal iletimiyle ilgili protokolleri tanımlar. Bir IP (Internet Protocol) adresi vererek kullanıcı makinelerinin adreslemesiyle ilgilenir ve çoklu ağlarda paketlerin route edilmesini ele alır.

DoD modelinin en altında, Network Access katmanı, kullanıcı makinesi ile ağ arasında dolaşan bilgiyi izler. OSI modelindeki Data Link katmanı ve Physical katmanın eşdeğeridir. Network Access katmanı, donanımsal adreslemeyi denetler ve verinin fiziksel iletimi için protokoller tanımlar.

DoD ve OSI modellerinin, tasarım ve kavramları aynıdır ve benzer katmanlarda, benzer fonksiyonlara sahiptir. Şekil 2.2, TCP/IP protokol ailesini ve protokollerinin, DoD model katmanlarıyla nasıl ilişkilendirildiğini gösterir.




***Şekil 2.2:*** *TCP/IP protokol ailesi.*

Sonraki bölümlerde, Process/Application katman protokollerinden başlayarak, değişik protokollere detaylı bir şekilde bakacağız.

## Process/Application Katman Protokolleri

Bu bölümde, tipik olarak IP ağlarında kullanılan farklı uygulama ve servisleri açıklayacağım. Aşağıdaki protokol ve uygulamalar bu bölümde işlenmektedir:

- Telnet
- FTP
- TFTP
- NFS
- SMTP
- LPD
- X Window
- SNMP
- DNS
- DHCP/BootP

### Telnet

Uzmanlık alanı, terminal emülasyonu olan Telnet, protokollerin bukalemunudur. Telnet client denilen uzak istemci makinesindeki bir kullanıcıya, Telnet sunucusu olan diğer bir makinenin kaynaklarına erişim sağlar. Telnet, sunucuyu kandırarak ve kullanıcı makinesine, lokal ağa doğrudan bağlı bir terminal gibi görünerek bunu başarır. Bu projeksiyon aslında, seçilen uzak makine ile birbirini etkileyebilen, sanal bir terminal olan yazılımsal bir imajdır.

Bu terminaller, text-mod tiptedir ve kullanıcılara, sunucudaki uygulamalara erişim ve alternatifleri seçme olanağı sağlayan menüleri göstermek gibi hafifletilmiş prosedürleri çalıştırabilir. Kullanıcılar, Telnet client yazılımı kullanarak bir Telnet oturumu başlatır ve sonra Telnet sunucusuna bağlanırlar.

### File Transfer Protocol (FTP)

File Transfer Protocol (FTP), gerçekten dosyaları transfer etmemizi sağlar ve bunu, protokolü kullanan iki makine arasında başarır. Fakat sadece bir protokol değildir; ayrıca bir programdır. Bir protokol olarak çalıştığında, FTP uygulamaları tarafından kullanılır. Bir program olduğunda, kullanıcılar tarafından manuel olarak dosya hizmetlerini çalıştırmak için kullanılır. FTP, ayrıca her iki yönde erişime izin verir ve farklı dizine yerleştirmek gibi, dizin operasyonlarını başarabilir. FTP, sizi FTP sunucusuna bağlamak ve sonra dosyaların transferini sağlamak için Telnet'le birlikte çalışır.

FTP yardımıyla bir kullanıcıya erişim sadece ilk adımdır. Kullanıcılar, erişimi sınırlandırmak için sistem yöneticileri tarafından uygulanan, kullanıcı adı ve şifre ile güvenli bir kimlik doğrulamasından geçirilmelidir. Erişiminiz sınırlı olsa da, kullanıcı adını anonim (anonymous) kabul ederek bundan biraz kurtulabilirsiniz.

Bir program olarak, kullanıcılar tarafından manuel çalıştırıldığı zaman dahi, FTP'nin fonksiyonları, dizinleri listelemek ve manipüle etmek, dosya bileşenlerini yazmak ve kullanıcılar arasında dosyaları kopyalamakla sınırlıdır. Uzak dosyaları, bir program gibi çalıştıramaz.

### Trivial File Transfer Protocol (TFTP)

Trivial File Transfer Protocol (TFTP), FTP'nin atası, basitleştirilmiş bir versiyonudur. Fakat neyi istediğinizi ve onu nerede bulacağınızı tam olarak biliyorsanız seçeceğiniz protokol budur. Artı, kullanımı kolay ve çok hızlıdır. Yine de, FTP'nin sağladığı fonksiyon zenginliğini size vermez. TFTP, dizin-arama kabiliyetine sahip değildir; gönderme ve alma dışında hiçbir şey yapamaz. Bu küçük kapasiteli protokol ayrıca, FTP'den daha ufak veri blokları göndererek, veri kısmında tasarruf sağlar ve FTP'de olduğu gibi kimlik doğrulaması yoktur. Bu sebeple güvensizdir. Doğal güvenlik riskleri sebebiyle çok az site onu destekler.

## Gerçek Dünya Senaryosu

## FTP'yi Ne Zaman Kullanmalıyız?

*San Francisco ofisinizdeki çalışanların, e-mail ile göndermeniz gereken 50MB boyutunda bir dosyaya ihtiyaçları var. Ne yaparsınız? Birçok e-mail sunucusu, sınırlı boyut kısıtlamalarınızdan dolayı bu dosyayı göndermenize izin vermeyecektir. Sunucuda, boyut sınırı olmasa dahi, bu büyüklükte bir dosyayı San Fransisco'ya göndermek uzun bir zaman alacaktır. İşte FTP, burada hayat kurtarır!*

*Birine büyük bir dosya vermeniz gerekiyorsa veya birinden büyük bir dosya alacaksanız, FTP iyi bir seçimdir. Bir kablo modeme veya DSL bant genişliğine sahipseniz, küçük dosyalar (5MB'tan daha az), e-mail yoluyla gönderilebilir. Bununla beraber, birçok ISP, 5MB'tan büyük dosyaların e-mail ile gönderilmelerine izin vermez, Bu nedenle FTP, büyük dosyalar göndermek ve almak zorunda olduğunuzda, düşünmek zorunda olduğunuz bir seçimdir. (Bu günlerde bunu yapmaya kim mecbur değil ki?) FTP kullanmak için, İnternette, dosyaların paylaşılabileceği bir FTP sunucusu kurmanız gerekecektir.*

*Ayrıca, FTP, e-mail'den daha hızlıdır, buda büyük dosyaları alıp göndermede FTP kullanmak için başka bir sebeptir. İlave olarak, TCP kullanması ve connection-oriented olması nedeniyle, oturum kapansa dahi, FTP, tekrar başladığında kaldığı yerden devam edebilir. Bunu birde e-mail ile deneyin!*

### Network File System (NFS)

Network File System (NFS), dosya paylaşımında uzman bir protokoldür. Birlikte çalışmak için iki farklı dosya sistemine izin verir. Şu şekilde çalışır: NFS sunucu yazılımının, bir NT sunucuda ve NFS kullanıcı yazılımının, bir Unix kullanıcı makinesinde çalıştığını farz edelim. NFS, sırası gelince Unix kullanıcıları tarafından kullanabilen Unix dosyalarını, açık olarak saklamak için NT sunucuda RAM'in bir bölümüne izin verir. NT ve Unix dosya sistemleri birbirinden farklı olsalar da (farklı büyük/küçük harf duyarlılığı, dosya adı uzunluğu, güvenlik v.s.sahiptirler), hem Unix hem de NT kullanıcıları, aynı dosyaya kendi normal dosya sistemleriyle erişebilirler.

### Simple Mail Transfer Protocol (SMTP)

Simple Mail Transfer Protocol (SMTP), e-mail göndermek için hazır isteklerimize cevap vererek, bekletme veya kuyruklama kullanan bir mail teslim yöntemidir. Mesaj, bir hedefe gönderilince, mesaj bir cihazda (genelde bir diskte) bekleme listesine yerleştirilir. Hedefteki sunucu yazılımı, mesajlar için kuyruğu düzenli olarak kontrol ederek bir uyarı postalar. Onları fark ettiğinde, hedeflerine teslim etmeye başlar. SMTP mail göndermek, POP3 mail almak için kullanılır.

### Line Printer Daemon (LPD)

Line Printer Daemon (LPD) protokolü, yazıcı paylaşımı için tasarlanmıştır. LPD, Line Printer (LPR) programı ile birlikte, yazıcı görevlerinin bekletilmesini sağlar ve TCP/IP kullanarak ağ yazıcılarına gönderir.

### X Window

İstemci/sunucu operasyonları için tasarlanan X Window, GUI (graphical user interface) tabanlı istemci/sunucu uygulamaları yazmak için bir protokol tanımlar. Amaç, bir bilgisayarda istemci olarak tanımlanan bir programın çalışmasını sağlamak ve başka bilgisayarda, bir window sunucu üzerinde olanları görüntülemektir.

### Simple Network Management Protocol (SNMP)

Simple Network Management Protocol (SNMP), önemli ağ bilgilerini toplar ve işler. Belirli veya rastgele zaman aralıklarında, bir yönetim istasyonundan ağdaki cihazları sorgulayarak veri toplar. Cihazlardan, kesin bilgilerini açıklamalarını ister. Her şey normal olduğunda SNMP, bazen baseline denilen, sağlıklı bir ağın işlevsel özelliklerini belirten bir rapor alır. Bu protokol, ağda bir bekçi köpeği gibi bekleyebilir ve olayların ani değişimini yöneticilere haber verir. Bu ağ bekçi köpekleri, acenteler olarak tanımlanır ve anormallik olduğunda acenteler, yönetim istasyonuna trap olarak bilinen bir alarm gönderir.

### Domain Name Service (DNS)

Domain Name Service (DNS), bilgisayar isimlerini, özellikle de `www.routersim.com` gibi internet isimlerini çözümler. DNS kullanmak zorunda değilsiniz, sadece bağlanmak istediğiniz bir cihazın IP adresini yazabilirsiniz. Bir IP adresi, hem ağdaki hem de internetteki kullanıcı makinelerini tespit eder. Bununla birlikte DNS, yaşamımızı kolaylaştırmak için tasarlanmıştır. Şunu düşünün: Web sayfanızı, başka bir servis sağlayıcıya taşımak isterseniz ne olurdu? IP adresiniz değişecekti ve hiç kimse yeni IP'nizi bilmeyecekti. DNS, size, IP adresi belirtmek için bir domain adı kullanmanızı sağlar. IP adresinizi istediğiniz sıklıkta değiştirebilirsiniz böylece hiç kimse değişikliği bilmeyecektir.

**NOT** *DNS ile ilgili hatırlamanız gereken önemli bir nokta, bir cihazı IP adresi ile ping'leyebiliyor fakat onun FQDN'ini kullanamıyorsanız, bazı DNS konfigürasyon hatalarınız olabileceğidir.*

DNS; örneğin `www.lammle.com` veya `todd.lammle.com` gibi bir FQDN'i (fully qu-

alified domain name) çözümlemek için kullanılır. FQDN, domain identifier tabanlı bir sistemi, mantıksal olarak yerleştirebilen bir hiyerarşidir.

### Dynamic Host Configuration Protocol (DHCP) / Bootstrap Protocol (BootP)

Dynamic Host Configuration Protocol (DHCP), kullanıcı makinelerine IP adresi tahsis eder. Daha kolay yönetim sağlar ve küçükten, çok geniş ağ ortamlarına kadar iyi çalışır. Cisco router dahil donanımların tümü, bir DHCP sunucu olarak kullanılabilir.

DHCP, BootP'den farklıdır. Şöyle ki BootP, bir kullanıcı makinesine IP adresi tanımlar fakat makinenin donanım adresi, BootP tablosuna manuel olarak girilmelidir. DHCP'yi dinamik bir BootP olarak düşünebilirsiniz. Fakat BootP'nin ayrıca, bir host makinesinin boot edebileceği bir işletim sistemi göndermek için kullanıldığını da hatırlayın. DHCP, bunu yapamaz.

Kullanıcı makinesi, DHCP sunucusundan bir IP adresi istediğinde, DHCP sunucusunun, istemci makineye sağlayabileceği birçok bilgi vardır. Aşağıda, DHCP sunucusunun sağladığı bilgilerin listesini bulabilirsiniz:

- IP adresi
- Subnet mask
- Domain ismi
- Default gateway (router'lar)
- DNS
- INS bilgisi

Bir DHCP sunucusu, bunlardan fazla bilgi de verebilir, fakat listedekiler en yaygın olanlarıdır.

IP adresi almak için bir DHCP Discover mesajı gönderen bir istemci, hem katman2 hem de katman3 broadcast'i yayımlar. Katman2 broadcast, FF:FF:FF:FF:FF:FF olarak görünen, hexdecimal olarak tamamı F'lerden oluşur. Katman3 broadcast'i, tüm ağlar ve kullanıcılar anlamına gelen, 255.255.255.255'dir. DHCP, connectionless'dir. Bu (sonra bahsedeceğimiz) Host-to-host katmanı olarak da bilinen Transport katmanında UDP (User Datagram Protocol) kullanıyor anlamına gelir.

Burada, güvenilir Ethereal analizöründen bir çıktı örneği bulabilirsiniz:

```
 Ethernet II, Src: 192.168.0.3 (00:0b:db:99:d3:5e), Dst:
Broadcast  (ff:ff:ff:ff:ff:ff)
 Internet Protocol, Src: 0.0.0.0     (0.0.0.0),   Dst:
255.255.255.255  (255.255.255.255)
```

Data Link katmanı ve Network katmanı, herkese "Yardım edin-IP adresimi bilmiyorum" şeklinde broadcast'ler yayımlar.

*Broadcast adresleri, detaylı şekilde bu bölümün sonunda işlenecektir.*

NOT

## Host-to-host Protokolleri

Host-to-host katmanının ana amacı, üst-katman uygulamalarını, ağın karmaşıklığından korumaktır. Bu katman, üst katmanlara "Veri akışlarını, açıklamalarıyla bana verin, ben bilgilerinizi göndermeye hazır hale getirmek için işleme başlayacağım." der.

Aşağıdaki bölümler, bu katmandaki iki protokolü açıklamaktadır.

- Transmission Control Protocol (TCP)
- User Datagram Protocol (UDP)

*Bunun hala katman4'te dikkate alındığını ve Cisco'nun, katman4'ün kullanabildiği acknowledgment, sequencing ve akış kontrolü yöntemlerinden hoşlandığını hatırlayın.*

NOT

İlave olarak, hem bazı anahtar host-to-host protokol kavramlarına hem de port numaralarına bakacağız.

## Transmission Control Protocol (TCP)

Transmission Control Protocol (TCP), bir uygulamadan, geniş bilgi parçalarını alır ve onları segment'lere ayırır. Her segment'i numaralar ve sıralar, böylece hedefin TCP yığını, segment'leri tasarlanan uygulama için tekrar sıraya koyar. Bu segment'lerin gönderilmesinden sonra, TCP (verici makinesinde), alıcı ucun TCP sanal devre oturumu için bir acknowledgment bekler, onaylanmayanları tekrar gönderir.

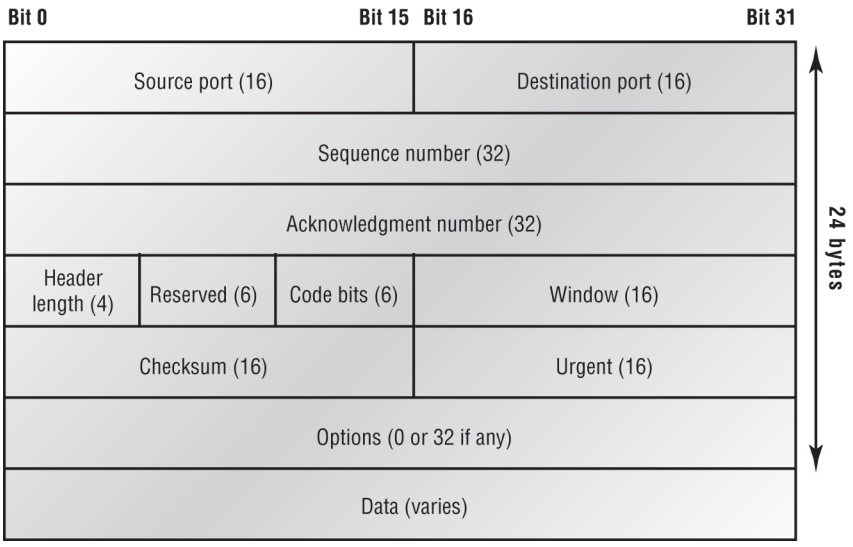
Verici makinelerin segment'leri modele göndermeye başlamasından önce, göndericinin TCP yığını, bir bağlantı oluşturmak için hedefin TCP yığını ile iletişime geçer. Oluşturulan, sanal bir devre olarak bilinir. Bu iletişim tipi, connection-oriented'dir. Başlangıç anlaşması esnasında, iki TCP katmanı, alıcının TCP'sinin, tekrar bir acknowledgment göndermesinden önce gönderilecek bilgi miktarı konusunda anlaşır. Önceden her konuda anlaşarak, güvenli iletişim olması için bir yol oluşturulur.

TCP, full-duplex, connection-oriented, güvenli ve kusursuz bir protokoldür, fakat tüm bu şart ve koşulları sağlamak, ilave olarak hata kontrolü yapmak, az iş değildir. TCP, çok karmaşıktır ve ağın genel giderleri açısından oldukça pahalıdır. Ayrıca günümüz ağları, geçmiştekilerden çok daha güvenlidir, bu ilave güvenlik, çoğu kez gereksizdir.

## TCP Segment Formatı

Üst katmanlar, Transport katmanındaki protokollere sadece bir veri akımı gönderdiğinden, TCP'nin bir veri akımını nasıl segment'lediğini ve onları Internet katmanı için nasıl hazırladığını göstereceğim. Internet katmanı, veri akımını aldığında segment'leri bir ağ topluluğuna paket olarak route eder. Segment'ler, alıcı makinenin Host-to-host katman protokolüne gönderilir. Host-to-host katman protokolü, üst-katman uygulama ve protokollerine göndermek için veri akımlarını tekrar oluşturur.

Şekil 2.3, TCP segment formatını gösterir. Şekil, TCP başlığındaki farklı alanları göstermektedir.




***Şekil 2.3:*** *TCP segment formatı.*

TCP başlığı, 20 byte uzunluğundadır (isteğe göre 24 byte'a kadar çıkabilir). TCP segment'indeki her alanı anlamanız gerekir:

**Source port:** Veriyi gönderen kullanıcı makinesindeki uygulamanın port numarasıdır.

**Destination port:** Hedef makinede istenen uygulamanın port numarasıdır.

**Sequence number:** Veriyi doğru sıraya geri koyan ya da kayıp veya bozuk veriyi tekrar ileten (sequencing olarak bilinen bir işlem), TCP tarafından kullanılan bir numaradır.

**Acknowledgment number:** Bir sonraki olması beklenen TCP okteti.

**Header length:** TCP başlığındaki 32-bit numara. Bu, verinin nerede başladığını belirtir. TCP başlığı (seçenekler içeren bir dahi), 32 bit uzunluğunda integral bir numaradır.

**Reserved:** Daima sıfıra ayarlanır.

**Code bits:** Bir oturum kurmak ve sonlandırmak için kullanılan kontrol fonksiyonlarıdır.

**Window:** Göndericinin kabul edeceği, oktetlerdeki window boyutu.

**Checksum:** TCP, alt katmanlara güvenmediğinden, CRC (cyclic redundancy check), her şeyi denetler. CRC, başlık ve veri alanlarını kontrol eder.

**Urgent:** Sadece, kod bit'lerindeki Urgent pointer ayarlandığında geçerli bir alandır. Şayet ayarlandıysa, bu değer, zorunlu olmayan verinin ilk segment'inin başladığı oktetlerdeki geçerli sıra numarasından sapmayı gösterir.

**Options:** Sıfır (0) veya 32 bit'in katlarıdır. Bunun anlamı, hiçbir seçenek olmamasıdır (seçenek boyutu 0). Bununla birlikte, seçenek alanında, toplamda 32 bit'in katlarına ulaşmayan herhangi bir seçenek kullanıldığında, verinin 32-bit limiti ile başladığından emin olmak için 0'lardan oluşan bir takviye (dolgu) kullanılır.

**Data:** Transport katmanındaki TCP protokolüne gönderilir, üst katman başlıklarını içerir.

Bir ağ analizöründen kopyalanan TCP segment'ine bir bakalım:

```
TCP - Transport Control Protocol
 Source Port:      5973
 Destination Port: 23
 Sequence Number:  1456389907
 Ack Number:       1242056456
 Offset:           5
 Reserved:         %000000
 Code:             %011000
        Ack is valid
        Push Request
 Window:           61320
 Checksum:         0x61a6
 Urgent Pointer:   0
 No TCP Options
 TCP Data Area:
 vL.5.+.5.+.5.+.5  76 4c 19 35 11 2b 19 35 11 2b 19 35 11
  2b 19 35 +. 11 2b 19
 Frame Check Sequence: 0x0d00000f
```

Segment'te, daha önce bahsettiğim her şeye dikkat ettiniz mi? Başlıktaki alanların sayısından görebileceğiniz gibi, TCP birçok ek yük oluşturur. Uygulama geliştiriciler, ek yükü azaltmak için güvenirlik yerine verimliliği seçebilir. Bu nedenle, Transport katmanında, bir alternatif olarak User Datagram Protocol (UDP) tanımlanmıştır.

## User Datagram Protocol (UDP)

Şayet, User Datagram Protocol'ü (UDP), TCP ile mukayese edersiniz eski olan, bazen zayıf bir protokol olarak belirtilen, daha ekonomik bir modeldir. Park bankında oturan zayıf bir kişi gibi, zayıf protokol, fazla yer (bu durumda, bir ağda fazla bant genişliği) kaplamaz.

UDP, ayrıca TCP'nin birçok ekstra özelliğini sağlamaz, fakat güvenli teslim gerektirmeyen bilgi aktarımını çok iyi yapar ve böyle yaparak çok daha az ağ kaynağı kullanır. (UDP, RFC 768'de tanımlanmıştır.)

Geliştiricilerin TCP yerine UDP'yi seçmelerinin kesinlikle daha akıllıca olduğu bazı durumlar vardır. SNMP bekçi köpeğinin Process/Application katmanında olduğunu hatırladınız mı? SNMP, özellikle geniş bir ağda çalıştığında, aralıklı mesajlar ile yeterince düzenli güncelleme ve alarm akımları göndererek ağı izler. Tüm bu küçük mesajları kurmak, devam ettirmek ve kapatmak için ek yük maliyeti, sağlıklı ve hızlı çalışan bir ağ ile düşebilir.

**NOT** *Requests for Comment'ler (RFC), Internet (orijinali ARPAnet'tir) hakkında, 1969'da alınmaya başlanan notların bütününü düzenler. Notlar, bilgisayar iletişiminin farklı bakış açılarını ele alır: Ağ kurulum protokolleri, prosedürler, programlar ve kavramlara odaklanırlar, ayrıca, toplantı notları, görüşler ve bazen mizahı da içerir.*

TCP'ye karşı UDP'nin tercih edildiği diğer bir durum güvenirliliğin, zaten Process/Application katmanında ele alındığı zamandır. Network File System (NFS), kendi güvenilirlik sorunlarını ele alarak, TCP'nin kullanımını mantıksız ve gereksiz kılar. Fakat sonunda, UDP'nin mi TCP'nin mi kullanılacağına, veriyi daha hızlı transfer etmek isteyen kullanıcı değil, uygulama geliştirici karar verecektir.

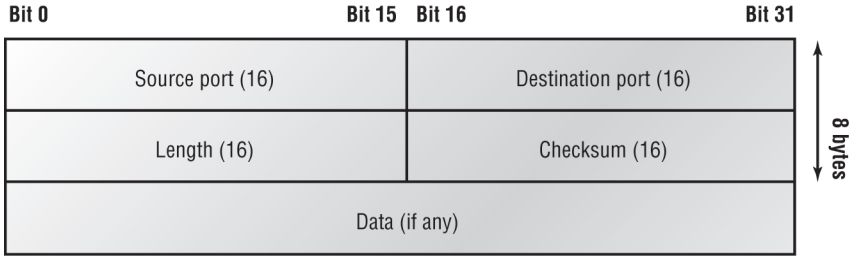
UDP, segment'leri sıraya almaz ve segment'lerin, hedefe hangi sırayla ulaşacağıyla ilgilenmez. UDP, segment'leri gönderir ve onları unutur. Takip etmez, onları kontrol etmez veya güvenli erişim onayına bile izin vermez. Bundan dolayı, güvenilmez bir protokol olarak belirtilir. Bu UDP'nin faydasız bir protokol olduğu anlamına gelmez, sadece güvenlik sorunlarını ele almaz.

Dahası, UDP ne sanal devre oluşturur ne de bilgileri ulaştırmadan önce hedefle bağlantı kurar. Bunun için, ayrıca connectionless bir protokol olarak kabul edilir. UDP, uygulamanın kendi güvenlik yöntemini kullanacağını farz ettiğinden, kendisi hiçbir şey kullanmaz. Bu, Internet Protokol yığını çalıştığında, uygulama geliştiricilere bir seçme şansı verir: TCP güvenilirlik için UDP daha hızlı transferler içindir.

Örneğin, VoIP (Voice over IP) kullanıyorsanız, gerçekten UDP kullanmayı istemezsiniz. Şayet segment'ler sıra ile ulaşmazlarsa (IP ağlarında çok yaygındır), hangi sırayla alınırsa alınsın, sonraki OSI (DoD) katmanına geçirilecektir, bu da oldukça bozuk bazı verilere neden olacaktır. Diğer yandan, TCP, segment'leri sıraya koyar ki, sonradan onlar tamamen doğru sıraya konarak bir araya getirilsin. (UDP bunu yapamaz.)

### UDP Segment Formatı

Şekil 2.4, TCP'nin aşırı kullanımı ile karşılaştırıldığında, UDP'nin önemli derecede düşük ek yükü açıkça görülür. UDP'nin windowing'i kullanmadığını ve UDP başlığında acknowledment'ları desteklemediğini görebiliyor musunuz?




***Şekil 2.4:*** *UDP segment'i.*

UDP segment'indeki her alanın ne olduğunu anlamanız önemlidir:

**Source port:** Veri gönderen kullanıcı makinesindeki uygulamanın port numarası.

**Destination port:** Hedef makinede istenen uygulamanın port numarası.

**Lenght:** UDP başlığı ve UDP verisinin uzunluğu.

**Checksum:** Hem UDP başlığı hem de UDP veri alanının sağlaması.

**Data:** Üst-katman verisi.

UDP, TCP gibi, alt-katmanlara güvenmez ve kendi CRC'sini çalıştırır. FCS'in (Frame Check Sequence ), FCS bilgisini görebilmeniz için CRC'yi barındıran alan olduğunu hatırlayın.

Aşağıda, bir ağ analizörü tarafından yakalanan UDP segment'i görülmektedir:

```
UDP - User Datagram Protocol
Source Port:      1085
Destination Port: 5136
Length:           41
Checksum:         0x7a3c
UDP Data Area:
..Z......00 01 5a 96 00 01 00 00 00 00 00 11 0000 00
...C..2._C._C  2e 03 00 43 02 1e 32 0a 00 0a 00 80 43 00 80
Frame Check Sequence: 0x00000000
```

Düşük ek yüke dikkat! UDP segment'inde, sequence number, ack number ve window size'ı bulmaya çalışın. Bulamazsınız, çünkü yoklar!

## Host-to-host Protokollerinin Anahtar Kavramları

Hem connection-oriented (TCP) hem de connectionless (UDP) protokollerinin etkilerini gördüğünüzden, ikisini burada özetlemek iyi olacaktır. Şekil 2.1, bu iki protokol hakkında aklınızda tutmanız gereken anahtar kavramları vurgulamaktadır. Bu tabloyu ezberlemeniz gerekmemektedir.

**Tablo 2.1:** TCP ve UDP'nin Anahtar Özellikleri

| TCP | UDP |
|---|---|
| Sequenced | Unsequenced |
| Reliable | Unreliable |
| Connection-oriented | Connectionless |
| Virtual circuit | Low overhead |
| Acknowledgment | Acknowledgment yok |
| Windowing flow control | Windowing ya da flow contol yok |

Bir telefon örneklemesi, TCP'nin nasıl çalıştığını anlamanıza gerçekten yardımcı olabilir. Çoğunuz, telefonda biriyle konuşmadan önce ilk olarak, nerede olurlarsa olsunlar diğer insanlarla bir bağlantı kurulması gerektiğini bilirsiniz. Bu, TCP protokol ile kurulan bir sanal devreye benzer. Şayet görüşmeniz sırasında birine önemli bilgi veriyorsanız, "Biliyor musun?" diyebilir veya "Onu anladın değil mi?" diye sorabiliriz. Böyle şeyler söylemek, sizi doğrulamak için tasarlanan birçok TCP acknowledgment'larına benzer. Bazen (özellikle telefon görüşmesinde) insanlar ayrıca "Hala orada mısın?" diye sorarlar. Görüşmelerinin sonunda "Hoşça kal" diyerek aramayı bitirirler. TCP ayrıca bu tür fonksiyonlar çalıştırır.

UDP kullanmak bir posta kartı göndermeye benzer. Bunu yapmak için ilk olarak diğer taraf ile bağlantı kurmaya ihtiyacınız yoktur. Basit olarak mesajınızı yazar, posta kartının gideceği adresi belirtir ve onu postalarsınız. Bu, UDP'nin connectionless oryantasyonuna benzer. Posta kartındaki mesaj, ölüm kalım meselesi olmadığından, onu gönderenden bir onaya ihtiyacınız yoktur. Benzer şekilde UDP, acknowledgment gerektirmez.

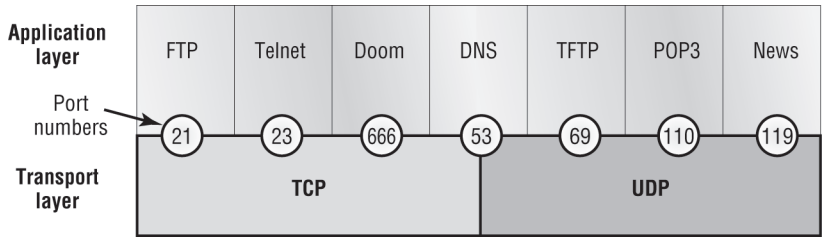
Şimdi, TCP, UDP ve her protokol'le ilgili uygulamaları içeren diğer şekle, Şekil 2.5'e bir bakalım.

## Port Numaraları

TCP ve UDP, ağ üzerinde eşzamanlı olarak, farklı görüşmeleri takip edebildikleri için, üst katmanlarla bağlantı kurmak için port numaralarını kullanmak zorundadırlar. Oluşturulan kaynak port numaraları, dinamik olarak kaynak makine tarafından atanır ve 1024 ile başlayan bazı numaralara eşittir. RFC 3232 (veya sadece `www.iana.com`) ile tanımlı, 1023 ve aşağısındaki numaralar, iyi bilinen port numaraları olarak belirtilirler.

En bilinen port numaraları ile bir uygulama yapmayan sanal devrelere, belirli bir aralıktan rastgele port numarası atanmaktadır. Bu port numaraları, kaynak ve hedef uygulamalarını veya TCP segment'indeki prosesi tanımlar.

Şekil 2.5, hem TCP hem de UDP'nin port numaralarını nasıl kullandığını göstermektedir.




***Şekil 2.5:*** *TCP ve UDP için port numaraları.*

Kullanılabilen farklı port numaraları aşağıda açıklanmaktadır:

- 1024 altındaki numaralar, iyi-bilinen port numaraları olarak kabul edilirler ve RFC 3232 ile tanımlanmıştır.
- 1024 ve üstü numaralar, diğer makineler ile oturumlar kurmak için üst katmanlar tarafından, TCP segment'inde kaynak ve hedef adresleri olarak TCP tarafından kullanılmaktadır.

Sonraki bölümde, TCP oturumunu gösteren bir analizör çıktısına bakacağız.

### TCP Oturumu: Source Port

Aşağıdaki liste, OmniPeek analizör yazılımı ile yakalanan bir TCP oturumunu göstermektedir:

```
TCP - Transport Control Protocol
Source Port:      5973
Destination Port: 23
Sequence Number:  1456389907
Ack Number:       1242056456
Offset:           5
Reserved:         %000000
Code:             %011000
    Ack is valid
    Push Request
Window:           61320
Checksum:         0x61a6
Urgent Pointer:   0
No TCP Options
TCP Data Area:
vL.5.+.5.+.5.+.5  76 4c 19 35 11 2b 19 35 11 2b 19 35 11
 2b 19 35 +. 11 2b 19
Frame Check Sequence: 0x0d00000f
```

Kaynak makinenin, bu durumda 5973 olan source portu oluşturduğuna dikkat edin. Alıcı makineye, planlanan bağlantının (Telnet) amacını belirtmek için kullanılan destination portu, 23'tür.

Bu oturuma bakarak, kaynak makinenin, 1024-65535 arası numaraları kullanarak source portu oluşturduğunu görebilirsiniz. Fakat kaynak, neden bir port numarası oluşturur? Farklı makineler ile oturumları ayırt etmek için. Bir sunucu, gönderici makine için farklı bir numaraya sahip değilse, bilginin nereden geldiğini nereden bilebilir? TCP ve üst katmanlar, gönderen makinenin adresini anlamak için Data Link ve Network katmanlarının yaptığı gibi donanımsal ve mantıksal adresler kullanmaz. Onun yerine port numaraları kullanırlar. Ve kolaylıkla tahmin edebileceğiniz gibi, şayet tüm makineler FTP'ye erişmek için aynı port numarasını kullanırsa, alıcı makinenin kafası tamamıyla karışır.

### TCP Oturumu: Destination Portu

Bazen bir analizöre bakacak ve sadece source portunun 1024 üstünde olduğunu, destination portunun iyi-bilinen bir port olduğunu göreceksiniz. Bunu aşağıda görebilirsiniz:

```
TCP - Transport Control Protocol
 Source Port:      1144
 Destination Port: 80 World Wide Web HTTP
 Sequence Number:  9356570
 Ack Number:       0
 Offset:           7
 Reserved:         %000000
 Code:             %000010
        Synch Sequence
 Window:           8192
 Checksum:         0x57E7
 Urgent Pointer:   0
 TCP Options:
 Option Type: 2 Maximum Segment Size
    Length:    4
    MSS:       536
 Option Type: 1 No Operation
 Option Type: 1 No Operation
 Option Type: 4
   Length:    2
   Opt Value:
 No More HTTP Data
 Frame Check Sequence: 0x43697363
```

Ve gerçekten, source portu, 1024'ün üzerinde, fakat destination portu 80 veya HTTP servisidir. Sunucu veya alıcı makine, gerekirse destination portunu değiştirecektir.

Bundan önceki çıktıda, bir "syn" paketi, hedef makineye gönderilmektedir. Syn sırası, uzak hedef makineye bir oturum oluşturma istediğini belirtmektedir.

### TCP Oturumu: Syn Paket Acknowledgment

Şimdiki çıktı, syn paketi için bir acknowledgment'ı gösterir:

```
TCP - Transport Control Protocol
Source Port:      80 World Wide Web HTTP
```

```
Destination Port: 1144
Sequence Number:  2873580788
Ack Number:       9356571
Offset:           6
Reserved:         %000000
Code:             %010010
         Ack is valid
         Synch Sequence
 Window:           8576
 Checksum:         0x5F85
 Urgent Pointer:   0
 TCP Options:
 Option Type: 2 Maximum Segment Size
   Length:    4
   MSS:       1460
  No More HTTP Data
Frame Check Sequence: 0x6E203132
```

Ack'nin geçerli olduğuna dikkat edin. Yani source portu kabul edilmiş ve cihaz, isteyen makine ile sanal bir devre kurmak konusunda anlaşmıştır.

Yine burada, sunucudan cevabın, kaynağı 80 ve hedefi, başlatılan makineden gönderilen 1144 olarak gösterdiğini izleyebilirsiniz.

Tablo 2.2, TCP/IP ailesinde kullanılan tipik uygulamaların bir listesini verir. Bunlar, onların iyi-bilinen port numaraları ve her uygulama ve işlem tarafından kullanılan Transport katmanı protokolleridir. Bu tabloyu çalışmanız ve ezberlemeniz önemlidir.

**Tablo 2.2:** TCP ve UDP'nin Kullandığı Anahtar Protokoller

| TCP | UDP |
|---|---|
| Telnet 23 | SNMP 161 |
| SMTP 25 | TFTP 69 |
| HTTP 80 | DNS 53 |
| FTP 21 | |
| DNS 53 | |
| HTTPS 443 | |

NOT *TCP'yi güvenli yapan sequencing, acknowledgment ve akış kontrolüdür (windowing). UDP'nin güvenilirliği yoktur.*

DNS'in hem TCP hem de UDP kullandığına dikkat edin. Hangisini seçeceği, ne yapmaya çalıştığına bağlıdır. Her iki protokolü kullanan tek uygulama olmamasına rağmen, kesinlikle çalışmalarınızda hatırlamanız gereken bu olacaktır.

## Internet Katmanı Protokolleri

DoD modelinde, Internet katmanı olmasının iki ana nedeni vardır: Routing ve üst katmanlara tek bir ağ arayüzü sağlamak.

Diğer üst veya alt katman protokollerinin hiçbirisi, routing ile ilgili özelliklere (tamamıyla Internet katmanına ait olan karmaşık ve önemli görevlere) sahip değildir. Internet katmanının ikinci görevi, üst-katman protokollerine tek bir ağ arayüzü sağlamaktır. Bu katman olmaksızın, uygulama programcıları, her farklı Network Access protokolü için uygulamalarının her birine açıklama yazmak

zorunda kalacaklardı. Sadece bu sorun olmayacaktı, her uygulama için farklı versiyonlara ihtiyaç olacaktı (Ethernet için bir, Token Ring için başka bir v.s.). Bunu engellemek için, IP, üst-katman protokolleri için tek bir ağ arayüzü sağlamaktadır. Bu tamamlandıktan sonra, artık bir araya gelip beraber çalışmak, IP ve farklı Network Access protokollerinin görevidir.

Bütün ağ yolları Roma'ya çıkmaz, onlar IP'ye gider. Ve hem bu katmandaki diğer protokoller hem de üst katmandakiler onu kullanır. DoD modeli boyunca tüm yollar, IP'den geçer. Aşağıdaki bölüm, Internet katmanındaki protokolleri açıklamaktadır:

- Internet Protocol (IP)
- Internet Control Message Protocol (ICMP)
- Address Resolution Protocol (ARP)
- Reverse Address Resolution Protocol (RARP)
- Proxy ARP

## Internet Protocol (IP)

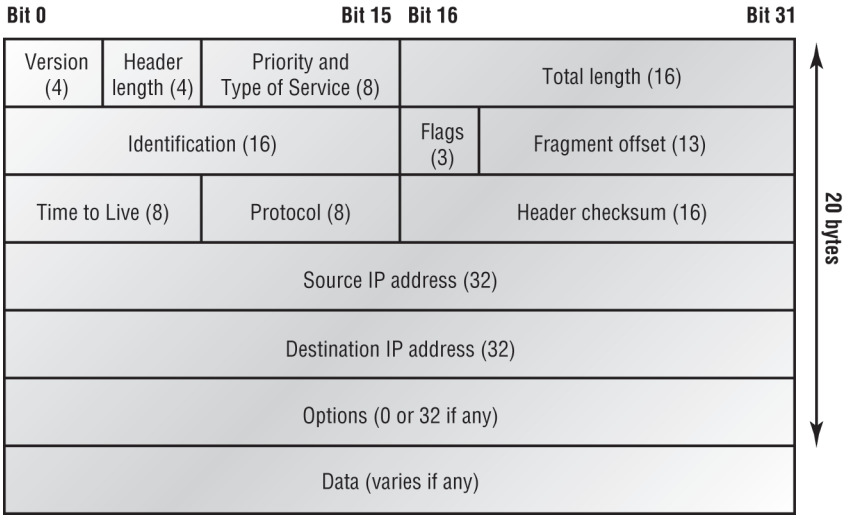
Internet Protocol (IP), aslında Internet katmanıdır. Burada bulunan diğer protokoller, sadece ona destek vermek içindir. IP, genel bakışa sahiptir, hepsini gördüğü söylenebilir ve birbirine bağlı ağların tamamından haberdardır. Bunu, ağdaki tüm makineler, IP adresi denilen bir yazılım veya mantıksal adrese sahip olduğundan yapabilir.(Bu bölümün sonlarına doğru bahsedeceğim.)

IP, her paketin adresine bakar. Sonra, bir routing tablosu kullanarak en iyi yolu seçip bir paketin nereye gönderileceğine karar verir. DoD modelinin en alttaki Network Access katmanı protokolleri, tüm ağda IP'nin aydınlatıcı kapsamına sahip değildir. Onlar sadece fiziksel linklerle (lokal ağlarla) ilgilenir.

Ağdaki cihazları tespit etmek, şu iki soruya cevap gerektirir: O hangi ağdadır? Ve onun bu ağdaki ID'si nedir? İlkinin cevabı, yazılım adresi veya mantıksal adres'tir (doğru cadde). İkincisinin cevabı, donanım adresidir (doğru posta kutusu). Bir ağdaki tüm kullanıcı makineleri, IP adresi olarak bilinen mantıksal bir ID'ye sahiptir. Bu, yazılım veya mantıksal adrestir ve değerli kodlanmış bilgiler içerir. Routing'in karmaşık görevini oldukça basitleştirir.(IP, RFC 791 ile düzenlenmiştir)

IP, Host-to-Host katmanından segment'leri alır ve gerekirse, onları datagram'lara (paketlere) böler. Daha sonra IP, alıcı tarafında, datagram'ları tekrar segment'lere dönüştürür. Her datagram'a, alıcının ve göndericinin IP adresi tanımlanır. Her router (katman3 cihazı), bir datagram alarak, paketin hedef IP adresine bağlı olarak routing kararları verir.

Şekil 2.6, bir IP başlığını gösterir. Bu şekil size, üst katmanlardan geldiğinde ve uzak bir ağa kullanıcı verisi gönderildiğinde, IP protokolünün nasıl çalıştığı hakkında bir fikir verecektir.




***Şekil 2.6:*** *IP başlığı.*

Aşağıdaki alanlar IP başlığını meydana getirir:

**Versiyon:** IP versiyon numarası.

**Header length:** 32-bit başlık uzunluğu.

**Priority and Type of Service:** Datagram'ın nasıl ele alınması gerektiğini söyleyen servis tipidir.

**Total length:** Başlık ve veri içeren paketin uzunluğudur.

**Identification:** Benzersiz IP-paket değeri.

**Flags:** Parçalanmanın olup olmadığını belirtir.

**Fragment offset:** Şayet, paket bir frame'e koymak için çok büyükse parçalama ve tekrar bir araya getirme sağlar. Aynı zamanda, internette farklı MTU'lara (maximum transmission unit) izin verir.

**Time to Live:** Time to Live, orijinal olarak üretildiğinde, bir pakete ayarlanır. TTL süresinin dolmasından önce, istediği yere gitmezse, paket iptal olur. Bu, IP paketlerinin ağda sürekli dolaşımını durdurur.

**Protocol:** Üst katman protokolün portudur (TCP, port 6'dır veya UDP, port 17'dir [hex]). Ayrıca ARP ve ICMP gibi, Network katmanı protokollerini destekler. Bazı analizörlerde, Type alanı olarak belirtilebilir. Bu alan hakkında detaylı şekilde bahsedilecektir.

**Header checksum:** Sadece başlıktaki CRC (cyclic redundancy check).

**Source IP address:** Gönderen istasyonun 32-bit IP adresi.

**Destination IP address:** Bu paketin hedeflendiği istasyonun 32-bit IP adresi.

**Options:** Ağı test etme, hata bulma, güvenlik v.s. için kullanılmaktadır.

**Data:** IP seçeneğinden sonraki alan, üst-katman verisi olacaktır.

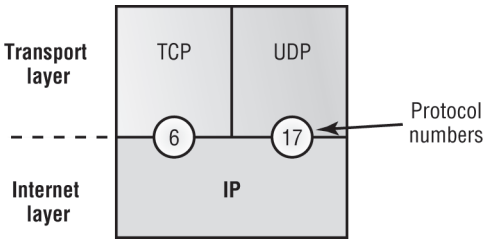
Aşağıda, bir ağ analizörünün yakaladığı IP paketinin anlık görüntüsünü bulabilirsiniz (daha önce anlatılan tüm başlık bilgilerinin burada göründüğüne dikkat edin).

```
IP Header - Internet Protocol Datagram
 Version:              4
 Header Length:        5
 Precedence:           0
 Type of Service:      %000
 Unused:               %00
 Total Length:         187
 Identifier:           22486
 Fragmentation Flags: %010 Do Not Fragment
 Fragment Offset:      0
 Time To Live:         60
 IP Type:              0x06 TCP
 Header Checksum:      0xd031
 Source IP Address:    10.7.1.30
 Dest. IP Address:     10.7.1.10
 No Internet Datagram Options
```

Genellikle bir Protocol alanı olan, fakat analizörün burada IP Type alanı olarak gösterdiği Type alanı önemlidir. Şayet başlık sonraki katmana protokol bilgisini taşımazsa IP, paketle taşınan

veriyi ne yapacağını bilmez. Yukarıdaki örnek, IP'nin segment'i TCP'ye geçirdiğini söylemektedir.

Şekil 2.7, Bir paketi üst-katmana aktarmaya ihtiyacı olduğunda, Network katmanının, Transport katmanındaki protokollerin nasıl farkına vardığını göstermektedir.




***Şekil 2.7:*** *IP başlığındaki protokol alanı.*

Bu örnekte, Protocol alanı IP'ye, veriyi hem TCP port 6 hem de UDP port 17'den (her ikisi de hex adresidir) göndermesini söyler. Şayet veri, bir üst katman servis veya uygulaması başlığına sahip bir veri akışının parçasıysa, ya UDP ya da TCP olacaktır. Veri, Internet Control Message Protocol (ICMP), Address Resolution Protocol (ARP) veya diğer Network ağ protokolü türleri için kolayca hedeflenebilir.

Tablo 2.3, Protocol alanında tanımlanabilen diğer bazı popüler protokollerin bir listesidir.

*Protokol alan numaralarının tam listesini www.iana.org/assignments/protocol-numbers adresinden bulabilirsiniz.*

NOT

**Tablo 2.3:** Bir IP Başlığının Protokol Alanında Bulunması Muhtemel Protokoller

| Protokol | Protokol Numarası |
|---|---|
| ICMP | 1 |
| IP in IP (tunelling) | 4 |
| IGRP | 9 |
| EIGRP | 88 |
| OSPF | 89 |
| IPv6 | 41 |
| GRE | 47 |
| Katman 2 tunnel (L2TP) | 115 |

## Internet Control Message Protocol (ICMP)

Internet Control Message Protocol (ICMP), Network katmanında çalışır ve birçok farklı servis için IP tarafından kullanılır. ICMP, bir yönetim protokolüdür ve IP için mesajlaşma servis sağlayıcısıdır. Mesajlar, IP datagram'ları gibi taşınmaktadır. RFC 1256, ağ geçitlerine route bulmak için genişletilmiş host kapasitesi sağlayan ICMP'ye bir ilavedir.

ICMP paketleri aşağıdaki özelliklere sahiptir:

- Kullanıcı makinelerine, ağ problemleri hakkında bilgi sağlar.
- IP datagram'larında enkapsüle edilirler.

Aşağıdakiler, ICMP ile ilgili bazı yaygın olay ve mesajlardır:

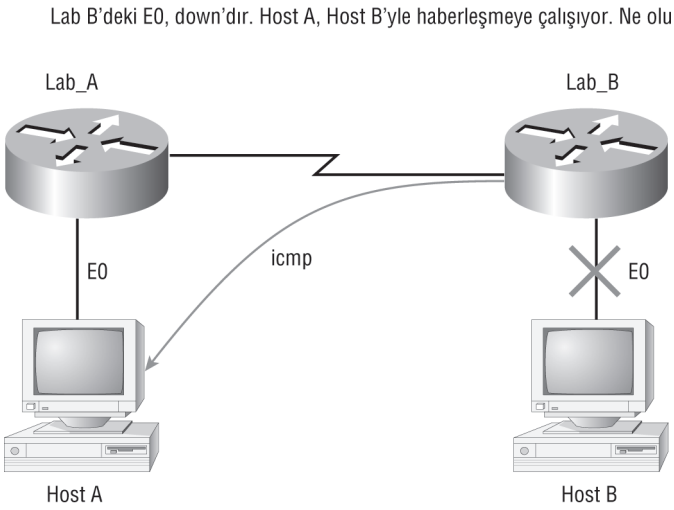
**Destination Unreachable (Hedef erişilemez):** Şayet bir router, artık bir IP datagram gönderemezse, göndericiye, durumunu belirten bir mesaj göndermek için ICMP'yi kullanır. Örneğin, Lab_B router'ının E0 interface'inin down olduğunu gösteren Şekil 2.8'e bir bakalım.

HostA, HostB için hedeflenen bir paket gönderdiğinde, Lab_B router, gönderen cihaza (bu örnekte HostA), bir ICMP hedefe erişilemez mesajı gönderecektir.

**Buffer Full:** Şayet, gelen datagram'ları almak için router'ın arabelleği doluysa, tıkanıklık giderilinceye kadar, bu mesajı göndermek için ICMP'yi kullanacaktır.

**Hops:** Her IP datagram'ı, üzerinden geçmesi için, hop olarak bilinen, belirli sayıda router'a gönderilir. Şayet, hedefine varmadan önce, hop limitine ulaşırsa, bu datagram'ı alan son router, onu

siler. Sonra, cellat router, bir ölüm ilanı mesajı göndermek için ICMP'yi kullanır. Gönderen makineye datagram'ının öldüğünü bildirir.




***Şekil 2.8:*** *ICMP hata mesajı, uzak router'dan, gönderene iletilmektedir.*

**Ping:** Packet Internet Groper (Ping), bir ağ topluluğundaki makinelerin fiziksel ve mantıksal bağlanırlığını kontrol etmek için, ICMP echo request ve replay mesajları kullanır.

NOT *Hem Ping hem de Troceroute (ayrıca sadece Trace olarak kullanılır, Microsoft Windows, tracert kullanır), ağ topluluğunuzda adres ayarlarınızı doğrulamanıza izin verir.*

**Traceroute:** ICMP time-out'lar kullanarak, Traceroute, bir paketin ağ topluluğu boyunca geçtiği yolu bulmak için kullanılmaktadır.

Aşağıdaki veri, ICMP echo request paketini yakalayan bir ağ analizöründen alınmıştır:

```
Flags:          0x00
 Status:         0x00
 Packet Length: 78
 Timestamp:     14:04:25.967000 12/20/03
Ethernet Header
 Destination: 00:a0:24:6e:0f:a8
 Source:      00:80:c7:a8:f0:3d
 Ether-Type:  08-00 IP
IP Header - Internet Protocol Datagram
 Version:              4
 Header Length:        5
 Precedence:           0
 Type of Service:      %000
 Unused:               %00
 Total Length:         60
 Identifier:           56325
 Fragmentation Flags: %000
 Fragment Offset:      0
 Time To Live:         32
 IP Type:              0x01 ICMP
 Header Checksum:      0x2df0
 Source IP Address:    100.100.100.2
```

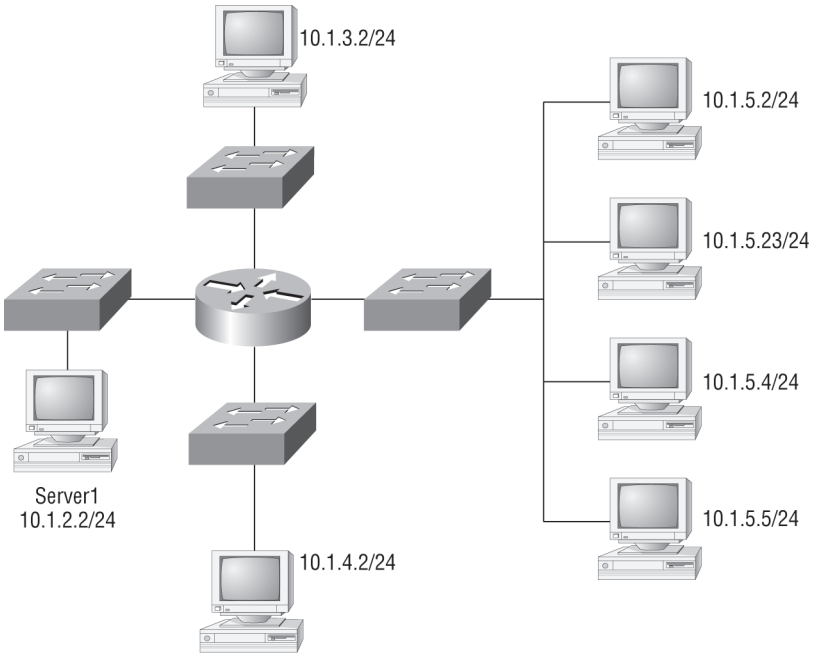
```
 Dest. IP Address:    100.100.100.1
 No Internet Datagram Options
ICMP - Internet Control Messages Protocol
 ICMP Type:       8 Echo Request
 Code:            0
 Checksum:        0x395c
 Identifier:      0x0300
 Sequence Number: 4352
 ICMP Data Area:
 abcdefghijklmnop  61 62 63 64 65 66 67 68 69 6a 6b 6c 6d
 qrstuvwabcdefghi  71 72 73 74 75 76 77 61 62 63 64 65 66
Frame Check Sequence: 0x00000000
```

Olağandışı bir şeyler fark ettiniz mi? ICMP'nin Internet (Network) katmanında çalıştığı halde, Ping isteği yapmak için hala IP'yi kullandığını yakalayabildiniz mi? IP başlığındaki Type alanının 0x01 olması, taşıdığımız verinin, ICMP protokolüne ait olduğunu belirtmektedir. Tüm yolların Roma'ya gitmesi gibi, tüm segment veya verilerin IP'den geçmesi gerektiğini hatırlayın.

NOT

*Ping programı, varsayılan 100 bayt olan bir kargo gibi, paketin veri bölümündeki alfabeyi kullanır. Tabii ki, alfabenin W'de durduğunu, X, Y ile Z'i içermediğini ve tekrar A'dan başladığını düşünen bir Windows cihazından ping'lemezseniz. Şimdi şekle gidin!*

Şayet, Modül 1'deki Data Link katmanı ve farklı frame tipleri hakkında okuduklarınızı hatırlıyorsanız, yukarıdaki çıktıya bakabilir ve bunun hangi Ethernet frame tipi olduğunu söyleyebilirsiniz. Alanlar sadece, hedef donanım adresi, kaynak donanım adresi ve Ether-Type'dır. Yalnız bir Ether-Type alanı kullanan frame, sadece Ethernet_II frame'dir. Fakat ARP protokolü ile ilgilenmeden önce, aktif ICMP'ye tekrar bakalım. Şekil 2.9, bir ağ topluluğunu göstermektedir. (bir router'a sahiptir ve bu nedenle bir ağ topluluğudur değil mi?)



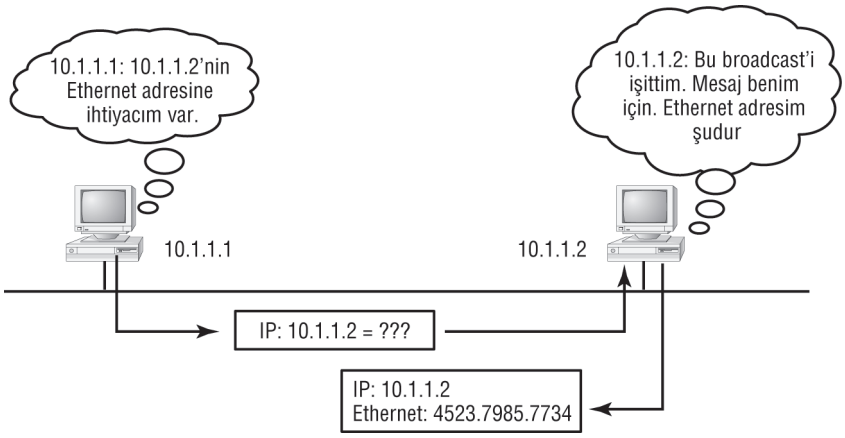

***Şekil 2.9:*** *Aktif ICMP.*

Server1 (10.1.2.2), DOS satırından 10.1.1.5'e bir telnet gönderir. Server1'in bir cevap alacağını düşünüyor musunuz? Server1, Telnet verisini varsayılan ağ geçidine (router'a) göndereceğinden, Router, routing tablosunda 10.1.1.0 ağı olmadığından paketi atacaktır. Bundan dolayı, Server1, ICMP'den bir hedefe ulaşılamaz mesajı alacaktır.

## Address Resolution Protocol (ARP)

Address Resolution Protocol (ARP), bilinen bir IP adresinden bir kullanıcı makinesinin donanım adresini bulur. IP, göndermek için bir datagram'a sahip olduğunda, lokal ağdaki hedefin donanım adresini, Ethernet veya Token Ring gibi bir Network katmanı protokolüne haber vermek zorundadır (üst-katman protokolleri tarafından hedefin IP adresi önceden bildirilmiştir). Şayet IP, ARP cache'inde hedef makinenin donanım adresini bulamazsa, bu bilgiyi bulmak için ARP'ı kullanacaktır.

IP'nin dedektifi gibi, ARP, belirli bir IP adresi ile sorduğu makinenin donanım adresini isteyen bir broadcast göndererek yerel ağı sorgular. Aslında ARP, yazılım (IP) adresini bir donanım adresine (örneğin hedef makinenin Ethernet bord adresine) çevirir ve bundan, adres için broadcast göndererek LAN'daki yeri hakkında sonuç çıkarır. Şekil 2.10 ARP'ın yerel ağa nasıl baktığını göstermektedir.




***Şekil 2.10:*** *Lokal ARP broadcast'i.*

NOT *ARP, IP adreslerini Ethernet (MAC) adreslerine çözümler.*

Aşağıdaki çıktı, bir ARP broadcast'ini göstermektedir: (Hedef donanım adresinin bilinmediğine ve hepsinin F olduğuna dikkat edin, bu bir donanım adres broadcast'idir.)

```
Flags:          0x00

Status:         0x00
  Packet Length: 64
  Timestamp:     09:17:29.574000 12/06/03
Ethernet Header
  Destination:   FF:FF:FF:FF:FF:FF Ethernet Broadcast
  Source:        00:A0:24:48:60:A5
  Protocol Type: 0x0806 IP ARP
ARP - Address Resolution Protocol
  Hardware:                 1 Ethernet (10Mb)
  Protocol:                 0x0800 IP
  Hardware Address Length: 6
  Protocol Address Length: 4
  Operation:                1 ARP Request
  Sender Hardware Address: 00:A0:24:48:60:A5
  Sender Internet Address: 172.16.10.3
  Target Hardware Address: 00:00:00:00:00:00 (ignored)
  Target Internet Address: 172.16.10.10
```

```
Extra bytes (Padding):
 ..... 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A
  0A 0A 0A 0A 0A
 Frame Check Sequence: 0x00000000
```

## Reverse Address Resolution Protocol (RARP)

IP kullanacak makinenin, disksiz olduğu zaman, onun başlangıçta IP adresini bilmesinin bir yolu yoktur. Fakat onun MAC adresini bilir. Reverse Address Resolution Protocol (RARP), Mac adresini ve bu MAC adresine atanmış IP adresi için istek içeren bir paket göndererek disksiz makine için IP adresinin kimliğini tespit eder. RARP sunucu olarak bilinen tanımlı bir makine cevapla döner ve kimlik krizi sona erer. RARP, makinenin IP adresini bulmak ve makinenin ID portresini tamamlamak için bildiği MAC adresi bilgisini kullanır.

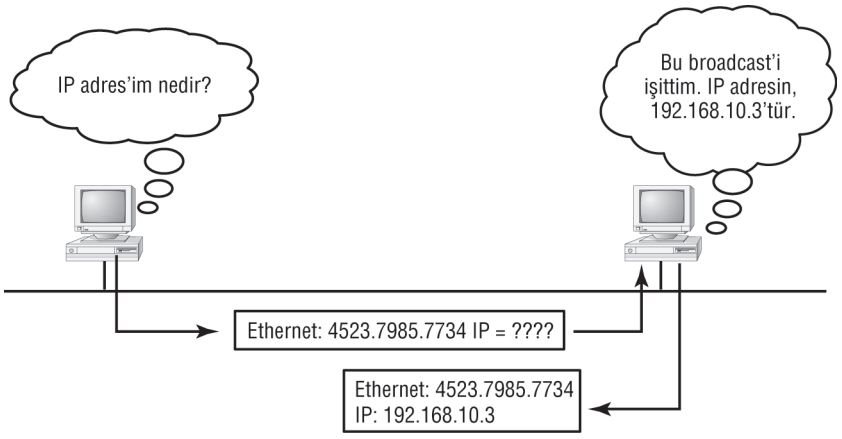
Şekil 2.11, bir RARP broadcast ile IP adresini öğrenmeye çalışan disksiz bir workstation'ı göstermektedir.

*RARP, Ethernet (MAC) adreslerini IP adreslerine çözümler.*

NOT

## Proxy Address Resolution Protocol (Proxy ARP)

Bir ağda, kullanıcı makineleriniz, birden fazla yapılandırılmış varsayılan ağ geçidine sahip olmayabilir. Şunu bir düşünün! Şayet varsayılan ağ geçidi kullanım dışı olursa ne olur? Makine, otomatik olarak başka router'a göndermeye başlamayacaktır (bu makineyi tekrar yapılandırmış olmanız gerekir). Fakat Proxy ARP, gerçekten, routing veya bir varsayılan ağ geçidi dahi yapılandırmadan, bir subnet'teki makinelerin, uzak subnet'lere erişmesine yardımcı olur.




***Şekil 2.11:*** *RARP broadcast örneği.*

Proxy ARP kullanmanın bir avantajı, ortamda bulunan diğer router'ların hepsine routing tablolarını dağıtmaksızın ağdaki tek bir router'a eklenebilmesidir. Fakat Proxy ARP kullanmanın cidden olumsuz bir yanı vardır. Proxy ARP kullanmak, ağ segment'inizdeki trafiğin miktarını çok artıracaktır ve makineler, tüm IP'den MAC adresi eşleşmelerini kullanmak için olağandan daha büyük bir ARP tablosuna sahip olacaktır. Proxy ARP, tüm Cisco router'larda varsayılan olarak yapılandırılmıştır. Şayet onu kullanmayı düşünmüyorsanız, kapatmanız gerekir.

Proxy ARP hakkında son bir düşünce, Proxy ARP'ın, gerçekte ayrı bir protokol olmadığıdır. Subnet'i, uzak cihazlarla paylaştıklarını düşünseler de sorgularından diğer cihazlarla bir router ile ayrılan diğer cihazlar (genelde PC'lerin) adına router'lar tarafından çalıştırılan bir servistir.

*Şayet mali gücünüz yeterse, Proxy ARP yerine, Cisco'nun Hot Standby Router Protocol (HSRP)'sini kullanın. Belki iki veya daha fazla cihaz almanız gerekir, fakat buna değecektir. HSRP hakkında, Cisco web sitesinde daha fazla bilgi bulabilirsiniz.*

NOT

# IP Adresleme

TCP/IP'nin görüşmelerindeki en önemli konulardan birisi, IP adreslemesidir. Bir IP adresi, IP ağında her makineye tahsis edilmiş sayısal bir kimlik belirleyicidir. Ağdaki bir cihazın belirli lokasyonuna tanımlanır.

IP adresi, bir yazılımsal adrestir, bir network interface card (NIC)'e kalıcı-kodlanan ve yerel ağda kullanıcı makinelerini bulmak için kullanılan donanım adresi değildir. IP adreslemesi, bir ağdaki makinelere, makinelerin yer aldıkları LAN tiplerine bakmaksızın diğer ağdaki makinelerle iletişim kurmalarına izin verir.

IP adreslemesinin, daha karmaşık yönlerine geçmeden önce, bazı temel bilgileri anlamanız gerekir. İlk olarak IP adreslemesi ve terminolojisinin bazı temel bilgilerini vereceğim. Sonra, hiyerarşik IP adres düzenlemesi ve özel IP adreslerini öğreneceksiniz.

## IP Terminolojisi

Bu modül boyunca, Internet protokolünü anlamanızda çok önemli bazı terimler öğreneceksiniz. Başlamanız için burada birkaç örnek var:

**Bit:** Bir bit, 1 veya 0 olan bir sayıdır.

**Byte:** Bir byte, kullanılan pariteye bağlı olarak, 7 veya 8 bit'tir. Bu modülün kalanında, bir byte'ı 8 bit olarak düşüneceğiz.

**Octet:** 8 bit'ten oluşan bir oktet, sıradan 8-bit binary bir numaradır. Byte ve octet terimleri, bu modülde tamamıyla birbirinin yerine kullanılabilir.

**Network adresi:** Bu, routing'de, paketleri uzak bir ağa göndermek için kullanılan uygulamadır. Örneğin, 10.0.0.0, 172.16.0.0 ve 192.168.10.0.

**Broadcast adresi:** Uygulamalar ve kullanıcı makineleri tarafından, ağdaki tüm düğümlere bilgi göndermek için kullanılan bu adres, broadcast adresi olarak tanımlanmaktadır. Örneğin, 255.255.255.255, tüm network ve düğümleri içerir, 172.16.255.255, 172.16.0.0 ağındaki tüm network'leri belirtir ve 10.255.255.255, 10.0.0.0 ağındaki tüm subnet ve kullanıcılar için broadcast adresidir.

## Hiyerarşik IP Adresleme Planlaması

32-bit bilgiden oluşan bir IP adresidir. Bu bit'ler, her biri 1 byte (8 bit) içeren ve octet veya byte olarak belirtilen, dört bölüme ayrılmıştır. Şu üç yöntemi kullanarak bir IP adresini anlatabilirsiniz:

- 172.16.30.56 'de olduğu gibi noktalı-ondalık,
- 10101100.00010000.00011110.00111000 'de olduğu gibi binary (ikilik sayı düzeni),
- AC.10.1E.38 'de olduğu gibi hexadecimal olarak

Tüm bu örnekler gerçekte aynı IP adresini belirtir. IP adresinden bahsedildiğinde hexadecimal, decimal ve binary kadar sık kullanılmaz, fakat hala bazı programlarda hexadecimal saklanan IP adresleri bulabilirsiniz. Window registery, bir makinenin IP adresini hex olarak saklayan programlara iyi bir örnektir.

32-bit IP adresi, düz veya hiyerarşik olmayan adreslerin tersine, yapısal veya hiyerarşik bir adrestir. Her iki tip adresleme sistemi kullanıldığı halde, hiyerarşik adresleme, iyi bir nedenle seçilmiştir. Bu sistemin avantajı, çok fazla sayıda (4.3 milyon) adresin kullanılabilmesidir. (her durum için iki uygun değer(0 veya 1) ile 32-bit adres alanı size $2^{32}$ veya 4,294,967,296'yı verir). Düz adresleme sisteminin dezavantajı ve IP adreslemesinde kullanılmamasının sebebi routing'le ilgilidir. Şayet her adres, eşsiz olursa, İnternetteki tüm router'lar, internetteki tüm makinelerin adresini saklamak zorunda olacaktı. Uygun adreslerin sadece bir bölümü dahi kullanılsa, bu, etkili routing'i imkansız kılar.

Bu iki veya üç-seviyeli sistem, bir telefon numarasıyla karşılaştırılabilir. İlk bölüm yani alan kodu, çok geniş bir alanı belirtir. İkinci bölüm, yani prefix, kapsamı, yerel arama alanına daraltır. Son segment, yani müşteri numarası belirli bağlantıya odaklanır. IP adresleri, aynı tip katmanlı yapıyı kullanır. Tüm 32 bit'in, eşsiz bir tanımlayıcı olarak işlem görmesindense, düz adreslemede, adresin bir bölümü network adresi, diğer bölümü hem subnet hem de kullanıcı makinesi veya sadece host adresi olarak tanımlanmaktadır.

Aşağıdaki bölümlerde, IP ağ adreslemesi ve ağlarımızda kullanabileceğimiz farklı adres sınıflarından bahsedeceğim.

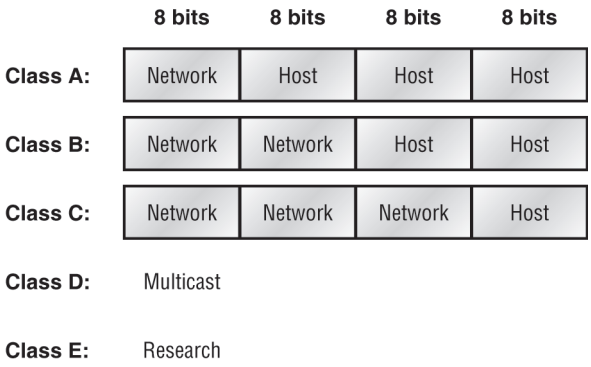
## Network Adreslemesi

Network adresi (network numarası olarak ta belirtilebilir), her ağı eşsiz olarak tanımlar. Aynı ağdaki her makine, IP adresinin bir parçası olarak network adresini paylaşır. Örneğin, 172.16.30.56 IP adresinde, 172.16 network adresidir.

Düğüm adresi, ağdaki her makineye atanır ve her network'te eşsiz olarak tanımlanır. Bu adres bölümü eşsiz olmalıdır, çünkü bir ağın (grubun) tersine belirli bir makineyi (bireyi) tanımlar. Bu numara ayrıca bir host adresi olarak ta belirtilebilir. 172.16.30.56 IP adresi örneğinde, 30.56 düğüm adresidir.

İnternet tasarımcıları, ağın boyutuna göre ağ sınıfları oluşturmaya karar verdiler. Çok sayıda düğümü kontrol eden az sayıda network için Klas A network mertebesini oluşturdular. Diğer kenardaki, az sayıda düğümle çok sayıda network için ayrılan Klas C network'üdür. Çok geniş ve çok küçük network'lerin arası ağlar için sınıf ayrımı, Klas B network olarak belirtilmektedir.

Bir IP adresini network ve host adresine bölmek, bir ağın sınıf atamasıyla belirlenir. Şekil 2.12, bu modül boyunca detaylı bir şekilde açıklayacağım bir konu olan ağın üç sınıfını özetlemektedir.




***Şekil 2.12:*** *Üç Network sınıfının özeti.*

Etkili routing'den emin olmak için internet tasarımcıları, her network sınıfına adresin öncü-bit bölümü için bir yetki tanımladılar. Örneğin, bir router, Klas A bir ağ adresinin daima 0'larla başlayacağını bildiğinden router, onun adresinin sadece ilk bit'ini okuduktan sonra, paketi yoluna gönderebilir. Bu, adres planının, Klas A, Klas B ve Klas C adresleri arasındaki farkın tanımlandığı yerdir. Sonraki bölümde, Klas D ve Klas E adreslerinin işlenmesine devam ederek bu üç sınıf arasındaki farklılıklardan bahsedeceğim.(Klas A, B ve C, ağlarımızda sadece kullanıcı makinelerine adres vermek için kullanılan adres aralıklarıdır.)

### Network Adres Aralığı: KlasA

IP adres planı tasarımcıları, bir Klas A network adresindeki ilk byte'ın ilk bit'inin daima 0(sıfır) olması gerektiğini söylerler. Bunun anlamı, bir KlasA adresinin, ilk byte'ta mutlaka 0 ve 127 arasında olması gerektiğidir.

Aşağıdaki network adresleri üzerinde düşünün:

**0xxxxxxx**

Şayet diğer 7 bit'in hiçbirini kullanmazsanız veya hepsini kullanırsanız, Klas A network adres aralığını bulursunuz:

**00000000 = 0**

**01111111 = 127**

Böylece, bir KlasA ağı, 0 ile 127 arasında (daha az veya çok olamaz) ilk oktette tanımlanmıştır. (Evet, 0 ve 127'nin KlasA network'ünde tanımlı olmadığını biliyorum. Rezerve edilmiş adreslerden birazdan bahsedeceğim.)

#### Network Adres Aralığı: KlasB

KlasB network'lerde, RFC'ler, ilk byte'ın daima kullanılmasını, fakat ikincinin kullanılmaması gerektiğini söylerler. Şayet diğer 6 bit'in hiçbirini kullanmazsanız veya tamamını kullanırsanız, KlasB network adres aralığını bulursunuz:

```
10000000 = 128
10111111 = 191
```

Görebileceğiniz gibi, bir KlasB network'ü, ilk byte, 128'den 191'e yapılandırıldığında tanımlanmaktadır.

#### Network Adres Aralığı: KlasC

KlasC ağlar için, RFC'ler, ilk oktet'in ilk iki bit'inin kullanıldığı, fakat üçüncü bit'inin kullanılmadığını tanımlamaktadır. Devamı, önceki sınıflarla aynı prosestir. Aralığı bulmak için binary'den ondalık sayıya çevrilir. KlasC network için aralık şudur:

```
11000000 = 192
11011111 =  223
```

Şayet, 192 den başlayıp 223'e giden bir IP adres görürseniz, bunun bir KlasC adres olduğunu bileceksiniz.

#### Network Adres Aralığı: KlasD ve E

224 ve 255 arası adresler, KlasD ve KlasE network'ler için rezerve edilmiştir. KlasD (224-239), multicast adresler için KlasE (240-255), bilimsel amaçlarla kullanılmaktadır. Bu kitapta bu tip adreslerden bahsetmeyeceğim (ve siz de onları bilmek zorunda değilsiniz).

#### Network Adresleri: Özel Amaçlı

Bazı IP adresleri, özel amaçlar için ayrılmıştır. Tablo 2.4 bu seçkin küçük kulübün elemanlarını ve sebep ilişkilerini listelemektedir.

**Tablo 2.4:** Rezerve IP Adresleri

| Adres | Fonksiyon |
|---|---|
| Tamamı 0 olan network adresi | Bu network veya segment" anlamında çevrilir. |
| Tamamı 1 olan network adresi | "Tüm network'ler" anlamında çevrilir. |
| Ağ 127.0.0.1 | Loopback testleri için ayrılmıştır. Lokal düğümü tanımlar ve ağ trafiği oluşturmaksızın bu hostun kendine bir test paketi göndermesine izin verir. |
| Tamamı 0 olan host adresi | "Network adresi" veya belirli ağdaki herhangi bir kullanıcı anlamında çevrilir. |
| Tamamı 1 olan host adresi | "Belirli ağdaki "tüm hostlar" olarak çevrilir. Örneğin, 128.2.255.255, 128.2 (KlasB adresi) ağındaki "tüm hostlar" anlamına gelir. |
| IP adresinin tamamının 0'a ayarlanması | Default route'u belirtmek için router'larda kullanılır. |
| IP adresinin tamamının 1'e ayarlanması (255.255.255.255 şeklinde) | Mevcut ağdaki tüm host'lara broadcast demektir. Bazen tüm 1'lerin broadcast'i veya limitli broadcast olarak bilinir. |

## KlasA Adresler

Bir KlasA network adresinde, ilk byte, network adresine atanmıştır ve kalan üç byte, host adresleri için kullanılmaktadır. KlasA formatı, şu şekildedir:

```
network.node.node.node
```

Örneğin, 49.22.102.70 IP adresinde 49, network adresidir ve 22.102.70, host adresidir. Bu belirli ağdaki her makine, 49'un özel network adresine sahip olacaktır.

KlasA network adresleri, ilk bit'i rezerve ve kalan 7 bit'i kullanmak (adresleme) için uygun olan, 1 byte uzunluğundadır. Sonuç olarak, oluşturulabilen KlasA network'lerinin maksimum sayısı 128'tir. Neden? Çünkü her bir 7 bit'in konumu, 0 veya 1 olabilir, bu sebeple $2^7$ veya 128 'dir.

Durumu biraz daha karmaşık hale getirelim, tamamı 0'lardan oluşan network adresi (0000 0000), default route'ı tanımlamak için ayrılmıştır. (önceki bölümdeki tablo 2.4'e bakın). İlave olarak, sistem kontrolü için rezerve edilen 127 adresi de kullanılamaz. Siz gerçekte, KlasA network adreslerine tanımlamak için sadece 1 ve 126 arasındaki numaraları kullanabilirsiniz. Yani, gerçek kullanılabilir KlasA network adres numarası, 128 eksi 2, 126'dır.

*127.0.0.1 IP adresi, tek bir düğümdeki IP yığınını test etmek için kullanılmaktadır ve geçerli bir host adresi olarak kullanılamaz.*

**NOT**

Her KlasA adresi, makinenin düğüm adresi için 3 byte'dır (24-bit şeklinde). Yani, $2^{24}$ veya 16,777,216 eşsiz kombinasyonudur ve bu yüzden, her KlasA network'ü için kesinlikle birçok uygun eşsiz host adresi vardır. Başka deyişle bu, bir network segment'inde çok sayıda kullanıcı makinesi olması demektir!

### KlasA Geçerli Host ID'leri

Burada, bir KlasA network adresindeki geçerli host ID'lerini nasıl anlayacağımızla ilgili bir örnek vardır:

- Tüm host bit'lerinin 0 olması, network adresi demektir: 10.0.0.0.
- Tüm host bit'lerinin 1 olması, broadcast adresi demektir: 10.255.255.255.

Geçerli host adresleri, network ve broadcast adresleri arasındaki numaralardır: 10.0.0.1 2den 10.255.255.254. 0 ve 255'lerin, geçerli host ID'leri olabileceğine dikkat edin. Geçerli host adreslerini bulmaya çalıştığınızda hatırlamanız gereken şey, host bit'lerinin hepsinin kullanılamaz olamayacağı veya aynı anda hepsinin kullanılamayacağıdır.

## KlasB Adresleri

Bir KlasB network adresinde, ilk 2 byte, network adresine atanır ve kalan 2 byte düğüm adresleri için kullanılmaktadır. Format aşağıdaki gibidir:

```
network.network.node.node
```

Örneğin, 172.16.30.56 IP adresinde, 172.16 network adresi ve 30.56 düğüm adresidir.

2 byte olan (her biri 8 bit) network adresiyle, $2^{16}$ eşsiz kombinasyon olabilir. Fakat İnternet tasarımcıları, tüm KlasB adreslerinin, binary 1 ve sonra 0 sayısı ile başlaması gerektiğine karar verdiler. Bu, dağıtmak için 14 bit konumu, o da 16,284 ($2^{14}$) eşsiz KlasB network adresi bırakır.

Bir KlasB adresi, host adresleri için 2 byte kullanır. Bu, $2^{16}$ 'dan 2 rezerve modelin (tüm 0'ların ve 1'lerin) çıkartılmasıdır ve her KlasB network'ü için toplam 65,534 uygun host adresi demektir.

### KlasB Geçerli Host ID'leri

Burada, bir KlasB network adresindeki geçerli host ID'lerini nasıl bulacağımızla ilgili bir örnek vardır:

- Tüm host bit'lerinin 0 olması, network adresi demektir: 172.16.0.0.
- Tüm host bit'lerinin 1 olması, broadcast adresi demektir: 172.16.255.255.

Geçerli host adresleri, network ve broadcast adresleri arasındaki numaralardır: 172.16.0.1'den 172.16.255.254'e kadar.

### KlasC Adresleri

KlasC network adreslerinin ilk üç byte'ı adresin network bölümü için tanımlanır ve sadece 1 yetersiz byte, host adresleri için kalır. Format şu şekildedir:

```
network.network.network.node
```

192.168.100.102 IP adresi örnek olarak kullanıldığında, 192.168.100 network adresi, 102 host adresidir.

Bir KlasC network adresinde, ilk üç bit yerleşimi, daima binary olarak 110 şeklindedir. Hesaplama şu şekildedir: 3 byte veya 24 bit'ten 3 rezerve yerleşim çıkartılırsa, 21 yerleşim kalır. Bu nedenle, $2^{21}$ veya 2,097,152 uygun KlasC network'ü vardır.

Her eşsiz KlasC adresi, host adreslerine kullanmak için 1 byte'a sahiptir. $2^8$ veya 256'dan 2 rezerve örnek (tümü 0 veya tümü 1 olan) çıkartılırsa, her KlasC network'üne toplamda 254 host adresi kalır.

#### KlasC Geçerli Host ID'leri

Burada, bir KlasC network adresindeki geçerli host ID'lerini nasıl bulacağımızla ilgili bir örnek vardır:

- Tüm host bit'lerinin 0 olması, network adresi demektir: 192.168.100.0
- Tüm host bit'lerinin 1 olması, broadcast adresi demektir: 192.168.100.255

Geçerli host adresleri, network ve broadcast adresleri arasındaki numaralardır: 192.168.100.1'den 192.168.100.254'e kadar.

## Özel IP Adresleri

IP adresleme planını geliştiren insanlar, özel IP adresi olarak belirttiğimiz IP adreslerini de oluşturdular. Bu adresler özel bir ağda kullanılabilir ve internete route edilemezler. Bu, çok gerekli güvenlik düzenlemesi oluşturmak amacıyla tasarlanmıştır, ayrıca uygun olarak bir IP aralığı da kazandırır.

Şayet ağdaki her kullanıcı makinesi, gerçek route edilebilir IP adresine sahip olmak zorunda olsaydı, biz IP adresi dağıtımını yıllar önce tamamlamış olurduk. Fakat özel IP adresi kullanarak, ISP'ler, firmalar ve ev kullanıcıları nispeten küçük bir IP adres grubunu kendi ağlarını internete bağlamak ihtiyacıyla kullanırlar. Bu ekonomiktir, çünkü özel IP adreslerini kendi iç network'lerinde kullanabilirler ve kolayca kurabilirler.

Bunu başarmak için ISP ve firma (son kullanıcı, kim olduğu önemli değil), Network Address Translation (NAT) olarak bilinen bir hizmeti kullanmak zorundadır. NAT, esasen özel bir IP adresini alır ve onu internette kullanılması için çevirir (NAT, "Network Address Translation" başlıklı bölüm 10'da işlenmektedir). Birçok kişi, aynı gerçek IP adresini, İnternet'e ulaşmak için kullanabilir. Bunu yaparak, çok sayıda adres aralığı kazanmış oluruz ve bu hepimiz için çok iyidir.

### Hangi Özel IP Adresini Kullanmalıyım?

*Bu gerçekten iyi bir soru: Ağınızı kurduğunuzda KlasA, KlasB veya KlasC özel adreslemeden hangisini kullanmalıyım? SF'daki Acme Corporation örneğine bir bakalım. Bu firma yeni bir binaya taşınıyor ve tamamen yeni bir ağa ihtiyacı var. Her birinde 70 kullanıcısı olan 14 departmanı var. Muhtemelen, kullanıcıları bir veya iki KlasC adresine sıkıştırabilir veya bir KlasB adresi hatta eğlence olsun diye bir KlasA adresi bile kullanabilirsiniz.*

*Danışmanlık dünyasındaki temel ilke, ne kadar küçük olduğuna bakılmaksızın bir şirket network'ü kurduğunuzda, KlasA network adresi kullanmanızdır. Çünkü bu size çok büyük esneklik ve büyüme imkânı verir. Örneğin, /24 mask ile 10.0.0.0 network adresini kullanıyorsanız, 65,536 network'e ve her birinde 254 host'a sahip olursunuz.*

*Fakat bir ev network'ü kuruyorsanız, KlasC adresi seçmelisiniz. Çünkü insanların anlaması ve yapılandırması için en basiti odur. Varsayılan KlasC adresi kullanmak size 254 kullanıcılı bir network sağlar.*

*Acme Corporation için, /24 mask ile 10.1.x.0 (x, her departman için subnet'tir), tasarımı, kurulumu ve hata tespitini kolaylaştırır.*

Rezerve adresler, Tablo 2.5'te listelenmiştir.

**Tablo 2.5:** Rezerve IP Adres Aralığı

| Klas Adresi | Rezerve Adres Aralığı |
|---|---|
| Klas A | 10.0.0.0'dan 10.255.255.255'e |
| Klas B | 172.16.0.0'dan 172.31.255.255'e |
| Klas C | 192.168.0.0'dan 192.168.255.255'e |

# Broadcast Adresleri

*Özel adres aralığınızı bilmek zorundasınız!* NOT

Birçok insan broadcast terimini genel bir terim olarak kullanır ve çoğu zaman ne demek istediklerini anlarız. Fakat her zaman değil. Örneğin, "Host, bir router yardımıyla DHCP'ye broadcast gönderdi" diyebilirsiniz. Fakat bunun gerçekleşmesi neredeyse olanaksızdır. Muhtemelen demek istediğiniz, "Host bir IP adresi için broadcast gönderdi, daha sonra bir router, DHCP sunucusuna bunu bir unicast paketi olarak gönderdi"dir. IPv4 ile broadcast'lerin çok önemli olduğunu, IPv6 ile hiç broadcast gönderilmediğini hatırlayın. (Şimdi, Bölüm 13'e geldiğinizde sizi heyecanlandıracak şeyler vardır).

Bölüm 1 ve 2 boyunca broadcast adreslerine işaret ettim, hatta bazı örneklerde verdim. Fakat gerçekten, farklı terimlere ve onlarla ilgili kullanımlara henüz girmedim. Burada, size açıklamak istediğim dört farklı broadcast tipi vardır:

**Katman2 broadcast'ler:** Bunlar, bir LAN'daki tüm host'lara gönderilir.

**Broadcast'ler (katman3):** Bunlar, network'teki tüm host'lara gönderilirler.

**Unicast:** Bunlar tek bir hedef kullanıcı makinesine gönderilir.

**Multicast:** Bunlar, tek bir kaynaktan gönderilen ve farklı ağlardaki birçok cihaza iletilen paketlerdir.

İlk olarak, donanım broadcast'leri olarak da bilinen katman 2 broadcast'lerini anlayın. Sadece bir LAN'da gönderilir ve LAN sınırını (router'ı) geçmezler. Tipik bir donanım adresi, 6 byte'dır (48 bit) ve 0c.43.a4.f3.12.c2 gibi görünür. Broadcast, binary olarak hepsi 1'lerden, hexadecimal'de tamamı F'lerden (FF.FF.FF.FF.FF.FF şeklinde) oluşur.

Sonra, katman 3'te geleneksel broadcast adresleri vardır. Broadcast mesajları, bir broadcast domain'indeki tüm kullanıcı makinelerine ulaşmak için gönderilir. Bunlar, tüm host bit'lerinin kullanıldığı network broadcast'leridir. Burada sizin zaten aşina olduğunuz bir örnek var: 172.16.0.0 255.255.0.0 network adresi, 172.16.255.255 (tüm host bit'lerinin 1 olduğu) bir broadcast adresine sahiptir. Broadcast'ler, 255.255.255.255 olarak belirtilen "tüm network ve host'lar" da olabilir. Broadcast mesajına güzel bir örnek, bir Address Resolution Protocol (ARP) isteğidir. Host bir pakete sahip olduğunda, hedefin mantıksal (IP) adresini bilir. Paketi hedefe ulaştırmak için hedef farklı bir IP ağında bulunuyorsa, host, paketi varsayılan ağ geçidine göndermek zorundadır. Frame'i göndermek için ihtiyacı olan MAC adresine sahip olmadığından, kaynak broadcast domain'indeki her cihazın dinleyeceği bir broadcast gönderir. Aslında bu broadcast, uygun bilgi veren kaynak tarafından "Şayet, 192.168.2.3 IP adresinin sahibiysen, lütfen MAC adresini bana ilet" der.

Unicast farklıdır, çünkü o, 255.255.255.255'ten gerçek bir IP adresine giden broadcast paketidir. Başka bir deyişle, o belirli bir kullanıcı makinesine gönderilir. DHCP istemci isteği, unicast'in na-

sıl çalıştığına güzel bir örnektir. Şu örneğe bakalım: LAN'daki bir kullanıcınız, bir FF.FF.FF.FF.FF.FF katman 2 broadcast'i ve LAN'da bir DHCP sunucu aramak için 255.255.255.255 katman 3 hedef broadcast'i gönderir. Hedef port numarası 67 (BootP sunucusu) olduğundan, router, bunun bir DHCP için hedeflenen broadcast olduğunu anlayacak ve başka LAN'da olan DHCP sunucusunun IP adresine gönderecektir. Böylece, aslında DHCP sunucunuzun IP adresi 172.16.10.1 ise host'unuz, sadece 255.255.255.255 DHCP istemci broadcast'i isteği gönderir ve router bu broadcast'i 172.16.0.1 özel adresi olarak değiştirir. (Bu bir varsayılan servis olmadığından router'ın bu servisi sağlaması için her interface'ini IP `helper-address` komutu ile yapılandırmanız gerekmektedir.)

Multicast tamamıyla farklı bir iletişimdir. İlk bakışta, unicast ve broadcast iletişimin bir karışımı olarak görünür. Fakat tamamıyla aynı şey değildir. Multicast, broadcast'lere benzeyen point-to-multipoint iletişime izin verir. Fakat farklı şekilde olur. Multicast'in özü, bir broadcast domain'indeki tüm kullanıcılara mesajları dağıtmadan, çok sayıda alıcının mesajları almasını sağlamasıdır.

Multicast, IP multicast grup adreslerine mesajları veya veriyi göndererek çalışır. Daha sonra router, paketin kopyalarını bu grup adresine üye kullanıcılara sahip tüm interface'lerden gönderir. Burası, multicast iletişimle, teoride paketlerin kopyalarının sadece üye makinelere gönderildiği, multicast'in broadcast mesajlarından farklı olduğu noktadır. Teoride bunun anlamı şudur; örneğin host, 224.0.0.9 için hedeflenen bir multicast paket alacaktır (bu bir EIGRP paketidir ve sadece EIGRP çalışan bir router bunu okuyacaktır). Broadcast LAN'ındaki (Ethernet, bir broadcast multiaccess LAN teknolojisidir) tüm kullanıcı makinesi, frame'i yakalayacak, hedef adresini okuyacak ve multicast grupta olmayınca, frame'i atacaklardır. Bu, LAN bant genişliğini değil de, PC işlemcisini korumaktadır. Dikkatli oluşturulmadığında Multicasting, farklı şekillerde güçlü LAN tıkanıklığına sebep olabilir.

Kullanıcı ve uygulamaların üye olabilecekleri farklı gruplar vardır. Multicast adres aralığı, 224.0.0.0 ile başlar ve 239.255.255.255'e kadar gider. Görebileceğiniz gibi, bu adres aralığı, classful IP atamasına göre IP KlasD adres aralığında kalır.

# Özet

Şayet her şeyi ilk seferde bu kadar ilerletip anlayabiliyorsanız, kendinizle gurur duymalısınız. Bu bölümde gerçekten çok sayıda şey gördük. Fakat bu bölümdeki bilgilerin, kitabın geri kalanını takip etmek için anahtar olduğunu bilmelisiniz. İlk seferinde tamamıyla anlamadıysanız bile bunu sorun yapmayın. O, bu bölümü bir defadan fazla okumaktan daha fazla canınızı acıtmayacaktır. Hala kavramak için çok sayıda konu var ve bu nedenle konuların hepsini anladığınızdan ve fazlasına hazır olduğunuzdan emin olmalısınız. Bizim yaptığımız, bir temel inşa etmektir. Siz de, güçlü bir temel olmasını istersiniz, değil mi?

DoD modeli, katmanlar ve ilgili protokoller hakkında bilgi aldıktan sonra, çok önemli olan IP adreslemesini öğrendiniz. Bölüm 3'e geçmeden anlaşılması oldukça önemli olan, klas adresleri arasındaki farklılıklar ile network adresi, broadcast adresi ve geçerli host aralıklarının nasıl bulunacağını detaylı bir şekilde işledim.

Buraya kadar geldikten sonra, durmak ve tüm bu beyin dalgalarını ve sinir hücrelerini çöpe atmak için bir sebep yok. Öyleyse, durmayın, bu bölüm sonundaki yazılı lab ile gözden geçirme sorularına gidin ve her cevabın açıklamasını anladığınızdan emin olun.

# Sınav Temelleri

**Process/Application katman protokollerini hatırlamak:** Telnet, size uzak bir makineye bağlanmanızı ve programları çalıştırmayı sağlayan terminal emülasyon programıdır. File Transfer Protocol (FTP), size dosya transferi sağlayan, connection-oriented bir programdır. Trivial FTP (TFTP), connectionless bir dosya transfer programıdır. Simple Mail Transfer Protocol (SMTP), mail gönderme programıdır.

**Host-to-host katman protokollerini hatırlamak:** Transmission Control Protocol (TCP), acknowledgement ve akış kontrolü kullanarak güvenli network servisleri sağlayan, connection-oriented bir protokoldür. User Datagram Protocol (UDP), düşük ek yük sağlayan ve güvenilir kabul edilmeyen, connectionless bir protokoldür.

**Internet katmanı protokollerini hatırlamak:** Internet Protocol (IP), network adresi ve ağ topluluğu boyunca routing sağlayan, connectionless bir protokoldür. Address Resolution Protocol (ARP), bilinen bir IP adresinden bir donanım adresi bulur. Reverse ARP (RARP), bilinen donanım adresinden bir IP adresi bulur. Control Message Protocol (ICMP), sistem kontrolü ve hedefe erişilemez mesajı sağlar.

**KlasA aralığını hatırlamak:** KlasA network için IP aralığı, 1-126'dır. Bu, varsayım olarak, 8 bit network ve 24 bit host adreslemesi sağlar.

**KlasB aralığını hatırlamak:** KlasB network için IP aralığı, 128-192'dir. Bu, varsayım olarak, 16 bit network ve 16 bit host adreslemesi sağlar.

**KlasC aralığını hatırlamak:** KlasC network için IP aralığı, 192-223'dür. Bu, varsayım olarak, 24 bit network ve 8 bit host adreslemesi sağlar.

**Özel IP aralıklarını hatırlayın:** KlasA özel IP adres aralığı, 10.0.0.0'dan 10.255.255.255'e kadardır. KlasB özel IP adres aralığı, 172.16.0.0'dan 172.16.255.255'e kadardır. KlasC özel IP adres aralığı, 192.168.0.0'dan 192.168.255.255'e kadardır.

# Yazılı Lab 2

TCP ile ilgili aşağıdaki soruları cevaplayın:

1. KlasC adres aralığı, ondalık ve binary sayı olarak nedir?
2. DoD modelinin hangi katmanı, OSI modelinin Transport katmanına denk gelmektedir?
3. KlasA network adresinin geçerli aralığı nedir?
4. 127.0.0.1 adresi ne için kullanılır?
5. Listelenmiş bir IP adresinden, network adresini nasıl bulursunuz?
6. Listelenmiş bir IP adresinden, broadcast adresini nasıl bulursunuz?
7. KlasA özel IP adres aralığı nedir?
8. KlasB özel IP adres aralığı nedir?
9. KlasC özel IP adres aralığı nedir?
10. Hexadecimal adreslemede kullanabileceğiniz tüm karakterler nelerdir?

*(Yazılı Lab2 'nin cevaplarını, bu bölüm için gözden geçirme sorularının cevaplarından sonra bulabilirsiniz.)*

# Gözden Geçirme Soruları

NOT

*Aşağıdaki sorular, bu modülün materyallerini anlayıp anlamadığınızı test etmek için tasarlanmıştır. İlave sorulara nasıl sahip olacağınızla ilgili daha fazla bilgi almak için, lütfen bu kitabın Önsözüne bakınız.*

1. 1. Binary 10011101 sayısının, ondalık ve hexadecimal karşılığı nedir? (iki şık seçin)
   A. 159
   B. 157
   C. 185
   D. 0x9D
   E. 0xD9
   F. 0x159
2. Aşağıdakilerden hangisi, bir router'a, uzak bir host için planlanan bir ARP isteğine cevap sağlar?
   A. Gateway DP
   B. Reverse ARP (RARP)
   C. Proxy ARP
   D. Inverse ARP (IARP)
   E. Address Resolution Protocol (ARP)
3. IP adresi, subnet maskı, varsayılan ağ geçidi ve DNS bilgisi içeren IP yapılandırmasını otomatikleştiren bir mekanizma oluşturmak istiyorsunuz. Bunu gerçekleştirmek için hangi protokolü kullanacaksınız?
   A. SMTP
   B. SNMP
   C. DHCP
   D. ARP
4. Lokal bir cihazın donanım adresini bulmak için hangi protokol kullanılır?
   A. RARP
   B. ARP
   C. IP
   D. ICMP
   E. BootP
5. Aşağıdakilerden hangileri TCP/IP modelindeki katmanlardır? (Üç şık seçin.)
   A. Application
   B. Session
   C. Transport
   D. Internet
   E. Data Link
   F. Physical

6. Hangi IP adres sınıfı, her network ID için maksimum 254 host adresi sağlar?
   A. Klas A
   B. Klas B
   C. Klas C
   D. Klas D
   E. Klas E
7. Aşağıdakilerden hangisi DHCP discover mesajını tanımlamaktadır? (iki şık seçin)
   A. Bir katman 2 broadcast'i gibi FF:FF:FF:FF:FF:FF kullanır.
   B. Transport katmanı protokolü gibi UDP kullanır.
   C. Transport katmanı protokolü gibi TCP kullanır.
   D. Katman 2 hedef adresi kullanmaz.
8. Bir Telnet bağlantısı için hangi katman 4 protokolü kullanılmaktadır?
   A. IP
   B. TCP
   C. TCP/IP
   D. UDP
   E. ICMP
9. ICMP paketleri hakkında hangi açıklamalar doğrudur? (İki şık seçin.)
   A. Bir TCP segment alındısını onaylarlar.
   B. Datagram teslimini garanti ederler.
   C. Kullanıcı makinelerine, ağ problemleri hakkında bilgi sağlarlar.
   D. IP datagram'larına enkapsüle edilirler.
   E. UDP datagram'larına enkapsüle edilirler.
10. Aşağıdaki servislerin hangisi TCP kullanır? (Üç şık seçin.)
    A. DHCP
    B. SMTP
    C. SNMP
    D. FTP
    E. HTTP
    F. TFTP
11. Aşağıdaki servislerin hangisi UDP kullanır? (Üç şık seçin.)
    A. DHCP
    B. SMTP
    C. SNMP
    D. FTP
    E. HTTP
    F. TFTP

12. Aşağıdakilerden hangileri, OSI modelinin Application katmanında kullanılan TCP/IP protokolleridir? (üç şık seçin)

    A. IP

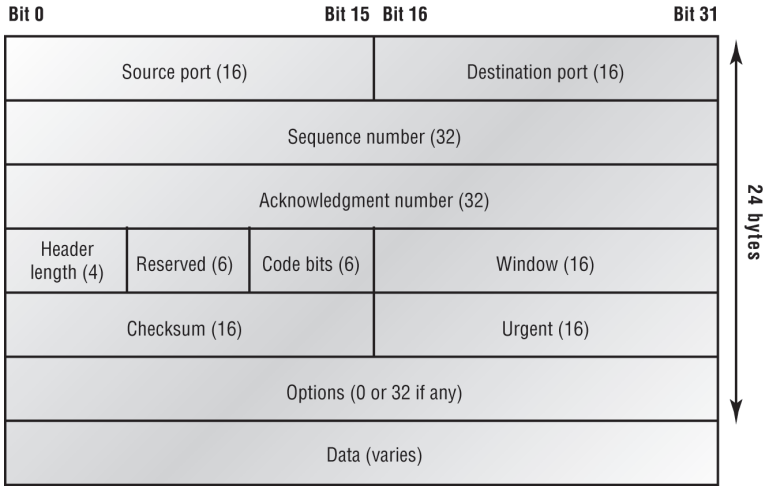
    B. TCP

    C. Telnet

    D. FTP

    E. TFTP

13. Aşağıdaki resim, bir veri yapısı başlığını göstermektedir. Bu başlık hangi protokol'den gelmektedir?




    A. IP

    B. ICMP

    C. TCP

    D. UDP

    E. ARP

    F. RARP

14. Şayet, Telnet veya FTP kullanıyorsanız, hangisi veri iletmek için kullandığınız en yüksek katmandır?

    A. Application

    B. Presentation

    C. Session

    D. Transport

15. DoD modeli (ayrıca TCP/IP yığını olarak da bilinir), dört katmandır. DoD modelinin hangi katmanı, OSI modelinin Network katmanına eşdeğerdir?

    A. Application

    B. Host-to-Host

    C. Internet

    D. Network Access

16. Aşağıdakilerden hangisi özel IP adresidir?
    A. 12.0.0.1
    B. 168.172.19.39
    C. 172.20.14.36
    D. 172.33.194.30
    E. 192.168.24.43
17. TCP/IP yığınındaki hangi katman, OSI modelinin Transport katmanına denk gelir?
    A. Application
    B. Host-to-Host
    C. Internet
    D. Network Access
18. ICMP paketleriyle ilgili hangi ifadeler doğrudur?
    A. ICMP, datagram teslimini garanti eder.
    B. ICMP, ağ problemleri hakkında host'lara bilgi sağlar.
    C. ICMP, IP datagram'larına enkapsüle edilir.
    D. ICMP, UDP datagram'larına enkapsüle edilir.
19. KlasB network adresinin, binary olarak adres aralığı nedir?
    A. 01xxxxxx
    B. 0xxxxxxx
    C. 10xxxxxx
    D. 110xxxxx
20. Aşağıdaki hangi protokol hem TCP hem de UDP kullanır?
    A. FTP
    B. SMTP
    C. Telnet
    D. DNS

# Gözden Geçirme Sorularının Cevapları

1. B, D Bir binary sayıyı ondalık sayıya çevirmek için sadece, 1 olan bit değerlerini eklemelisiniz. 10011101, 128, 16, 8, 4 ve 1 'dir. 128+16++4+1= 157 'dir. Hexadecimal, 16'lı sayı sistemidir. Hexadecimal değerleri, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F olarak, ihtiyacınız olan tüm sayıları oluşturmanız için toplam 16 karakterdir. Şayet 1001 binary sayı, 9 ise, hexadecimal karşılığı da 9 'dur. Bundan sonraki binary değer 1101 sayısı 13 'tür, hexadecimal karşılığı, D'dir ve tüm hexadecimal cevabı 0x9D 'dir. Binary/hexadecimal numaraları, Modül1 'de işlenmiş olsa da, burada iyi bir tekrar yapmış olduk.
2. C Proxy ARP, routing veya varsayılan bir ağ geçidi yapılandırmadan, bir subnet'teki makinelerin, uzak subnet'lere ulaşmalarına yardım edebilir.
3. C Dynamic Host Configuration Protocol (DHCP), ağınızdaki kullanıcı makinelerine, IP bilgisi sağlamak için kullanılmaktadır. En yaygın olanları IP adresi, subnet mask, varsayılan ağ geçidi ve DNS bilgisidir.
4. B Address Resolution Protocol (ARP), bilinen bir IP adresinden donanım adresi bulmak için kullanılmaktadır.
5. A, C, D Bu ilk bakışta zor bir soru olarak görülmektedir, çünkü anlamı yoktur. Listelenen cevaplar OSI modelindendir ve soru TCP/IP protokol yığını (DoD) hakkındadır. Bununla beraber, gelin neyin yanlış olduğuna bakalım. İlk olarak, ne Session katmanı, TCP/IP modelindedir, ne de Data Link ve Physical katmanı. Bize, Transport katmanı (DoD modelinde host-to-host katmanı), Internet katmanı (OSI'de Network katmanı) ve Application katmanı (DoD'da Application/Process katmanı) kalmaktadır.
6. C Bir KlasC network adresi, host'ları tanımlamak için sadece 8 bit'e sahiptir: $2^8$-2 = 254.
7. A, B Bir IP adresi almak için DHCP Discover mesajı atan bir istemci, hem katman 2 hem de katman 3 broadcast gönderir. Katman 2 broadcast, hex'de tamamı F'lerden oluşur (FF:FF:FF:FF:FF:FF). Katman 3 broadcast, tüm network ve host'lar anlamında 255.255.255.255'dir. DHCP, connectionless'dır. Yani, host-to-host katmanı da denilen Transport katmanında User Datagram Protocol (UDP) kullanır.
8. B Telnet TCP ve IP (TCP/IP) kullandığı halde, soru özellikle katman 4 için bilgi istemektedir. IP katman 3'de çalışmaktadır. Telnet, katman4'de TCP kullanır.
9. C, D Internet Control Message Protocol (ICMP), ağ üzerinden hata mesajı göndermek için kullanılmaktadır. Fakat tek başına çalışmazlar. Her segment veya ICMP yükü, bir IP datagram'ında (veya pakette) enkapsüle edilmek zorundadır.
10. B, D, E SMTP, FTP ve HTTP, TCP kullanır.
11. A, C, F Açıklama: DHCP, SNMP ve TFTP, UDP kullanır. SMTP, FTP ve HTTP, TCP kullanır.
12. C, D, E Telnet, File Transfer Protocol (FTP) ve Trivial FTP (TFTP)'nin hepsi Uygulama katmanı protokolleridir. IP, Network katman protokolüdür. Transmission Control Protocol (TCP), bir Transport katman protokolüdür.
13. C İlk olarak, TCP ve UDP'nin, Transport katmanında çalıştığını kolayca bilmelisiniz, bu nedenle şimdi %50 şansınız vardır. Bununla birlikte, başlık, sequencing, acknowledgment ve window numaralarına sahip olduğundan, cevap, sadece TCP olabilir.
14. A Hem FTP hem de Telnet, Transport katmanında TCP kullanır. Bununla birlikte, her ikisi de Application katmanı protokolleridir. Bu nedenle Application katmanı bu soru için en iyi cevaptır.
15. C DoD modelindeki dört katman, Application /Process, Host-to-host, Internet ve Network Access katmanlarıdır. Internet katmanı, OSI modelinin Network katmanına denk gelir.

16. C,E KlasA özel adres aralığı, 10.0.0.0'dan 10.255.255.255'e kadardır. KlasB özel adres aralığı, 172.16.0.0'dan 172.16.255.255'e ve KlasC özel adres aralığı 192.18.0.0'dan 192.68.255.255'e kadardır.
17. B TCP/IP yığının (ayrıca DoD model olarak ta belirtilir) dört katmanı, Application /Process, Host-to-Host, Internet ve Network Access'dir. Host-to-host katmanı, OSI modelindeki Transport katmanının eşdeğeridir.
18. B, C ICMP, sistem denetimi ve hedef erişilemez mesajı için kullanılmaktadır. ICMP, IP datagram'larında enkapsüle edilir ve sistem denetimi için kullanıldığından, Host'a ağ problemleri ile ilgili bilgi sağlayacaktır.
19. C Klas B network adres aralığı 128-191 'dir. Bu binary aralığımızı 10xxxxxx yapar.
20. D DNS sunucular arasında zone transferi için TCP, bir istemci kullanıcı adını bir IP adresine çözümlemeye çalıştığında UDP kullanır.

# Yazılı Lab 2 Cevapları

1. 192–223, 110xxxxx
2. Host-to-Host
3. 1–126
4. Loopback veya diagnostic.
5. Host bit'lerinin hiçbirini kullanmayın.
6. Host bit'lerinin hepsini kullanın.
7. 10.0.0.0'dan 10.255.255.255'e kadar
8. 172.16.0.0'dan 172.31.255.255 'e kadar
9. 192.168.0.0'dan 192.168.255.255'e kadar
10. 0–9 ve A, B, C, D, E ve F

# 3 Subnet'leme, Variable Length Subnetmask'lar (VLSM'ler) ve TCP/IP Hata Giderme

# 3 Subnet'leme, Variable Length Subnetmask'lar (VLSM'ler) ve TCP/IP Hata Giderme

- Subnet'leme Temelleri
- Variable Lenght Subnet Mask'lar (VLSM'ler)
- Summarization
- IP Adreslemesinde Hata Giderme
- Özet
- Sınav Gereklilikleri
- Yazılı Lab'lar 3
- Gözden Geçirme Soruları
- Gözden Geçirme Sorularının Cevapları
- Yazılı Lab 3.1 Cevapları
- Yazılı Lab 3.2 Cevapları
- Yazılı Lab 3.3 Cevapları

# Subnet'leme, Variable Length Subnetmask'lar (VLSM'ler) ve TCP/IP Hata Giderme

Bu bölümde, önceki bölümde atladığımız konuları ele alacağız. IP adresleme tartışmalarına devam edeceğiz.

Bir IP ağını subnet'lere ayırarak işe başlayacağız. Gerçekten kendinizi vermek zorundasınız, çünkü subnet'leme zaman alacaktır ve pekiştirmek için uygulama gerekecektir. Bu nedenle sabırlı olun. Bu bölüm gerçekte çok önemlidir, muhtemelen bu kitaptaki anlaşılması en önemli bölümdür.

En başından IP adreslemesini işleyeceğim. Bu size biraz garip gelebilir, ama bu bölümü okumadan önce, subnet'leme hakkında öğrendiğiniz her şeyi unutmaya çalışırsanız sizin için daha iyi olacağını düşünüyorum. IP subnet'lemeyi tartıştıktan sonra, hem tamamen Variable Length Subnet Mask'lardan (VLSM) bahsedeceğim hem de VLSM ağları kullanarak, bir ağı nasıl tasarlayıp kuracağınızı göstereceğim.

VLSM tasarım ve kurulumunu iyice öğrenince, classfull sınırlarını nasıl summarize edeceğinizi göstereceğim. Bunu, EIGRP ve OSPF routing protokolleri kullanarak, summarize etmeyi göstereceğim yer olan "Enhanced IGRP (EIGRP) ve Open Shortest Path First (OSPF)" 'başlıklı bölüm 7'de biraz daha fazla inceleyeceğiz.

IP adresi hata tespitini inceleyerek bu bölümü tamamlayacağım ve IP ağında hata tespiti yapıldığında, sizi Cisco'nun önerdiği adımlara götüreceğim.

Zihnen hazırsınız ve bir gezintiye çıkmak üzeresiniz! Bu bölüm, gerçekten IP adreslemesi ve network kurulumunu anlamanıza yardımcı olacaktır. Bu yüzden sakın hevesiniz kırılmasın ve bırakmayın. Şayet subnet'lemeyi sebatla çalışmaya devam ederseniz, bir gün geriye bakıp, katlanmaya karar verdiğinize çok memnun kalacağınıza eminim.
Onu anladıktan sonra, daha önce neden çok zor olduğunu düşündüğünüze şaşıracaksınız. Hazır mısınız? Hadi başlayalım!

*Bu bölüm ile ilgili son güncellemeler için lütfen, www.lammle.com ve/veya www.sybex.com 'e bakınız.*

NOT

## Subnet'leme Temelleri

bölüm 2'de, host bit'lerinin hepsini önce 0 sonrada 1 yaparak KlasA, KlasB ve KlasC ağ adreslerinde kullanılan geçerli host aralığını nasıl tanımlayıp bulacağımızı öğrendik. Bu çok güzel, fakat anlaşılması gereken nokta şu: Sadece bir network tanımlıyordunuz. Şayet tek network adresi alıp, bundan altı network oluşturmak isterseniz ne olacak? Subnet'leme olarak belirtilen yöntemi uygulamak zorunda kalacaktınız. Çünkü geniş bir network alıp onları küçük network parçalarına bölmenize bu izin verecektir.

Subnet'leme kullanmanın, aşağıdaki faydaları içeren, tonlarca sebebi vardır:

**Düşük network trafiği:** Biz düşük trafiğin her türünden memnuniyet duyarız. Ağlar farklı değildir. Güvenilir router'lar olmaksızın, paket trafiği tüm ağı durma noktasına getirir. Router'larla çoğu trafik lokal ağda kalacaktır, sadece diğer network'lere hedeflenen paketler, router üzerinden geçirilecektir. Router'lar, broadcast domain'ler oluşturur. Ne kadar çok broadcast domain'i oluşturursanız, her bir network segment'inde daha az network trafiği ve daha küçük broadcast domain'leri oluşur.

**Optimize edilmiş network performansı:** Bu, düşük bir network trafiğinin neticesidir.

**Basitleştirilmiş yönetim:** Network problemlerini tespit edip ayırmak, daha küçük birbirine bağlı network gruplarında, devasa bir ağdakinden daha kolaydır.

**Kolaylaştırılmış, geniş coğrafi mesafe dağılımı:** WAN linklerinin, LAN linklerinden oldukça yavaş ve pahalı olmasından dolayı, geniş bir alana yayılmış büyük bir network, önceden listelenen her alanda problem oluşturacaktır. Çok sayıda küçük ağın bağlanması sistemi daha verimli yapacaktır.

Sonraki bölümde, bir network adresinin subnetlenmesine geçeceğim. Bu da işin iyi tarafı. Hazır mısınız?

## IP Subnet-Zero

`Ip subnet-zero` yeni bir komut değil, fakat geçmişteki kurs programı ve Cisco sınav konuları onu içermiyordu. Şimdi kesinlikle sınava dahi! Bu komut, network tasarımınızda ilk ve son subnet'i kullanmanızı sağlar. Örneğin, 192 KlasC mask'ı, 64 ve 128 ağlarını verir (bu bölümün ilerleyen kısımlarında tamamıyla incelenecektir) fakat `ip subnet-zero` komutu ile şimdi 0, 64, 128 ve 192 subnetlerini kullanabilirsiniz. Bu, kullandığımız her subnet mask için iki fazla subnet demektir.

Bir sonraki "Cisco Internetworking Operating System (IOS) ve Security Device Manager (SDM)" bölümüne kadar command line interface'i (CLI) tartışmasak da, bu komuta aşina olmak sizin için önemlidir:

```
P1R1#sh running-config
Building configuration...
Current configuration : 827 bytes
!
hostname Pod1R1
!
ip subnet-zero
!
```

NOT *Cisco sınavlarınız için çalıştığınızda, Cisco'nun, ip subnet-zero 'yu kullanmamayı isteyip istemediğini, çok iyi okuyup anladığınızdan emin olun.*

Bu router çıktısı, `ip subnet-zero` komutunun router'da etkin olduğunu gösterir. Cisco bu komutu, Cisco IOS versiyon 12x ile varsayılan olarak başlatarak aktif hale getirdi.

## Subnet'ler Nasıl Oluşturulur?

Alt ağlar oluşturmak için IP adresinin host bölümünden bit alırsınız ve subnet adreslerini tanımlamak için onları rezerve edersiniz. Bunun anlamı host için az sayıda bit demektir, böylece ne kadar çok subnet olursa, host tanımlamak için daha az bit olacaktır.

Bu bölümün devamında, KlasC adreslerden başlayarak subnet'lerin nasıl oluşturulduğunu öğreteceğim. Fakat gerçekte subnet'leme yapmadan önce, hem mevcut gereksinimlerinizi hem de gelecekteki durumunuza göre planlarınızı tanımlamak zorundasınız.

NOT *Bir subnet mask tasarlayıp oluşturmaya geçmeden önce, bu bölümde, ağdaki tüm host'ların (düğümlerin) tamamen aynı subnet mask'ını kullanmaları anlamına gelen classfull routing'i işleyeceğimizi anlamanız gerekir. Variable Length Subnet Mask'a (VLSM) geçtiğimizde network segment'indeki herkesin farklı subnet mask'ını kullanabileceği classless routing'i anlatacağım.*

Bir subnet oluşturmak için aşağıdaki adımları takip edin:

1. Gerekli network ID sayısını belirleyin:
   - Her subnet için bir tane
   - Her WAN bağlantısı için bir tane
2. Subnet başına gerekli host ID sayısını tespit edin:
   - Her TCP/IP host'u için bir tane
   - Her router interface'i için bir tane

3. Yukarıdaki gereksinimlere göre, aşağıdakileri oluşturun:
   - Tüm ağınız için bir subnet mask
   - Her fiziksel segment için eşsiz bir subnet ID'si
   - Her subnet için bir host ID aralığı

## 2'nin Üslerini Anlamak

*2 sayısının üslerinin, IP subnet'lemesi için anlaşılması ve akılda tutulması önemlidir. 2'nin üssüne göz atmak için, bir sayıyı onun üssü olan sayı ile gördüğünüzde, numaranın kendisini, belirtilen üs sayısı kadar çarpacaksınız demektir. Örneğin, $2^3$, 2x2x2 = 8 dir. Burada, ezberlemeniz gereken, 2'nin üsleri listesi vardır:*

*$2^1 = 2$*

*$2^2 = 4$*

*$2^3 = 8$*

*$2^4 = 16$*

*$2^5 = 32$*

*$2^6 = 64$*

*$2^7 = 128$*

*$2^8 = 256$*

*$2^9 = 512$*

*$2^{10} = 1,024$*

*$2^{11} = 2,048$*

*$2^{12} = 4,096$*

*$2^{13} = 8,192$*

*$2^{14} = 16,384$*

*Bu üslerin hepsini öğrenmekle ilgili strese girmeden önce, onları bilmenin, zorunlu değil ama faydalı olduğunu hatırlayın. Burada, 2'lerle çalışırken, ufak bir ipucu bulabilirsiniz: 2'nin her ardışık üssü, bir öncekinin çift katıdır.*

*Örneğin, $2^9$ değeri ile ilgili hatırlamanız gereken tek şey, ilk olarak $2^8$'in 256 olduğunu bilmektir. Neden? Çünkü sekiz üssünün (256) çift katını (512) alırsanız, $2^9$'u (ya da 512'yi) bulursunuz. $2^{10}$ değerini bulmak için basitçe $2^8$=256 ile başlayın ve sonra onun iki defa çiftini alın.*

*Bir diğer yolu da kullanabilirsiniz. Şayet $2^6$'nın ne olduğunu bilmeniz gerekiyorsa, sadece 256'yı iki kez yarıya bölün: önce $2^7$ ve sonra $2^6$'ya ulaşırsınız.*

## Subnet Mask'lar

Subnet adres sisteminin çalışması için ağdaki her makinenin, host adresinin hangi bölümünü, subnet adresi olarak kullanılacağını bilmesi gerekir. Bu, her makineye bir subnet mask'ı atayarak başarılır. Bir subnet mask, IP paketlerinin alıcısının, IP adresinin host ID bölümünden, IP adresinin network ID bölümünü ayırt etmesini sağlayan, 32-bit bir değerdir.

Network yöneticileri, 1 ve 0'lardan oluşan 32-bit bir subnet mask oluştururlar. Subnet mask'taki 1'ler, network ve subnet adreslerine işaret eden konumları belirtir.

Tüm ağlar, subnet'lere ihtiyaç duymaz, yani onlar varsayılan subnet mask'ı kullanır. Bu, esasen bir ağın bir subnet adresine sahip olmadığını söylemekle aynı anlama gelir. Tablo 3.1 KlasA, B ve C için varsayılan subnet mask'ları göstermektedir. Bu varsayılan mask'lar değişemez. Başka bir deyişle, bir KlasB subnet mask'ını 255.0.0.0 olarak değiştiremezsiniz. Şayet denerseniz host, bunu geçersiz adres olarak okuyacaktır ve genelde bu şekilde yazmanıza bile izin vermeyecektir. Bir KlasA ağı için subnet mask'ındaki ilk byte'ı değiştiremezsiniz; tamamı 1'lerden oluşan 255.255.255.255 olarak bir broadcast adresi gibi tanımlayamazsınız. Bir KlasB adresi, 255.255.0.0 ile ve KlasC adresi de 255.255.255.0 ile başlamak zorundadır.

**Tablo 3.1:** Varsayılan Subnet Mask'ı

| Klas | Format | Varsayılan Subnet Mask |
|---|---|---|
| A | ***network.düğüm.düğüm.düğüm*** | 255.0.0.0 |
| B | ***network.network.düğüm.düğüm*** | 255.255.0.0 |
| C | ***network.network.network.düğüm*** | 255.255.255.0 |

## Classless Inter-Domain Routing (CIDR)

İyi bilmeniz gereken diğer bir terim, Classless Inter-Domain Routing'dir (CIDR). Aslında ISP'lerin (internet servis sağlayıcıların), bir firma ya da ev kullanıcısı için bir adres ayırmak için kullandıkları yöntemdir. Onlar, tam blok boyutunda adresler sağlar. (Bu konuyla ilgili daha detaylı bilgiyi bölümün devamında vereceğim.)

ISP'den bir blok adres alındığında şu şekilde olacaktır: 192.168.10.32/28. Bu size, subnet mask'ınızın ne olduğunu söyler. Slaş gösterimi (/) kaç tane bit'in 1 olacağı anlamına gelir. Açıkça, bir byte'ın 8 bit ve bir IP adresinde 4 byte (4x8=32) olmasından dolayı maksimum /32 olabilir. Fakat unutmayın ki, host bit'leri için en az 2 bit ayırmak zorunda olduğunuzdan, uygun olan en büyük subnet mask'ı (adresin sınıfına bakılmaksızın) /30 olabilir.

Örneğin, 255.0.0.0 olan bir KlasA varsayılan subnet mask adresini alın. Yani, subnet mask'ın ilk byte'ı, tamamen 1 lerden oluşmaktadır. (11111111). Bir slaş gösterimine işaret edildiğinde, mask'ınızı anlamak için 1 olan tüm bit'leri saymanız gerekir. 255.0.0.0, 1'lerden oluşan 8 bit'i olduğu için, /8 olarak belirtilmektedir.

Bir KlasB varsayılan mask'ın 16 bit'i 1'lerden oluştuğundan /16 olan 255.255.0.0'dır:

**11111111.11111111.00000000.00000000**

Tablo 3.2, uygun tüm subnet mask'larını ve onların CIDR slash gösterim eşdeğerlerini listelemektedir.

**Tablo 3.2:** CIDR Değerleri

| Subnet Mask | CIDR Değeri |
|---|---|
| 255.0.0.0 | /8 |
| 255.128.0.0 | /9 |
| 255.192.0.0 | /10 |
| 255.224.0.0 | /11 |
| 255.240.0.0 | /12 |
| 255.248.0.0 | /13 |
| 255.252.0.0 | /14 |

**Tablo 3.2:** CIDR Değerleri (devam)

| Subnet Mask | CIDR Değeri |
|---|---|
| 255.254.0.0 | /15 |
| 255.255.0.0 | /16 |
| 255.255.128.0 | /17 |
| 255.255.192.0 | /18 |
| 255.255.224.0 | /19 |
| 255.255.240.0 | /20 |
| 255.255.248.0 | /21 |
| 255.255.252.0 | /22 |
| 255.255.254.0 | /23 |
| 255.255.255.0 | /24 |
| 255.255.255.128 | /25 |
| 255.255.255.192 | /26 |
| 255.255.255.224 | /27 |
| 255.255.255.240 | /28 |
| 255.255.255.248 | /29 |
| 255.255.255.252 | /30 |

/8'den /15'e kadar sadece KlasA network adresleri ile kullanılabilmektedir. /16'dan /23'e kadar KlasA ve KlasB adresleri ile kullanılabilmektedir. /24'den /30'a kadar,KlasA, B ve C network adresleri tarafından kullanılabilir. Birçok firmanın KlasA network adresi kullanmasının ana sebebi budur. Tüm subnet mask'ları kullanabildiklerinden, network tasarımında maksimum esnekliğe sahip olacaklardır.

*Bir Cisco router'ı, bu slaş formatını kullanarak yapılandıramazsınız. Yine de slaşlı gösterimde (CIDR) subnet mask'ları bilmeniz gerçekten önemlidir.*

NOT

## KlasC Adreslerinin Subnet'lenmesi

Bir ağı subnet'lerine ayırmanın birçok farklı yöntemi vardır. En doğru yol, sizin için en iyi çalışandır. Bir KlasC adreste, host'ları tanımlamak için sadece 8 uygun bit vardır. Subnet bit'lerinin, bit'leri atlamaksızın soldan başladığını ve sağa doğru gittiğini hatırlayın. KlasC subnet mask'ı sadece şunlar olabilir:

```
Binary Decimal CIDR
_____________________

00000000 = 0        /24
10000000 = 128      /25
11000000 = 192      /26
11100000 = 224      /27
11110000 = 240      /28
11111000 = 248      /29
11111100 = 252      /30
```

Host'lara IP adresi atamamız için en az iki host bit'ine sahip olmamız gerektiğinden, /31 veya /32 kullanamayız. Geçmişte, bir KlasC ağında /25'i asla tartışmazdım. Cisco, daima en az 2 subnet

mask'ına sahip olunmasıyla ilgilenirdi, fakat şimdi Cisco'nun ders müfredatı ve sınav konularında `ip subnet-zero` komutuna yer vermesinden dolayı, sadece 1 subnet bit'i kullanabiliriz.

Sonraki bölümlerde daha büyük sayıları subnet'lemeyi kolaylaştıran bir alternatif subnet'leme yöntemini öğreteceğim. Emin olun, hızlı bir şekilde subnet'lemeye ihtiyaç duyacaksınız.

## Bir KlasC Adresi Subnet'lemesi: Hızlı Yol!

Ağınız için uygun bir subnet mask seçtiğinizde ve mask'ın sağladığı subnet sayısını, geçerli host'ları ve bir subnet'in broadcast adreslerini belirlemeye ihtiyacınız olduğunda, tek yapmanız gereken beş basit soruya cevap vermektir:

- Seçili subnet mask kaç tane subnet oluşturur?
- Her subnet için kaç tane geçerli host vardır?
- Geçerli subnet'ler nelerdir?
- Her subnet'in broadcast adresi nedir?
- Her subnet'teki geçerli host'lar nelerdir?

Bu noktada 2'nin üslerini anlamanız ve hatırlamanız önemlidir. Şayet ihtiyacınız olursa, bu bölümde daha önceki "2'nin Üslerini Anlamak" başlığına bakınız. Bu beş büyük soruya nasıl cevap bulacağımızı aşağıda görebilirsiniz:

- Kaç tane subnet? $2^x$ =subnet sayısıdır. X, mask'lanmış bit ya da 1'lerin sayısıdır. Örnek olarak, 11000000'da 1'lerin sayısı bize, 22 subnet sayısını verir. Bu örnekte 4 subnet vardır.
- Her subnet için kaç host? $2^y – 2$ = her subnet için host sayısıdır. Y mask'lanmamış bit ya da 0'ların sayısıdır. Örneğin, 11000000'da, 0'ların sayısı bize $2^6 – 2$ host'ları verir. Bu örnekte her subnet için 62 host vardır. Subnet adres ve broadcast adres için 2'yi çıkarmanız gerekir, bunlar geçerli değildir.
- Geçerli subnet'ler nelerdir? 256-subnet mask= blok boyutu veya artırım sayısıdır. 256-192=64 bir örnek olabilir. 192 mask'ın blok boyutu daima 64'dür. 64 bloğundaki sıfırları, subnet mask'ınıza ulaşana kadar saymaya başlayın. Bunlar sizin subnet'lerinizdir: 0, 64, 128, 192. Kolay değil mi?
- Her subnet'in broadcast adresi nedir? Şimdi, gerçekten kolay bölüme geldik. Son bölümde, subnet'lerimizi, 0, 64, 128 ve 192 olarak saydığımızda, broadcast adresi daima bir sonraki subnet'ten bir önceki sayıdır. Örneğin, 0 subnet'inin broadcast adresi, bir sonraki subnet 64 olduğundan, 63'tür. 64 subnet'inin broadcast adresi, bir sonraki subnet 128 olduğundan 127'dir. Son subnet'in broadcast adresinin daima 255 olduğunu hatırlayın.
- Geçerli host'lar nelerdir? Geçerli host'lar, tüm 0 ve 1'ler göz ardı edilerek, subnet'ler arasındaki numaralardır. Örnek olarak, şayet 64 subnet ve 127 broadcast adresi ise 65-126 arası geçerli host aralığıdır. Daima subnet adresi ile broadcast adresleri arasındaki sayılardır.

Bunun tamamıyla akıl karıştırıcı olabileceğini biliyorum. Fakat gerçekten ilk göründüğü kadar zor değildir. Neden kendinizi biraz sınamıyorsunuz?

## Subnet'leme ile ilgili Uygulama Örnekleri: KlasC Adreslemesi

Burada, açıkladığım yöntemi kullanarak, KlasC subnet'leme pratiği yapma fırsatı bulabilirsiniz. Heyecanlandınız, değil mi? İlk olarak KlasC mask'larıyla başlayacağız ve bir KlasC adresi kullanarak, yapabileceğimiz her subnet ile çalışma yapacağız. Tamamladığımızda, bunun KlasA ve KlasB ağlarla ne kadar kolay olduğunu da göstereceğim!

### Uygulama Örneği#1C: 255.255.255.128 (/25)

128, binary olarak 10000000 olduğundan, subnet'leme için sadece 1 bit ve host'lar için 7 bit vardır. 192.168.10.0 KlasC bir network adresini subnetleyeceğiz.

192.168.10.0 = Network adresi

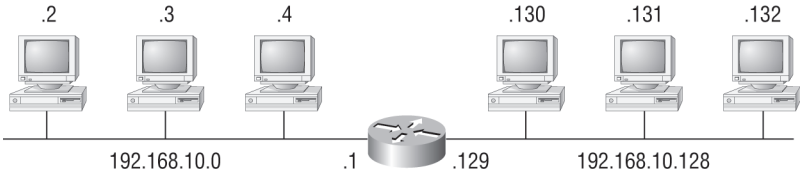
255.255.255.128 = Subnet mask

Şimdi, şu beş büyük sorumuzu cevaplayalım:

- Kaç tane subnet? 128 ,(10000000)'de 1 bit olduğundan, cevap, $2^1 = 2$'dir.
- Her subnet için kaç kullanıcı? 7 sıfırdan oluşan host bit'imiz var (10000000), bu nedenle $2^7 - 2 = 126$ host olacaktır.
- Geçerli subnet'ler nelerdir? 256 – 128 = 128. 0'dan başlayacağımızı ve blok boyutumuzu sayacağımızı hatırlayın. Bu nedenle subnet'lerimiz 0 ve 128'dir.
- Her subnet için broadcast adresi nedir? Bir sonraki subnet'ten hemen önceki numara, host bit'lerinin tamamının 1 olduğu numaradır ve broadcast adresine eşittir. 0 subnet'i için sonraki subnet 128'dir, bundan dolayı broadcast adresi 127'dir.
- Geçerli host'lar nelerdir? Bunlar, subnet ve broadcast adresleri arasındaki numaralardır. Host'ları bulmanın en kolay yolu, subnet ve broadcast adreslerini yazmaktır. Bu yolla geçerli host'lar açıkça görülmektedir. Aşağıdaki tablo, 0 ve 128 subnet'lerini, her birinin geçerli host aralığını ve her iki subnet'in broadcast adreslerini göstermektedir:

| **Subnet** | 0 | 128 |
|---|---|---|
| **İlk host** | 1 | 129 |
| **Son host** | 126 | 254 |
| **Broadcast** | 127 | 255 |

Diğer örneğe geçmeden önce, Şekil 3.1'e bir bakalım. KlasC /25'e bakarak, kolayca iki subnet'in olduğunu söyleyebilirsiniz. Peki bunda ne var, bu niçin önemlidir? Aslında önemli değil, fakat bu doğru bir soru değil. Gerçekten bilmek istediğiniz, bu bilgiyle ne yapacağınızdır.




```
Router#show ip route
[output cut]
C 192.168.10.0 is directly connected to Ethernet 0.
C 192.168.10.128 is directly connected to Ethernet 1.
```

***Şekil 3.1:*** *Bir KlasC /25 mantıksal network çalıştırmak.*

```
Router#show ip route
[output cut]
C192.168.10.0 direkt olarak Ethernet 0'a bağlı.
C 192.168.10.128 direkt olarak Ethernet 1'e bağlı.
```

Bunun herkes için çok eğlenceli geçmediğini biliyorum, fakat bu konu gerçekten önemlidir, bu nedenle orada kalın: Subnet'lemeden bahsedeceğiz. Subnet'lemeyi anlamanın anahtarının, onu yapmanızı gerektiren gerçek sebebi anlamak olduğunu bilmeye ihtiyacınız vardır. Ve ben, fiziksel bir network oluşturma işlemi (bir router ekleyerek) ile bunu göstereceğim. Router eklediğimizden, ağ topluluğumuzdaki host'larımızın iletişime geçmesi için mantıksal network adresleme planına sahiptir. IPX veya IPv6 kullanabiliriz, fakat IPv4 hala en popüler olanıdır ve şimdi çalışacağımızda zaten o olacak. Tamam, şimdi Şekil 3.1'e tekrar bakalım. İki fiziksel network vardır bu nedenle, iki mantıksal network için izin verilen bir mantıksal adresleme planı gerçekleştireceğiz. Her zamanki

gibi geleceğe bakmak ve (yakın ve uzun vade) muhtemel büyüme senaryolarını göz önünde tutmak gerçekten iyi fikirdir. Bu örnek için /25 olacaktır.

### Uygulama Örneği#2C: 255.255.255.192 (/26)

Bu ikinci örnekte, 255.255.255.192 subnet maskını kullanarak, 192.168.10.0 network adresini subnetleyeceğiz.

192.168.10.0 = Network adresi

255.255.255.192 = Subnet mask'ı

Şimdi, şu beş büyük sorumuzu cevaplayalım:

- Kaç tane subnet? 192'de (11000000) 2 bit olduğundan cevap $2^2$ = 4'dür.
- Her subnet için kaç kullanıcı? 6 sıfırdan oluşan host bit'imiz var (11000000) bu nedenle, $2^6$ – 2 = 62 host olacaktır.
- Geçerli subnet'ler nelerdir? 256 – 192 = 64. 0'dan başlayacağımızı ve blok boyutumuzu sayacağımızı hatırlayın. Bu nedenle subnet'lerimiz 0, 64, 128 ve 192'dir.
- Her subnet için broadcast adresi nedir? Bir sonraki subnet'ten hemen önceki sayı, host bit'lerinin tamamının 1 olduğu sayıdır ve broadcast adresine eşittir. 0 subnet'i için sonraki subnet 64'dür, bundan dolayı broadcast adresi 63'dür.
- Geçerli host'lar nelerdir? Bunlar, subnet ve broadcast adresleri arasındaki numaralardır. Host'ları bulmanın en kolay yolu, subnet ve broadcast adreslerini yazmaktır. Bu yolla, geçerli host'lar açıkça görülmektedir. Aşağıdaki tablo 0, 64, 128 ve 192 subnet'lerini, her birinin geçerli host aralığını ve her iki subnet'in broadcast adreslerini göstermektedir:

| **Subnet'ler (ilk olarak bunu yapın)** | 0 | 64 | 128 | 192 |
|---|---|---|---|---|
| **İlk host'umuz (host adreslemesini en son yapın)** | 1 | 65 | 129 | 193 |
| **Son host'umuz** | 62 | 126 | 190 | 254 |
| **Broadcast adresi (bunu ikinci olarak yapın)** | 63 | 127 | 191 | 255 |

Bir sonraki örneğe geçmeden önce, bir /26'yı subnet'leyebileceğimizi görebilirsiniz. Bu çok değerli bilgiyle ne yapacaksınız? Onu uygulayın! Bir /26 network uygulaması için Şekil 3.2 'yi kullanacağız.

/26 mask, dört alt network sağlar ve her router interface'i için bir subnet'e ihtiyacımız var. Bu örnekte, bu mask ile aslında başka router interface'i eklemeye yetecek kapasitemiz vardır.

## Uygulama Örneği#3C: 255.255.255.224 (/27)

Bu ikinci örnekte, 255.255.255.224 subnet mask'ını kullanarak, 192.168.10.0 network adresini subnetleyeceğiz.

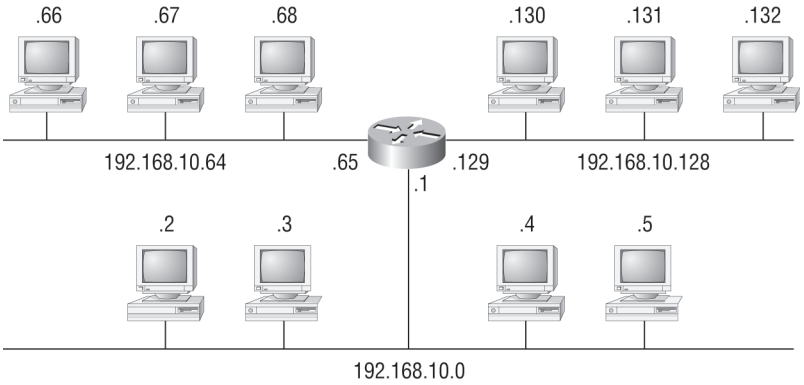
192.168.10.0 = Network adresi

255.255.255.224 = Subnet mask'ı

Şimdi, şu beş büyük sorumuzu cevaplayalım:

- Kaç tane subnet? 224'de (11100000) 3 bit olduğundan cevap, $2^3$ =8'dir.
- Her subnet için kaç kullanıcı? 5 sıfırdan oluşan host bit'imiz var (11100000), bu nedenle $2^5$ – 2 = 30 host olacaktır.
- Geçerli subnet'ler nelerdir? 256 – 224 = 32. Sıfırdan başladık ve 32 'nin bloklarındaki (artırımlarında) subnet mask değerini saydık. 0, 32, 64, 96, 128, 160, 192 ve 224'dür.
- Her subnet için broadcast adresi nedir? (Daima, sonraki subnet'ten bir önceki sayı mı?)

- Geçerli host'lar nelerdir (subnet ve broadcast numaraları arasındaki sayılar)?




***Şekil 3.2:*** *Bir KlasC /26 mantıksal ağını kurmak.*

```
Router#show ip route
[output cut]
C192.168.10.0 direkt olarak Ethernet 0'a bağlı.
C 192.168.10.64 direkt olarak Ethernet 1'e bağlı.
C 192.168.10.128 direkt olarak Ethernet 2'ye bağlı
```

Son iki soruya cevap vermek için, subnet'leri yazın, sonra, bir sonraki subnet'ten hemen önceki sayı olan broadcast adreslerini yazın. Son olarak host adreslerini doldurun. Aşağıdaki tablo size, 255.255.255.224 KlasC subnet mask'ı için tüm subnet'leri verecektir:

| **Subnet adresi** | 0 | 32 | 64 | 96 | 128 | 160 | 192 | 224 |
|---|---|---|---|---|---|---|---|---|
| **İlk geçerli host** | 1 | 33 | 65 | 97 | 129 | 161 | 193 | 225 |
| **Son geçerli host** | 30 | 62 | 94 | 126 | 158 | 190 | 222 | 254 |
| **Broadcast adresi** | 31 | 63 | 95 | 127 | 159 | 191 | 223 | 255 |

## Uygulama Örneği#4C: 255.255.255.240 (/28)

Gelin diğeri üzerinde çalışalım:

192.168.10.0 = Network adresi

255.255.255.240 = Subnet mask'ı

Şimdi, şu beş büyük sorumuzu cevaplayalım:

- Subnet'ler? 240 binary'de 11110000'dır, $2^4$ =16'dır.
- Host'lar? 4 host bit'i ya da, $2^4 - 2 = 14$ host olacaktır.
- Geçerli subnet'ler? 256 – 240 = 16. Sıfırdan başlayın: 0 + 16 = 16. 16 + 16 = 32. 32 + 16 = 48. 48 + 16 = 64. 64 + 16 = 80. 80 + 16 = 96. 96 + 16 = 112. 112 + 16 = 128. 128 + 16 = 144. 144 + 16 = 160. 160 + 16 = 176. 176 + 16 = 192. 192 + 16 = 208. 208 + 16 = 224. 224 + 16 = 240.
- Her subnet için broadcast adres?
- Geçerli host'lar?

Son iki soruya cevap vermek için aşağıdaki tabloyu kontrol edin. Size subnet'leri, geçerli host'ları ve her subnet için broadcast adresini verir. İlk olarak blok boyutunu (artım değerini) kullanarak

her subnet'in adresini bulun. İkinci olarak, her subnet artımının broadcast adresini bulun (daima sonraki geçerli subnet'ten bir önceki sayıdır.) Aşağıdaki tablo geçerli subnet'leri, host'ları ve bir KlasC 255.255.255.240 mask'ından sağlanan broadcast adreslerini göstermektedir:

| **Subnet** | 0 | 16 | 32 | 48 | 64 | 80 | 96 | 112 | 128 | 144 | 160 | 176 | 192 | 208 | 224 | 240 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **İlk host** | 1 | 17 | 33 | 49 | 65 | 81 | 97 | 113 | 129 | 145 | 161 | 177 | 193 | 209 | 225 | 241 |
| **Son host** | 14 | 30 | 46 | 62 | 78 | 94 | 110 | 126 | 142 | 158 | 174 | 190 | 206 | 222 | 238 | 254 |
| **Broadcast** | 15 | 31 | 47 | 63 | 79 | 95 | 111 | 127 | 143 | 159 | 175 | 191 | 207 | 223 | 239 | 255 |

NOT *Cisco, birçok kişinin 16'ları sayamayacağını ve KlasC 255.255.255.240 mask ile geçerli subnet, host ve broadcast adreslerini bulmanın sıkıcı geçeceğini düşünmüştür. Bu mask'ı çalışmak akıllıca olacaktır.*

## Uygulama Örneği#5C: 255.255.255.248 (/29)

Gelin uygulama yapmaya devam edelim:

192.168.10.0 = Network adresi

255.255.255.248 = Subnet mask'ı

- Subnet'ler? 248 binary'de 11111000'dır, $2^5$ =32'dır.
- Host'lar? $2^3 - 2 = 6$.
- Geçerli subnet'ler? 256 – 240 = 0, 8, 16, 24, 32, 40, 48, 56, 64, 72, 80, 88, 96, 104, 112, 120, 128, 136, 144, 152, 160, 168, 176, 184, 192, 200, 208, 216, 224, 232, 240 ve 248.
- Her subnet için broadcast adresi?
- Geçerli host'lar?

Aşağıdaki tabloya bir bakalım. Subnet'lerin bazılarını (sadece ilk ve son dört), geçerli host'lar ve KlasC 255.255.255.248 mask'ının broadcast'ini göstermektedir.

| **Subnet** | 0 | 8 | 16 | 24 | ... | 224 | 232 | 240 | 248 |
|---|---|---|---|---|---|---|---|---|---|
| **İlk host** | 1 | 9 | 17 | 25 | ... | 225 | 233 | 241 | 249 |
| **Son host** | 6 | 14 | 22 | 30 | ... | 230 | 238 | 246 | 254 |
| **Broadcast** | 7 | 15 | 23 | 31 | ... | 231 | 239 | 247 | 255 |

## Uygulama Örneği#6C: 255.255.255.248 (/30)

Sadece bir tane daha:

192.168.10.0 = Network adresi

255.255.255.252 = Subnet mask'ı

- Subnet'ler? 64.
- Host'lar? 2.
- Geçerli subnet'ler? 0, 4, 8, 12 vs…, 252'ye kadar hepsi.
- Her subnet için broadcast adres (daima sonraki subnet'ten tam önceki sayı mı)?
- Geçerli host'lar (subnet ve broadcast numaraları arasındaki numaralar)?

Aşağıdaki tablo size subnet, geçerli host ve 255.255.255.252 KlasC subnet'indeki ilk ve son dört subnet'in broadcast adreslerini göstermektedir:

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **Subnet** | 0 | 4 | 8 | 12 | ... | 240 | 244 | 248 | 252 |
| **İlk host** | 1 | 5 | 9 | 13 | ... | 241 | 245 | 249 | 253 |
| **Son host** | 2 | 6 | 10 | 14 | ... | 242 | 246 | 250 | 254 |
| **Broadcast** | 3 | 7 | 11 | 15 | ... | 243 | 247 | 251 | 255 |

## Sadece İki Host Sağlayan Bu Mask'ı Kullanmak Zorunda mıyız?

*San Francisco'daki Acme Corporation'da network yöneticisisiniz ve firma ofisinize düzinelerce WAN linki bağlanmaktadır. Ağınız, classful bir ağdır,. Farklı boyutlarda mask'lara sahip olabileceğiniz classless routing ile ilgili bir şeyler okudunuz. 255.255.255.252 (/30), bu durum için kullanışlı bir mask mıdır?*

*Evet, bu, WAN'larda oldukça kullanışlı bir mask'tır.*

*Şayet, 255.255.255.0 mask'ını kullanıyorsanız, her network, 254 host'a sahip olacaktır, fakat bir WAN linki için sadece iki adres kullanacaksınız! Bu, her subnet için 252 host'un çöpe atılması demektir. Şayet 255.255.255.252 mask'ını kullanırsanız, her subnet sadece iki hosta sahip olacaktır ve değerli adreslerinizi çöpe atmazsınız. Bu gerçekten önemli bir konudur, bu modülün ilerisindeki VLSM network tasarımında, daha detaylı göreceğiz.*

### Zihinden Subnet'leme: KlasC adreslemesi

Zihinden subnet hesabı yapmak gerçekten mümkündür. Bana inanmasanız da, nasıl olacağını göstereceğim. Ve zor da sayılmaz. Aşağıdaki örneğe bakalım:

192.168.10.33 = Düğüm adresi

255.255.255.224 = Subnet maskı

İlk olarak, yukarıdaki IP adresinin subnet ve broadcast adresini tespit edelim. Bunu, beş büyük sorunun üçüne cevap vererek yapabilirsiniz: 256 – 224 = 32. 0, 32, 64. 33 adresi, 32 ve 64 subnet'lerinin arasında kalır ve 192.168.10.32'nin parçası olmak zorundadır. Sonraki subnet 64'tür, bu nedenle 32 subnet'inin broadcast adresi, 63'tür (bir subnet'in broadcast adresinin, daima bir sonraki subnet'ten hemen önceki adres olduğunu hatırlayın). Geçerli host aralığı 33-62'dir (subnet ve broadcast adresleri arasındaki numaralardır). Bu çok kolay!

Tamam, gelin başka bir tane deneyelim. Başka bir KlasC adresini subnetleyelim:

192.168.10.33 = Düğüm adresi

255.255.255.240 = Subnet mask

Yukarıdaki IP adresi için broadcast ve subnet adresleri nelerdir? 256-240= 16. 0, 16, 32, 48. Bingo, host adresi 32 ve 48 subnet'leri arasındadır. Subnet, 192.168.10.32'dir ve broadcast adresi, 47'dir (sonraki subnet, 48'dir). Geçerli host aralığı, 33-46'dır (subnet numarası ve broadcast adresleri arasındaki numaralar).

Sadece eğlenmek için bir tane daha. Bu, tüm KlasC subnet'lemenin en kolay olanıdır:

192.168.10.17 = Düğüm adresi

255.255.255.252 = Subnet mask

Yukarıdaki IP adresi için broadcast ve subnet adresleri nelerdir? 256-252=0 (aksi söylenmedikçe daima 0 ile başlar) 4, 8, 12, 16, 20 v.s. Host adresi, 16 ve 20 subnet'leri arasındadır. Subnet, 192.168.10.16'dır ve broadcast adresi, 19'dur. Geçerli host aralığı 17-18'dir.

Şimdi, KlasC subnet'lemeyi tamamen bitirdiniz, gelin KlasB adreslemeye geçelim. Fakat bunu yapmadan önce, çabuk bir gözden geçirme yapalım.

## Neler Biliyoruz ?

Tamam, burası, gerçekten şimdiye kadar ne öğrendiyseniz uygulayabileceğiniz yerdir ve tamamını hafızanıza işlemeye başlayabilirsiniz. Burası, senelerdir sınıflarımda kullandığım en güzel bölümdür. Gerçekten, subnet'lemeyi kavramanızda size yardımcı olacaktır!

Bir subnet mask'ı ya da slaş gösterimi (CIDR) gördüğünüzde aşağıdakileri bilmelisiniz:

**/25**

Bir /25 hakkında ne biliyoruz?

- 128 mask
- 1 bit kullanılıyor ve 7 bit kullanılmıyor (10000000)
- Blok boyutu, 128
- Her biri 126 host'a sahip 2 subnet

**/26**

Bir /26 hakkında ne biliyoruz?

- 192 mask
- 2 bit kullanılıyor ve 6 bit kullanılmıyor (11000000)
- Blok boyutu, 64
- Her biri 62 host'a sahip 4 subnet

**/27**

Bir /27 hakkında ne biliyoruz?

- 224 mask
- 3 bit kullanılıyor ve 5 bit kullanılmıyor (11100000)
- Blok boyutu, 32
- Her biri 30 host'a sahip 8 subnet

**/28**

Bir /28 hakkında ne biliyoruz?

- 240 mask
- 4 bit kullanılıyor ve 4 bit kullanılmıyor (11110000)
- Blok boyutu, 16
- Her biri 14 host'a sahip 16 subnet

**/29**

Bir /29 hakkında ne biliyoruz?

- 248 mask
- 5 bit kullanılıyor ve 3 bit kullanılmıyor (11111000)
- Blok boyutu, 8
- Her biri 6 host'a sahip 32 subnet

**/30**

Bir /30 hakkında ne biliyoruz?

- 252 mask
- 6 bit kullanılıyor ve 2 bit kullanılmıyor (11111100)
- Blok boyutu, 4
- Her biri 2 host'a sahip 64 subnet

KlasA, KlasB veya KlasC adrese sahip olduğunuza bakılmaksızın /30 mask'ı, size sadece iki host sağlayacaktır. Bu mask, sadece noktadan-noktaya linklerin kullanılması için uygundur. (ayrıca Cisco tarafından da tavsiye edilir).

Şayet bu "Ne biliyoruz?" bölümünü akılda tutarsanız, işinizde ve günlük çalışmalarınızda her gün daha iyi durumda olursunuz. Onu yüksek sesle söylemeyi deneyin (bir şeyleri akılda tutmak için yardımcı olacaktır). Eşiniz ve/veya iş arkadaşlarınız, aklınızı kaybettiğinizi düşünecektir, şayet network alanında çalışıyorsanız, onlar muhtemelen zaten öyle olduğunuzu düşünüyorlardır. Ve henüz network alanında değilseniz, fakat girmek için çalışıyorsanız, tuhaf biri olduğunuzu düşünmeye başlayabilirler, ya da zaman içinde başlayacaklardır.

Bunları bazı çeşit flash kartlara yazmanız ve insanların sizin becerinizi test etmesi faydalı olur. Hem blok boyutlarını hem de bu "Ne biliyoruz?" bölümünü ezberlediğinizde, subnet'leme konusunu nasıl hızlı hallettiğinize şaşırırsınız.

## KlasB Adreslerini Subnet'lemek

Bu konuya dalmadan önce, gelin ilk olarak olası tüm KlasB subnet mask'larına bir bakalım. KlasC network adresleri ile yaptığımızdan çok daha fazla olası subnet mask'a sahip olduğumuza dikkat edin:

```
255.255.0.0     (/16)
255.255.128.0   (/17)       255.255.255.0     (/24)
255.255.192.0   (/18)       255.255.255.128   (/25)
255.255.224.0   (/19)       255.255.255.192   (/26)
255.255.240.0   (/20)       255.255.255.224   (/27)
255.255.248.0   (/21)       255.255.255.240   (/28)
255.255.252.0   (/22)       255.255.255.248   (/29)
255.255.254.0   (/23)       255.255.255.252   (/30)
```

KlasB network adresinin, host adreslemesi için uygun 16 bit'e sahip olduğunu biliyoruz. Yani, subnet'leme için 14 bit'e kadar kullanabiliriz.(çünkü en az 2 bit'i host adreslemesi için bırakmak zorundayız). /16 kullanılması, KlasB ile subnet'leme yapmadığınız anlamına gelir, fakat o, kullanabileceğiniz bir mask'tır.

*Sırası gelmişken, bu subnet değerleri (belki de model) ile ilgili bir şeyler dikkatinizi çekti mi? Bölümün başında binary'den ondalık sayıya dönüşümleri hatırlatmamın sebebi budur. Subnet mask'larının soldan başlayıp sağa doğru gitmesi ve bit'lerin sıralı olmasından dolayı, sayılar, adres sınıfı düşünülmeksizin daima aynıdır. Bu modeli unutmayın.*

NOT

Bir KlasB ağını subnet'leme işleyişi, daha fazla host bit'ine sahip olmanız ve üçüncü oktet'ten başlamanız dışında KlasC ile yapılanla nerdeyse aynıdır.

KlasC ile dördüncü oktet için kullandığınız, subnet numaralarının aynısını KlasB ile üçüncü oktet için kullanın, fakat dördüncü oktet'teki network bölümüne sıfır ve broadcast bölümüne 255 ekleyin. Aşağıdaki tablo size, bir KlasB 240 (/20) subnet mask'ında kullanılan iki subnet'in host aralığıyla ilgili bir örnek verir:

NOT *Yukarıdaki örnek sadece, /24 'e ulaşana kadar doğrudur. Bundan sonra, sayısal olarak KlasC ile aynıdır.*

| | | |
|---|---|---|
| İlk subnet | 16.0 | 32.0 |
| İkinci subnet | 31.255 | 47.255 |

Sadece sayılar arasındaki geçerli host'ları ekleyin ve bitti!

## Subnet'leme Uygulama Örnekleri: KlasB Adresleri

Bu bölüm size, KlasB adresleri ile subnet'leme uygulaması için bir fırsat verecektir. Üçüncü oktet'le başlamak dışında, bunun, KlasC ile subnet'leme ile aynı olduğunu tekrar belirtmek zorundayım:

## Uygulama Örneği#1B: 255.255.128.0 (/17)

172.16.0.0 = Network adresi

255.255.128.0 = Subnet mask'ı

- Subnet'ler? $2^1 = 2$ (C ile aynı).
- Host'lar? $2^{15} – 2 = 32{,}766$ (üçüncü oktet'te 7 , ve dörtte 8 bit).
- Geçerli subnet'ler? 256 – 128 = 128. 0, 128. Subnet'lemenin üçüncü oktet'te yapıldığını hatırlayın, bundan dolayı, subnet numaraları, sonraki tabloda görüldüğü gibi, gerçekte, 0.0 ve 128.0'dır. Bunlar, KlasC ile kullandıklarımızla tamamıyla aynıdır. Onları, üçüncü oktet'te kullanırız ve network adresleri için dördüncü oktet'te bir 0 ekleriz.
- Her subnet için broadcast adresi?
- Geçerli host'lar?

Aşağıdaki tablo, uygun iki subnet'i, geçerli host aralığını ve her biri için broadcast adreslerini göstermektedir:

| **Subnet** | 0.0 | 128.0 |
|---|---|---|
| **İlk host** | 0.1 | 128.1 |
| **Son host** | 127.254 | 255.254 |
| **Broadcast** | 127.255 | 255.255 |

Sadece dördüncü oktet'in en düşük ve en yüksek değerlerini ekleyip cevabı bulduğumuza dikkat edin. Bir KlasC subnet için yapılanla tamamen aynı yöntemle hesaplanmıştır. Üçüncü oktette aynı numaraları kullandık ve dördüncü oktet'te 0 ve 255 ekledik. Çok basit değil mi?

"Zor değildir; sayılar asla değişmez; biz onları sadece farklı oktet'lerde kullanırız" diyemem.

## Uygulama Örneği#2B: 255.255.192.0 (/18)

172.16.0.0 = Network adresi

255.255.192.0 = Subnet mask'ı

- Subnet'ler? $2^2 = 4$.
- Host'lar? $2^{14} – 2 = 16{,}382$ (üçüncü oktet'te 6 ve dörtte 8 bit).
- Geçerli subnet'ler? 256 – 192 = 64. 0, 64, 128, 192. Subnet'lemenin üçüncü oktet'te yapıldığını hatırlayın, bundan dolayı, subnet numaraları, sonraki tabloda görüldüğü gibi, gerçekte, 0.0, 64.0, 128.0 ve 192.0 'dır.
- Her subnet için broadcast adresi?
- Geçerli host'lar?

Aşağıdaki tablo, uygun iki subnet'i, geçerli host aralığını ve her biri için broadcast adreslerini göstermektedir:

| **Subnet** | 0.0 | 64.0 | 128.0 | 192.0 |
|---|---|---|---|---|
| **İlk host** | 0.1 | 64.1 | 128.1 | 192.1 |
| **Son host** | 63.254 | 127.254 | 191.254 | 255.254 |
| **Broadcast** | 63.255 | 127.255 | 191.255 | 255.255 |

Bir KlasC subnet için yapılanla nerdeyse aynıdır. Dördüncü oktet'te, üçüncü oktet'teki her subnet için 0 ve 255 ekledik.

## Uygulama Örneği#3B: 255.255.1240.0 (/20)

172.16.0.0 = Network adresi

255.255.240.0 = Subnet mask'ı

- Subnetler? $2^4 = 16$.
- Host'lar? $2^{12} – 2 = 4094$.
- Geçerli subnet'ler? 256 – 192 = 0, 16, 32, 48, vs. 240'a kadar. Bunların, bir KlasC 240 mask'ı ile aynı sayılar olduğuna dikkat edin. Onları sadece üçüncü oktet'e koyduk ve dördüncü oktet'te 0 ve 255 ekledik.
- Her subnet için broadcast adresi?
- Geçerli host'lar?

Aşağıdaki tablo, ilk dört ve son dört subnet'i, geçerli host aralığını ve bir KlasB 255.255.240.0 mask'ındaki broadcast adreslerini göstermektedir:

| Subnet | 0.0 | 16.0 | 32.0 | 48.0 |
|---|---|---|---|---|
| İlk host | 0.1 | 16.1 | 32.1 | 48.1 |
| Son host | 15.254 | 31.254 | 47.254 | 63.254 |
| Broadcast | 15.255 | 31.255 | 47.255 | 63.255 |

## Uygulama Örneği#4B: 255.255.1254.0 (/23)

172.16.0.0 = Network adresi

255.255.254.0 = Subnet mask'ı

- Subnet'ler? $2^7 = 128$.
- Host'lar? $2^9 – 2 = 510$.
- Geçerli subnet'ler? 256 – 254 = 0, 2, 4, 6, 8 vs. 254'e kadar.
- Her subnet için broadcast adresi?
- Geçerli host'lar?

Aşağıdaki tablo, ilk beş subnet'i, geçerli host'ları ve bir KlasB 255.255.254.0 mask'ındaki broadcast adreslerini göstermektedir:

| Subnet | 0.0 | 2.0 | 4.0 | 6.0 | 8.0 |
|---|---|---|---|---|---|
| İlk host | 0.1 | 2.1 | 4.1 | 6.1 | 8.1 |
| Son host | 1.254 | 3.254 | 5.254 | 7.254 | 9.254 |
| Broadcast | 1.255 | 3.255 | 5.255 | 7.255 | 9.255 |

## Uygulama Örneği#5B: 255.255.1255.0 (/24)

Yaygın inancın tersine, bir KlasB network adresi ile kullanılan 255.255.255.0, KlasC subnet mask'ı ile KlasB ağı olarak belirtilmez. Birçok insanın KlasB network'te kullanılan bu mask'ı, görünce bir

KlasC subnet mask'ı olarak düşünmesi şaşırtıcıdır. Bu, sebnetleme için 8 bit'li bir KlasB subnet mask'ıdır. Bir KlasC mask'ı ile oldukça farklıdır. Bu adresi subnet'lemek oldukça basittir:

172.16.0.0 = Network adresi

255.255.255.0 = Subnet mask'ı

- Subnet'ler? $2^8 = 256$.
- Host'lar? $2^8 - 2 = 254$.
- Geçerli subnet'ler? 256 – 255 = 1. 0, 1, 2, 3, vs. 255'e kadar.
- Her subnet için broadcast adresi?
- Geçerli host'lar?

Aşağıdaki tablo, ilk dört ve son iki subnet'i, geçerli host'ları ve bir KlasB 255.255.255.0 mask'ındaki broadcast adreslerini göstermektedir:

| **Subnet** | 0.0 | 1.0 | 2.0 | 3.0 | ... | 254.0 | 255.0 |
|---|---|---|---|---|---|---|---|
| **İlk host** | 0.1 | 1.1 | 2.1 | 3.1 | ... | 254.1 | 255.1 |
| **Son host** | 0.254 | 1.254 | 2.254 | 3.254 | ... | 254.254 | 255.254 |
| **Broadcast** | 0.255 | 1.255 | 2.255 | 3.255 | ... | 254.255 | 255.255 |

## Uygulama Örneği#6B: 255.255.1255.128 (/25)

Bu, çalışabileceğiniz en zor subnet mask'larından biridir. Ve daha kötüsü, her subnet için 126 host ile 500'ün üzerinde subnet oluşturduğundan, firmalarda kullanmak için gerçekten iyi bir subnet'tir. Bu yüzden sakın atlamayın!

172.16.0.0 = Network adresi

255.255.255.128 = Subnet mask'ı

- Geçerli Subnet'ler? 29= 512.
- Host'lar? 27 – 2 = 126.
- Geçerli subnet'ler? Burası ustalık isteyen bölüm. Üçüncü oktete kadar 256–255=1.0, 1, 2, 3, vs. Fakat dördüncü oktette kullanılan bir subnet bit'ini unutamazsınız. Bir KlasC mask'ı ile bir subnet mask'ının nasıl hesaplandığını gösterdiğimi hatırlayın. Bunu aynı yoldan hesaplarsınız. (Şimdi, KlasC bölümünde, 1-bit subnet mask'ını, bu kısmı size kolaylaştırmak için gösterdiğimi anlamışsınızdır.) Her üçüncü oktet değeri için iki subnet'e sahipsiniz, bu toplamda 512 subnet eder. Örneğin, üçüncü oktet, subnet 3 'ü gösteriyorsa, iki subnet aslında 3.0 ve 3.128 olacaktır.
- Her subnet için broadcast adresi?
- Geçerli host'lar?

Aşağıdaki tablo, subnet'leri nasıl oluşturacağınızı, geçerli host'ları ve bir KlasB 255.255.255.128 mask'ındaki broadcast adreslerini göstermektedir (ilk sekiz ve son iki subnet görülmektedir):

| **Subnet** | 0.0 | 0.128 | 1.0 | 1.128 | 2.0 | 2.128 | 3.0 | 3.128 | ... | 255.0 | 255.128 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **İlk host** | 0.1 | 0.129 | 1.1 | 1.129 | 2.1 | 2.129 | 3.1 | 3.129 | ... | 255.1 | 255.129 |
| **Son host** | 0.126 | 0.254 | 1.126 | 1.254 | 2.126 | 2.254 | 3.126 | 3.254 | ... | 255.126 | 255.254 |
| **Broad-cast** | 0.127 | 0.255 | 1.127 | 1.255 | 2.127 | 2.255 | 3.127 | 3.255 | ... | 255.127 | 255.255 |

## Uygulama Örneği#7B: 255.255.1255.192 (/26)

Bu, KlasB subnet'lemenin kolay olduğu yerdir. Üçüncü oktet, mask bölümünde, 255'e sahip olduğundan, üçüncü oktet'te listelenen herhangi bir numara, bir subnet numarasıdır. Bununla birlikte, dördüncü oktet'te bir subnet sayısına sahip olduğundan, bu oktet'i, KlasC subnet'lemede yaptığımız gibi subnet'leyebiliriz. Gelin bunu deneyelim:

172.16.0.0 = Network adresi

255.255.255.192 = Subnet mask'ı

- Subnetler? $2^{10} = 1024$.
- Host'lar? $2^6 – 2 = 62$.
- Geçerli subnet'ler?
- Her subnet için broadcast adresi? 256-192=64. Subnet'ler aşağıdaki tabloda gösterilmektedir. Bu numaralar tanıdık görünüyor mu?
- Geçerli host'lar?

Aşağıdaki tablo, ilk sekiz subnet aralığını, geçerli host'ları ve broadcast adreslerini göstermektedir:

| **Subnet** | 0.0 | 0.64 | 0.128 | 0.192 | 1.0 | 1.64 | 1.128 | 1.192 |
|---|---|---|---|---|---|---|---|---|
| **İlk host** | 0.1 | 0.65 | 0.129 | 0.193 | 1.1 | 1.65 | 1.129 | 1.193 |
| **Son host** | 0.62 | 0.126 | 0.190 | 0.254 | 1.62 | 1.126 | 1.190 | 1.254 |
| **Broadcast** | 0.63 | 0.127 | 0.191 | 0.255 | 1.63 | 1.127 | 1.191 | 1.255 |

Üçüncü oktet'teki her subnet değeri için dördüncü oktette, 0, 64, 128 ve 192 subnetlerine sahip olduğunuza dikkat edin.

## Uygulama Örneği#8B: 255.255.1255.224 (/27)

Bu, her subnet için uygun daha fazla subnet ve daha az host'a sahip olmamız dışında, yukarıdaki subnet mask'ıyla aynı yöntemle yapılır.

172.16.0.0 = Network adresi

255.255.255.224 = Subnet mask'ı

- Subnetler? $2^{11} = 2048$.
- Host'lar? $2^5 – 2 = 30$.
- Geçerli subnet'ler? 256-224 = 32. 0, 32, 64, 96, 128, 160, 192, 224.
- Her subnet için broadcast adresi?
- Geçerli host'lar?

Aşağıdaki tablo, ilk sekiz subnet'i göstermektedir:

| **Subnet** | 0.0 | 0.32 | 0.64 | 0.96 | 0.128 | 0.160 | 0.192 | 0.224 |
|---|---|---|---|---|---|---|---|---|
| **İlk host** | 0.1 | 0.33 | 0.65 | 0.97 | 0.129 | 0.161 | 0.193 | 0.225 |
| **Son host** | 0.30 | 0.62 | 0.94 | 0.126 | 0.158 | 0.190 | 0.222 | 0.254 |
| **Broadcast** | 0.31 | 0.63 | 0.95 | 0.127 | 0.159 | 0.191 | 0.223 | 0.255 |

Sonraki tablo, son sekiz subnet'i göstermektedir:

| **Subnet** | 255.0 | 255.32 | 255.64 | 255.96 | 255.128 | 255.160 | 255.192 | 255.224 |
|---|---|---|---|---|---|---|---|---|
| **İlk host** | 255.1 | 255.33 | 255.65 | 255.97 | 255.129 | 255.161 | 255.193 | 255.225 |
| **Son host** | 255.30 | 255.62 | 255.94 | 255.126 | 255.158 | 255.190 | 255.222 | 255.254 |
| **Broadcast** | 255.31 | 255.63 | 255.95 | 255.127 | 255.159 | 255.191 | 255.223 | 255.255 |

## Zihinden Subnet'leme: KlasB Adresler

Deli misiniz? Kafadan KlasB adreslerini subnet'lemek mi? Aslında yazmaktan daha kolaydır. Dalga geçmiyorum! Gelin nasıl olduğunu size göstereyim:

**Soru:** 172.16.10.33 255.255.255.224 (/27) 'nin subnet ve broadcast adresleri nelerdir?

**Cevap:** İlgili oktet, dördüncü oktet'tir. 256–224= 32. 32+32=64. Bingo : 33, 32 ve 64 arasındadır. Bununla beraber, üçüncü oktet, subnet bölümüyle ilgilidir, böylece cevap, 10.32 subnet'i olacaktır. Broadcast, sonraki subnet 10.64 olduğundan, 10.63'tür. Bu oldukça basit bir örnekti.

**Soru:** 172.16.66.10 255.255.192.0'ın (/18) subnet ve broadcast adresleri nelerdir?

**Cevap:** İlgili oktet, dördüncü yerine üçüncü oktet'tedir. 256–192=64, 0, 64, 128. Subnet, 172.16.64.0'dır. Broadcast, sonraki subnet 128.0 olduğundan, 172.16.127.255'tir.

**Soru:** 172.16.50.10 255.255.224.0'ın (/19) subnet ve broadcast adresleri nelerdir?

**Cevap:** 256-224=0, 32, 64 (daima 0'dan saymaya başladığımızı hatırlayın). Subnet, 172.16.32.0'dır ve broadcast, 64.0 sonraki subnet olduğundan, 172.16.63.255'dir.

**Soru:** 172.16.46.255 255.255.240.0'ın (/20) subnet ve broadcast adresleri nelerdir?

**Cevap:** 256–240=16. Bizi üçüncü oktet ilgilendiriyor. 0, 16, 32, 48. Bu subnet adresi, 172.16.32.0 subnet'inde olmalı ve broadcast, 48.0, sonraki subnet olduğundan, 172.16.47.255 olmalıdır. Bu nedenle, 172.16.46.255, geçerli bir host adresidir.

**Soru:** 172.16.45.14 255.255.255.252'nin (/30) subnet ve broadcast adresleri nelerdir?

**Cevap:** İlgili oktet nerededir? 256-252=0, 4, 8, 12, 16 (dördüncü oktette). Subnet, 172.16.45.12 dir. Sonraki subnet 172.16.45.16 olduğundan, broadcast, 172.16.45.15 'tir.

**Soru:** 172.16.88.255/20 host'unun subnet ve broadcast adresleri nelerdir?

**Cevap:** /20 nedir? Buna cevap veremiyorsanız, bu soruyu cevaplayamazsınız, değil mi? /20, 255.255.240.0'dır , üçüncü oktet'te bir 16 blok boyutunu verir ve dördüncü oktette subnet bit'i olmadığından, cevap daima dördüncü oktetteki 0 ve 255'tir. 0, 16, 32, 48, 64, 80, 96…bingo. 88, 80 ve 96 arasındadır, bu nedenle subnet, 80.0'dır ve broadcast adresi, 95.255'tir.

**Soru:** Router, bir interface'inden, 172.16.46.191/26 hedef adresi ile bir paket alıyor. Router bu paketi ne yapacaktır?

**Cevap:** Onu kabul etmeyecektir. Sebebini biliyor musunuz? 172.16.46.191/26, bir 255.255.255.192 mask'ıdır, bize 64 bit blok boyutu verir. Subnet'lerimiz 0, 64, 128'dir. 191, 128 subnet'inin broadcast'idir, bu nedenle varsayılan olarak bir router, herhangi bir broadcast paketini atacaktır.

## KlasA adreslerini Subnet'lemek

KlasA subnet'leme, KlasB veya KlasC'den farklı çalışmamaktadır, fakat çalışmak için KlasB adresinde 16, KlasC adresinde 8 yerine, 24 bit vardır.

Tüm KlasA mask'larını listeleyerek başlayalım:

```
255.0.0.0     (/8)
255.128.0.0   (/9)          255.255.240.0   (/20)
255.192.0.0   (/10)         255.255.248.0   (/21)
255.224.0.0   (/11)         255.255.252.0   (/22)
255.240.0.0   (/12)         255.255.254.0   (/23)
255.248.0.0   (/13)         255.255.255.0   (/24)
```

```
255.252.0.0   (/14)          255.255.255.128   (/25)
255.254.0.0   (/15)          255.255.255.192   (/26)
255.255.0.0   (/16)          255.255.255.224   (/27)
255.255.128.0   (/17)        255.255.255.240   (/28)
255.255.192.0   (/18)        255.255.255.248   (/29)
255.255.224.0   (/19)        255.255.255.252   (/30)
```

İşte budur. Host'ları tanımlamak için en az 2 bit bırakmalısınız. Ve şimdiye kadar modeli görebildiğinizi umuyorum. Bunu, KlasB ve KlasC subnet'i ile aynı yöntemle yapacağımızı hatırlayın. Basit olarak daha fazla host bit'ine sahibiz ve KlasB ve KlasC ile kullandığımızla aynı subnet numaralarını kullanırız. Fakat bu numaraları, ikinci oktet'te kullanarak başlarız.

## Subnet'leme Uygulama Örnekleri: KlasA Adresleri

Bir IP adresi ve subnet mask'ına baktığınızda, subnet'ler için kullanılan bit'leri, belirli host'lar için kullanılan bit'lerden ayırabilmelisiniz. Bu zorunludur. Hala bu kavramı anlamakla uğraşıyorsanız bölüm 2'deki "IP Adresleme" bölümünü tekrar okuyun. Size, subnet ve host bit'leri arasındaki farklılığı nasıl belirleyeceğimizi gösterir ve bazı şeylerin netleşmesine yardımcı olacaktır.

### Uygulama Örneği#1A: 255.255.0.0 (/16)

KlasA adresleri, bir 255.0.0.0 varsayılan mask kullanırlar. Host adreslemesine en az 2 bit ayırmak zorunda olduğunuzdan, subnet'leme için 22 bit kalır. Bir KlasA adresi ile 255.255.0.0 mask'ı, 8 subnet mask'ı kullanır.

- Subnet'ler? $2^8 = 256$.
- Host'lar? $2^{16} – 2 = 65{,}534$.
- Geçerli subnet'ler? İlgili oktet nedir? 256 – 255 = 1. 0, 1, 2, 3, vs. (ikinci oktet'teki hepsi). Subnet,10.255.0.0'a kadar 10.0.0.0, 10.1.0.0, 10.3.0.0 vs. olabilir.
- Her subnet için broadcast adresi?
- Geçerli host'lar?

Aşağıdaki tablo, ilk iki ve son iki subnet'i, geçerli host aralığını ve özel KlasA 10.0.0.0 ağı için broadcast adreslerini göstermektedir:

| **Subnet** | 10.0.0.0 | 10.1.0.0 | ... | 10.254.0.0 | 10.255.0.0 |
|---|---|---|---|---|---|
| **İlk host** | 10.0.0.1 | 10.1.0.1 | ... | 10.254.0.1 | 10.255.0.1 |
| **Son host** | 10.0.255.254 | 10.1.255.254 | ... | 10.254.255.254 | 10.255.255.254 |
| **Broadcast** | 10.0.255.255 | 10.1.255.255 | ... | 10.254.255.255 | 10.255.255.255 |

### Uygulama Örneği#2A: 255.255.240.0 (/20)

255.255.240.0, bize 12 subnet'leme bit'i verir ve host adreslemesi için 12 bit kalır.

- Subnet'ler? $2^{12} = 4096$.
- Host'lar? $2^{12} – 2 = 4094$.
- Geçerli subnet'ler? İlgili oktet nedir? 256 – 240 = 16. İkinci oktet'teki subnet'ler, 1'in blok boyutudur ve üçüncü oktet'teki subnet'ler, 0,16,32'dir.
- Her subnet için broadcast adresi?
- Geçerli host'lar?

| **Subnet** | 10.0.0.0 | 10.0.16.0 | 10.0.32.0 | ... | 10.255.240.0 |
|---|---|---|---|---|---|
| **İlk host** | 10.0.0.1 | 10.0.16.1 | 10.0.32.1 | ... | 10.255.240.1 |
| **Son host** | 10.0.15.254 | 10.0.31.254 | 10.0.47.254 | ... | 10.255.255.254 |
| **Broadcast** | 10.0.15.255 | 10.0.31.255 | 10.0.47.255 | ... | 10.255.255.255 |

Yukarıdaki tablo, bazı host aralığı örneklerini (ilk üç ve son subnet'leri) göstermektedir:

### Uygulama Örneği#3A: 255.255.255.192 (/26)

Subnet'leme için ikinci, üçüncü ve dördüncü oktetleri kullanarak bir tane daha örnek yapalım.

- Subnet'ler? $2^{18}$ = 262,144.
- Host'lar? $2^6 - 2 = 62$.
- Geçerli subnet'ler? İkinci ve üçüncü oktetlerde, blok boyutu 1 ve dördüncü oktette, blok boyutu 64'dür.
- Her subnet için broadcast adresi?
- Geçerli host'lar?

Aşağıdaki tablo, KlasA 255.255.255.192 mask'ındaki ilk dört subnet'i ve onların geçerli host ve broadcast'lerini göstermektedir:

| **Subnet** | 10.0.0.0 | 10.0.0.64 | 10.0.0.128 | 10.0.0.192 |
|---|---|---|---|---|
| **İlk host** | 10.0.0.1 | 10.0.0.65 | 10.0.0.129 | 10.0.0.193 |
| **Son host** | 10.0.0.62 | 10.0.0.126 | 10.0.0.190 | 10.0.0.254 |
| **Broadcast** | 10.0.0.63 | 10.0.0.127 | 10.0.0.191 | 10.0.0.255 |

Aşağıdaki tablo, son dört subnet'i ve onların geçerli host ve broadcast'lerini göstermektedir:

| **Subnet** | 10.255.255.0 | 10.255.255.64 | 10.255.255.128 | 10.255.255.192 |
|---|---|---|---|---|
| **İlk host** | 10.255.255.1 | 10.255.255.65 | 10.255.255.129 | 10.255.255.193 |
| **Son host** | 10.255.255.62 | 10.255.255.126 | 10.255.255.190 | 10.255.255.254 |
| **Broadcast** | 10.255.255.63 | 10.255.255.127 | 10.255.255.191 | 10.255.255.255 |

### Zihinden Subnet'leme: KlasA Adresleri

Kulağa zor geliyor, fakat KlasB ve KlasC ile olduğu gibi, numaralar aynıdır; sadece ikinci oktet'ten başlarız. Bunu kolay yapan nedir? Sadece, en geniş blok boyutuna sahip olan oktet için kaygılanırsınız. (tipik olarak ilginç oktet olarak belirtilir; 0 veya 255 dışındakilerdir). Örneğin, Bir KlasA ağı ile 255.255.240.0 (/20). İkinci oktet'in blok boyutu 1'dir, bu nedenle, bu oktet'te listelenen herhangi bir numara, bir subnet'tir. Üçüncü oktet, bir 240 mask'ıdır, yani, üçüncü oktet'te 16 blok boyutuna sahibiz. Şayet host ID'niz, 10.20.80.30 ise, subnet, broadcast adresi ve geçerli host aralığı nedir?

İkinci oktetteki subnet, 1 blok boyutu ile 20'dir, fakat üçüncü oktet 16 blok boyutundadır. Bu nedenle, onları saymayacağız: 0, 16, 32, 48, 64, 80, 96…(bu arada, şimdi 16 artırımıyla sayabilirsiniz, değil mi?). Bu subnet'imizi 10.20.80.0 yapar, sonraki subnet 10.20.96.0 olduğundan broadcast 10.20.95.255'dir. Tanımlı host aralığı 10.20.80.1'den 10.20.95.254'e kadardır. Evet, yalan yok. Şayet blok boyutunuzu kavradıysanız, bunu gerçekten zihinden hesaplayabilirsiniz!

Gelin, sadece eğlence için bir örnek daha yapalım!

```
Host IP: 10.1.3.65/23
```

İlk olarak, şayet /23 'ün ne olduğunu bilmiyorsanız, bu soruyu cevaplayamazsınız. O, 255.255.254.0'dır. Buradaki ilginç oktet, üçüncü olandır: 256-254=2. Üçüncü oktet'teki subnet'lerimiz, 0, 2, 4, 6 vs.dir. Bu sorudaki host, subnet 2.0 'dadır ve sonraki subnet, 4.0 'dır. Bu nedenle broadcast adresi, 3.255'dir. Ve 10.1.2.1 ve 10.1.3.254 arasındaki tüm adresler, geçerli bir host adresi olarak kabul edilir.

## Variable Lenght Subnet Mask'lar (VLSM'ler)

Bir bölümün tamamını kolayca Variable Length Subnet Mask (VLSM) için ayırabilirdim, onun yerine bir network alıp, farklı network tasarım türlerinde farklı uzunlukta subnet mask'ları kullanarak

birçok network oluşturmanın basit bir yolunu göstereceğim. Bu VLSM ağ kurulumu olarak isimlendirilir ve bu bölümün başında belirttiğim başka konuyu gündeme getirir: Classful ve classless ağ kurulumu.

Ne RIPv1 ne de IGRP routing protokolleri subnet bilgisi için bir alana sahiptirler, bu nedenle subnet bilgisi atılır. Şayet RIP çalışan bir router, belirli bir değerde subnet mask'ına sahipse classful adres aralığındaki tüm interface'lerinin, aynı subnet mask'a sahip olduğu düşünülür. Bu classful routing olarak belirtilir ve RIP ile IGRP'nin classful routing protokolleri olduğu kabul edilir. (RIP ve IGRP hakkında, bölüm 6 "IP Routing"de bahsedeceğim.) Şayet RIP ve IGRP çalışan bir ağda subnet mask uzunluklarını karıştırıp karşılaştırırsanız, bu network çalışmayacaktır!

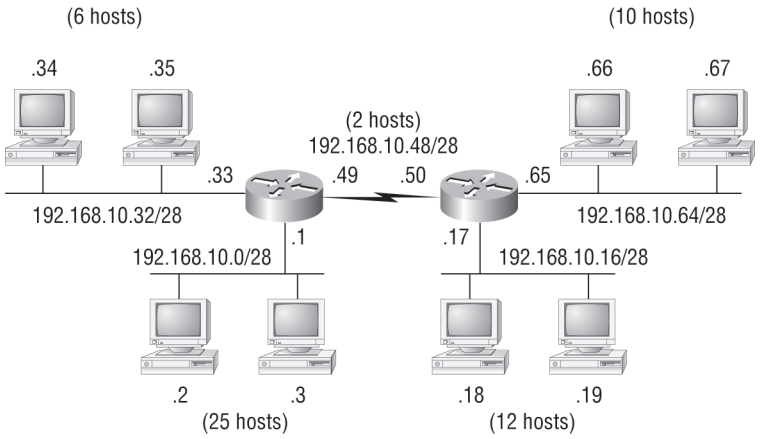
Bununla beraber classless routing protokolleri, subnet bilgisinin yayınlanmasını destekler. Bu yüzden, RIPv2, EIGRP ve OSPF gibi routing protokolleri ile VLSM kullanabilirsiniz. (EIGRP ve OSPF, bölüm 7'de tartışılacaktır). Bu network tiplerinin faydası, onunla IP adres aralığı grupları kazanmanızdır.

İsim önermek gibi, VLSM'lerle, farklı router interface'leri için farklı subnet mask'larına sahip olabiliriz. Classful network tasarımının neden verimsiz olduğunun bir örneğini görmek için Şekil 3.3'e bakın.

Bu şekle bakarak, her biri iki LAN'a sahip iki router'ın birbirlerine bir WAN seri linkle bağlandıklarını fark edeceksiniz. Tipik bir classful network tasarımında (RIP veya IGRP routing protokolleri), bir ağı şu şekilde subnet'lere ayırabilirsiniz:

192.168.10.0 = Network

255.255.255.240 (/28) = Mask




***Şekil 3.3:*** *Tipik classful ağı.*

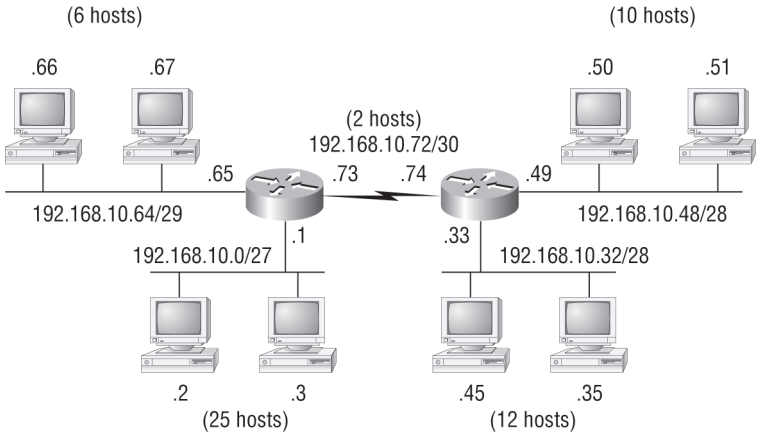
Subnet'lerimiz, 0, 16, 32, 48, 64, 80, vs olabilir. Bu bizim ağ topluluğumuza 16 subnet tanımlamamızı sağlar. Fakat her network için kaç kullanılabilir host olacaktır? Şimdiye kadar muhtemelen tahmin ettiğiniz gibi, her subnet sadece 14 host sağlar. Bu, her subnet 14 geçerli hotsa sahip demektir. Tek LAN bile, tüm hostlar için ihtiyaç duyulan, yeterli adrese sahip değildir! Fakat noktadan-noktaya WAN linki de 14 geçerli adrese sahiptir. Bu WAN linkinden bazı geçerli host'ları çalıp onları LAN'larımıza verememiz çok kötüdür!

Tüm host ve router interface'leri aynı mask'a sahiptir, bu classful routing olarak belirtilir. Ve şayet bu ağın daha verimli olmasını istiyorsak, kesinlikle her router interface'ine farklı mask'lar eklemeye ihtiyacımız vardır.

Fakat hala bir problem vardır, iki router arasındaki link asla iki geçerli host'tan fazlasını kullanmayacaktır! Bu, değerli IP adres aralıklarını çöpe atar ve bahsedeceğim VLSM network tasarımı kullanmanın ana sebebi budur.

## VLSM Tasarımı

Gelin Şekil 3.3'ü alalım ve classless bir tasarım kullanalım. Bu, Şekil 3.4'te görülen yeni network olacaktır. Önceki örnekte, adres aralığını çöpe atmıştık. Her router interface'i ve host, aynı subnet mask'ını kullandığından, bir LAN, yeterli adreslere sahip değildi. Çok iyi bir şey değil. Doğru olan, sadece her router interface'nin ihtiyacı kadar host sayısında sağlamaktır. Ve şayet LAN linklerimizde bir /30 ve LAN'larımızda /27, /28 ve /29 kullanırsak, her WAN interface'i için 2 host ve her LAN interface'i için 30, 14 ve 8 host'a sahip olacağız. Bu büyük bir değişiklik sağlar. Sadece her LAN için doğru sayıda hotsa sahip olmayız, aynı ağı kullanarak, hala daha fazla WAN ve LAN eklemek için yerimiz olur!




***Şekil 3.4:*** *Classless network' tasarımı.*

NOT

*Ağınızda bir VLSM tasarımı kurmak için route güncellemeleri ile subnet mask bilgisini gönderen bir routing protokolüne ihtiyacınız olduğunu hatırlayın. Bu RIPv2, EIGRP ve OSPF olabilir. RIPv1 ve IGRP, classless ağlarda çalışmayacaktır ve classless routing protokolü olarak kabul edilirler.*

## VLSM Tasarımı niçin can sıkar?

*Yeni bir işyerine henüz girdiniz ve onu mevcut ağa eklemeye ihtiyacınız var. Yeni bir IP adres planıyla başlamanızda bir sorun yok. Bir VLSM classless ağ mı, yoksa bir classful ağ mı kullanmalısınız?*

*Gelin, çok sayıda adres aralığına sahip olduğunuzu düşünelim. Firma ortamında KlasA 10.0.0.0 özel network adresi kullanıyorsunuz ve IP adreslerinizin tükenmesini hayal bile edemiyorsunuz. VLSM tasarım prosesi ile neden canınızı sıkmak isteyesiniz?*

*Güzel soru. Güzel de bir cevabı var!*

*Ağınızın belirli alanlarına bitişik adres blokları oluşturarak, ağınızı kolayca summarize edebilir ve bir routing protokolü ile route güncellemelerini minimumda tutabilirsiniz. Binalar arasında sadece bir summary route gönderip, aynı neticeyi alabilecekken, neden binalar arasında yüzlerce network yayınlamayı isteyeceksiniz?*

*Summary route'ların ne olduğu konusunda kafanız karışıksa, açıklamama izin verin. Supperneting olarak ta bilinen summarization, ayrı ayrı yerine, bir yayında birçok route'ı yayınlayarak en verimli yolla route güncellemesi sağlar. Bu, tonlarca bant genişliği kazancı sağlar ve router işleyişini minimuma indirir. Her zamanki gibi, summary route'larınızı yapılandırmak için adres bloklarını (blok boyutlarının tüm network türlerinde kullanıldığını hatırlayın) kullanın ve ağınızın performansını izleyin.*

*Fakat sadece ağınızı özenle tasarladığınızda, summarization'ın düzgün çalıştığını bilin. Şayet, IP subnet'lerini, ağdaki bir lokasyona dikkatsizce dağıtırsanız, artık herhangi bir summary sınırınız kalmadığını anında fark edersiniz. Ve bunsuz, summary route oluşturmak için fazla ileri gidemezsiniz, bu nedenle dikkatli olun!*

## VLSM Ağlar Oluşturmak

VLSM'leri çabuk ve etkili bir şekilde oluşturmak için VLSM mask'larını oluştururken, blok boyutları ve tablolarının beraber nasıl çalıştıklarını anlamalısınız. Tablo 3.3, size, KlasC ağları ile VLSM oluşturduğunuzda kullanılan blok boyutlarını verir. Örnek olarak, şayet 25 host'a ihtiyacınız varsa, 32 blok boyutuna sahip olmalısınız. Eğer, 11 host'a ihtiyacınız varsa, 16 blok boyutu kullanırsınız. 40 host'a ihtiyacınız olursa? O zaman 64 bloğa ihtiyacınız olacaktır. Sadece blok boyutlarını düzenleyemezsiniz (Tablo 3.3 'te gösterilen blok boyutlarında olmalıdırlar). Bu nedenle tablodaki blok boyutlarını ezberleyin. O kolaydır. Subnet'leme ile kullandığımızla aynı sayılardır.

**Tablo 3.3:** Blok Boyutları

| Prefix | Mask | Host'lar | Blok Boyutu |
|---|---|---|---|
| /25 | 128 | 126 | 128 |
| /26 | 192 | 62 | 64 |
| /27 | 224 | 30 | 32 |
| /28 | 240 | 14 | 16 |
| /29 | 248 | 6 | 8 |
| /30 | 252 | 2 | 4 |

Bir sonraki adım, VLSM tablosu oluşturmaktır. Şekil 3.5, bir VLSM ağı oluşturmakta kullanılan tabloyu göstermektedir. Bu tabloyu kullanmamızın sebebi, kazayla ağları çakıştırmamaktır.

Şekil 3.5'te gösterilen tabloyu çok faydalı bulacaksınız, çünkü o bir network adresi için kullanabileceğiniz tüm blok boyutlarını listelemektedir. Blok boyutlarının, 4 blok boyutuyla başlayıp, 128 blok boyutuna kadar tümünü listelediğine dikkat edin. Şayet 128 blok boyutunda iki ağa sahipseniz, sadece iki ağa sahip olabileceğinizi kolayca görebilirsiniz. 64 blok boyutu ile sadece dört ağa sahip olabilirsiniz ve 4 blok boyutunu kullanırsanız, 64 network olana kadar böylece devam eder. Bunun, network tasarımınızda, `ip subnet-zero` komutunu kullandığınızı göz önüne aldığını unutmayın.

Şimdi sol-alt köşedeki tabloyu doldurun ve sonra subnet'leri tabloya ekleyin.

Gelin şimdiye kadar blok boyutlarıyla ilgili öğrendiklerimizi ve VLSM tablosunu alalım ve Şekil 3.6'daki network için 192.168.10.0 KlasC network adresi kullanarak bir VLSM oluşturalım. Sonra, Şekil 3.7'de gösterildiği gibi VLSM tablosunu doldurun.

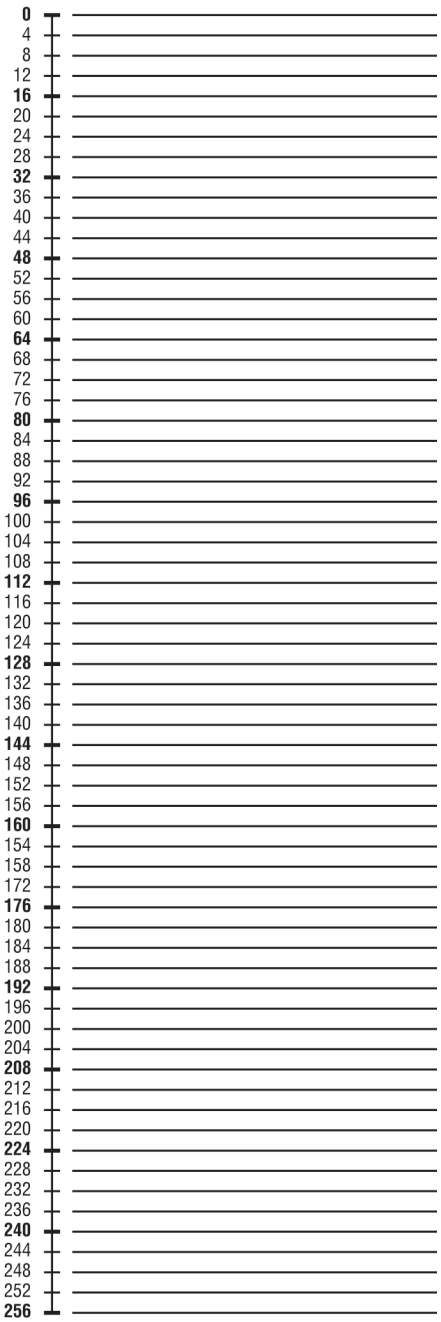
Şekil 3.6'da, dört WAN linki ve birbirine bağlı 4 LAN'a sahibiz. Bize adres alanı kazancı sağlayacak bir VLSM ağı oluşturmaya ihtiyacımız var. İki 32'li blok boyutuna sahip olmamız gibi, bir 16 blok boyutu ile 8 blok boyutu var ve WAN'larımızın her biri, 4 blok boyutuna sahiptir. Bakın ve Şekil 3.7'de VLSM çizelgesini nasıl doldurduğumu görün.

**Variable Length Subnet Masks Worksheet**

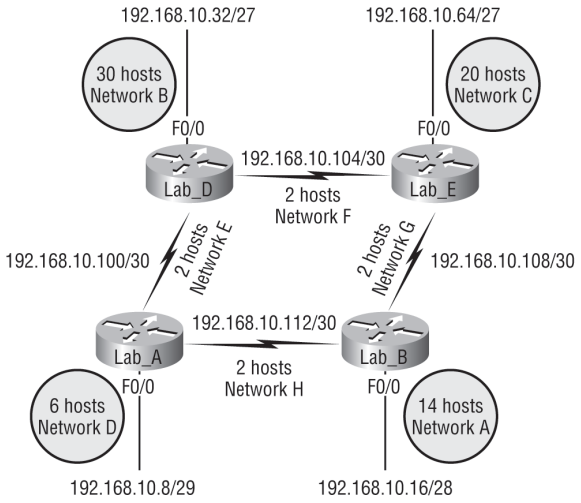
| Subnet | Mask | Subnets | Hosts | Block |
|---|---|---|---|---|
| /26 | 192 | 4 | 62 | 64 |
| /27 | 224 | 8 | 30 | 32 |
| /28 | 240 | 16 | 14 | 16 |
| /29 | 248 | 32 | 6 | 8 |
| /30 | 252 | 64 | 2 | 4 |

**Class C Network** 192.168.10.0

| Network | Hosts | Block | Subnet | Mask |
|---|---|---|---|---|
| A | | | | |
| B | | | | |
| C | | | | |
| D | | | | |
| E | | | | |
| F | | | | |
| G | | | | |
| H | | | | |
| I | | | | |
| J | | | | |
| K | | | | |
| L | | | | |




***Şekil 3.5:*** *VLSM tablosu.*




***Şekil 3.6:*** *VLSM network örneği 1.*

Bu VLSM network tasarımıyla büyümek için hala çok sayıda yere sahibiz.

Bunu, classful routing kullanarak tek subnet mask'ı ile asla başaramazdık. Gelin, başka bir örnek daha yapalım. Şekil 3.8, iki adet 64 blok boyutlu, bir adet 32, beş adet 16 ve üç adet 4 boyutlu 11 network'lü bir ağı göstermektedir.

İlk olarak, VLSM tablonuzu oluşturun ve ihtiyacınız olan subnet'lerle tablonuzu doldurmak için blok boyutu çizelgenizi kullanın. Şekil 3.9, uygun bir çözüm göstermektedir.

Bu çizelgenin tamamını doldurduğumuza ve fazladan sadece bir 4 blok boyutu için yerimiz olduğuna dikkat edin. Sadece bir VLSM ağı ile bu tip adres alanı kazancı sağlayabilirsiniz.

Daima sıfırdan saymaya başladığınız müddetçe, blok boyutlarınızın nereden başlayacağının bir önemi olmayacağını unutmayın. Örnek olarak, şayet 16 blok boyutuna sahipseniz, 0'dan başlamak ve saymak zorundasınız (0, 16, 32, 48 vs.). Bir 16 blok boyutuna, mesela 40 ile veya 16 'nın katlarından başka bir sayı ile başlayamazsınız.

Başka bir örneğe bakalım. 32 blok boyutuna sahipseniz sıfırdan başlamak zorundasınız: 0, 32, 64, 96, vs. Sadece, istediğiniz bir yerden başlayamayacağınızı hatırlayın: Daima, 0'dan sayarak başlamalısınız. Şekil 3.9'daki örnekte, 64 blok boyutuyla, 64 ve 128 ile başladım. Fazla seçme şansım yok. Çünkü seçeneklerim, 0, 64, 128 ve 192'dir. Bununla birlikte, blok boyutlarının doğru artırımlarda oldukları müddetçe istediğim yerlerde 32, 16, 8 ve 4 blok boyutlarını ekledim.

Tamam, adres vermeniz gereken üç lokasyonunuz var ve tüm ağı adreslemek için kullanmak amacıyla aldığınız IP ağınız, 192.168.55.0 'dır. `İp subnet-zero` komutu ve routing protokolü olarak RIPv2 kullanacaksınız. (RIPv2, VLSM ağlarını destekler, RIPv1 desteklemez. Her ikisi, Modül6'da işlenecektir). Şekil 3.10, network diyagramını ve RouterA'nın S0/0 interface'inin IP adresini göstermektedir.

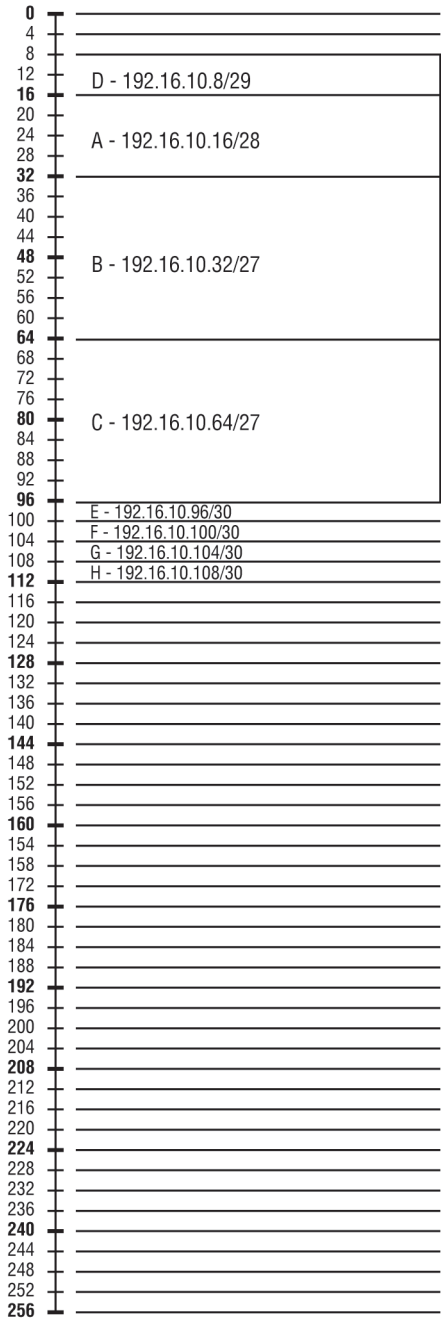
Şeklin sağındaki IP adresi listesinden, her router'ın FastEthernet 0/0 interface'ine ve RouterB'nin serial0/1'ine hangi IP adresi verecektir?

Bu soruya cevap vermek için Şekil 3.10'daki ipuçlarına bakın. İlk ipucu, RouterA'daki S0/0 interface'inin, 192.168.55.2/30 IP adresine sahip olmasıdır, bu kolay bir cevap sağlar. Bildiğiniz gibi, /30, 255.255.255.252 dir, size 4 blok boyutu sağlar. Subnet'leriniz, 0, 4, 8 vs.dir. Bilinen host'lar, 2 boyutunda IP adresine sahip olduğundan, zero subnet'indeki diğer tek geçerli host, 1'dir, bu nedenle üçüncü cevap, RouterB'nin S0/1 interface'i için istediğinizdir.

Sonraki ipuçları her LAN için listelenmiş host numaralarıdır. RouterA'nın, 16 blok boyutu (/28) ile 7 host'a ihtiyacı vardır. Bu aslında oldukça basit bir soruydu. Yapacağınız şey, doğru ipuçlarına bakmanız ve tabii ki blok boyutlarını bilmenizdir.

**Variable Length Subnet Masks Worksheet**

| Subnet | Mask | Subnets | Hosts | Block |
|---|---|---|---|---|
| /26 | 192 | 4 | 62 | 64 |
| /27 | 224 | 8 | 30 | 32 |
| /28 | 240 | 16 | 14 | 16 |
| /29 | 248 | 32 | 6 | 8 |
| /30 | 252 | 64 | 2 | 4 |




**Class C Network** 192.16.10.0

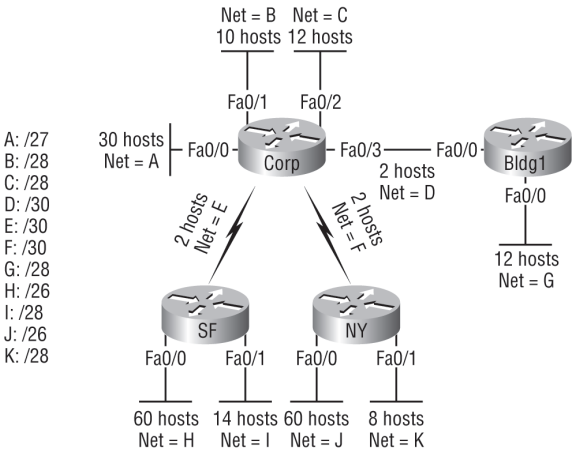
| Network | Hosts | Block | Subnet | Mask |
|---|---|---|---|---|
| A | 12 | 16 | /28 | 240 |
| B | 20 | 32 | /27 | 224 |
| C | 25 | 32 | /27 | 224 |
| D | 4 | 8 | /29 | 248 |
| E | 2 | 4 | /30 | 252 |
| F | 2 | 4 | /30 | 252 |
| G | 2 | 4 | /30 | 252 |
| H | 2 | 4 | /30 | 252 |

***Şekil 3.7:*** *VLSM tablosu, örnek 1.*

Summarization'a geçmeden önce son bir VLSM tasarım örneği yapalım. Şekil 3.12, hepsi RIPv2 çalışan üç router'ı göstermektedir. Mümkün olduğu kadar fazla adres alanı kazanarak, bu ağın ihtiyaçlarını karşılamak içi hangi KlasC adresleme planını kullanırsınız?

Bu, gerçekten güzel bir network. Sizin, çizelgeyi doldurmanızı bekliyor. 64, 32, 16 ve iki 4 blok boyutu vardır. Bu sizin için bir smaç basket olmalı. Şekil 3.13 'deki benim cevabıma bir bakın.

Yaptığım şudur: Subnet 0 ile başlayarak, 64 blok boyutunu kullandım.(yapmak zorunda değildim, 4 blok boyutu ile başlayabilirdim, ama genellikle en geniş blok boyutu ile başlarım ve en küçüğüne devam ederim). Tamam, daha sonra, 32, 16 ve iki 4 blok boyutlarını ekledim. Bu ağa eklemek için hala birçok alan var.



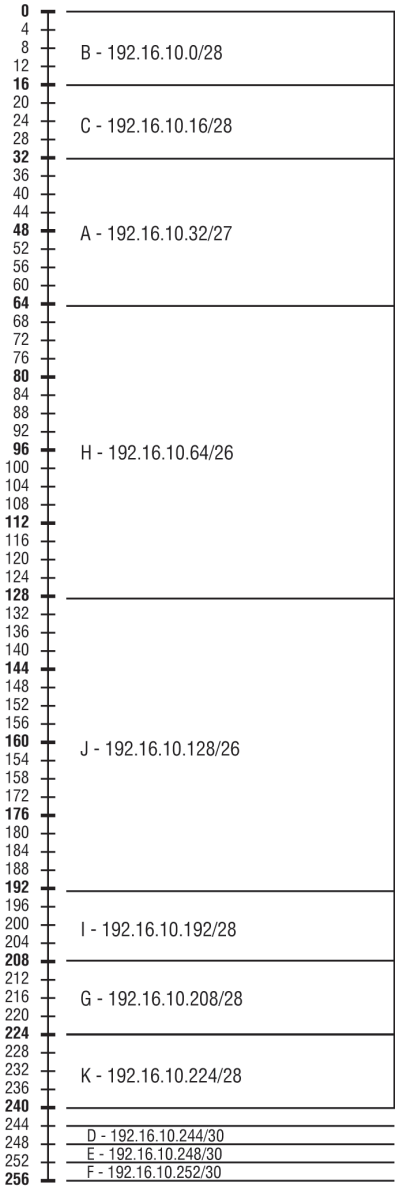

***Şekil 3.8:*** *VLSM network örneği 2.*

**Variable Length Subnet Masks Worksheet**

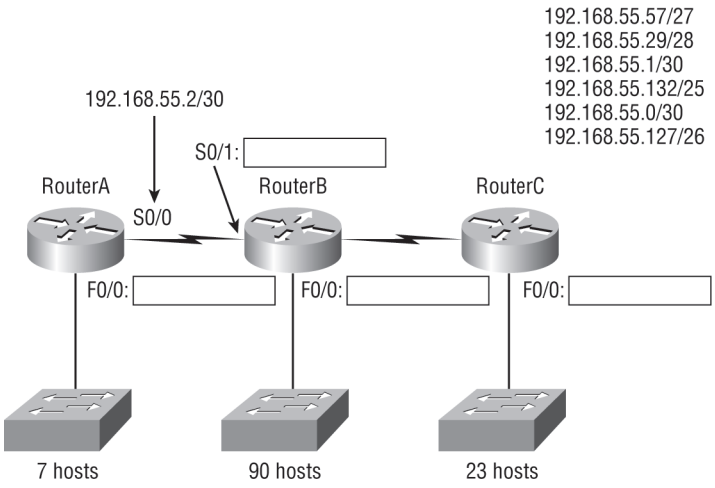
| Subnet | Mask | Subnets | Hosts | Block |
|---|---|---|---|---|
| /26 | 192 | 4 | 62 | 64 |
| /27 | 224 | 8 | 30 | 32 |
| /28 | 240 | 16 | 14 | 16 |
| /29 | 248 | 32 | 6 | 8 |
| /30 | 252 | 64 | 2 | 4 |

**Class C Network** 192.168.10.0

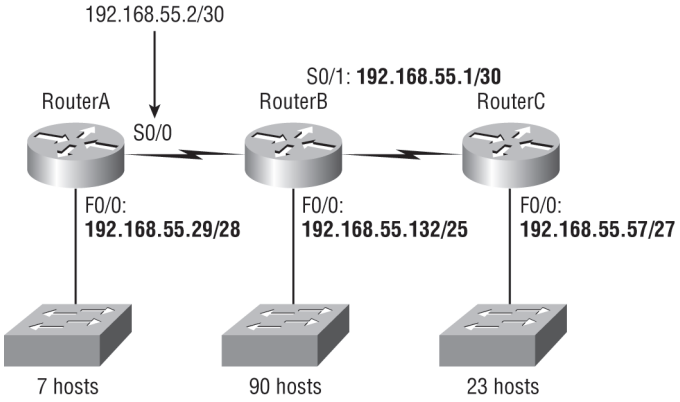
| Network | Hosts | Block | Subnet | Mask |
|---|---|---|---|---|
| A | 30 | 32 | 32 | 224 |
| B | 10 | 16 | 0 | 240 |
| C | 12 | 16 | 16 | 240 |
| D | 2 | 4 | 244 | 252 |
| E | 2 | 4 | 248 | 252 |
| F | 2 | 4 | 252 | 252 |
| G | 12 | 16 | 208 | 240 |
| H | 60 | 64 | 64 | 192 |
| I | 14 | 16 | 192 | 240 |
| J | 60 | 64 | 128 | 192 |
| K | 8 | 16 | 224 | 240 |



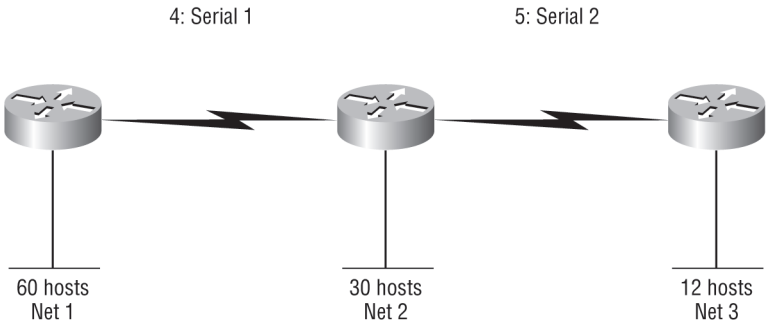

***Şekil 3.9:*** *VLSM tablosu, örnek 2.*



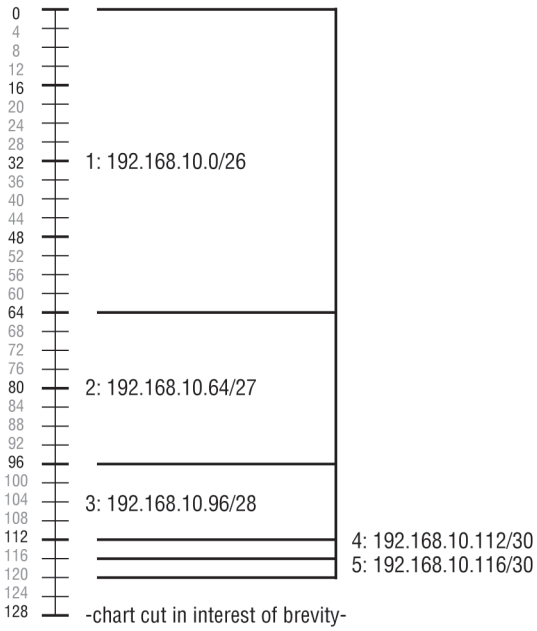

***Şekil 3.10:*** *VLSM tasarım örneği 1.*




***Şekil 3:11:*** *VLSM tasarımına çözüm, örnek 1.*




***Şekil 3.12:*** *VLSM tasarımı örnek 2.*

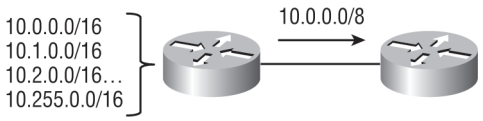



***Şekil 3.13:*** *VLSM tasarımına çözüm, örnek 2.*

# Summarization

Route aggregation olarak ta bilinen summarization, routing protokollerinin birçok ağı tek bir adres gibi yayınlamasına izin verir. Bunun amacı, bellek kazanmak için router'lardaki routing tablolarının boyutunu küçültmektir. Ayrıca, routing tablosunu ayrıştırma ve uzak bir ağa yol bulmak için IP'ye zamanı kısaltır.

Şekil 3.14, bir ağ topluluğunda kullanılabilecek bir summary adresi göstermektedir.




***Şekil 3.14:*** *Bir ağ topluluğunda kullanılan summary adresi.*

Summarization, aslında oldukça basittir, çünkü gerçekte sahip olmamız gereken, subnet'leme ve VLSM tasarımını öğrenmek için kullandığımız blok boyutlarıdır. Örnek olarak, aşağıdaki ağları tek bir network yayınına summarize etmek istiyorsanız, ilk olarak blok boyutunu bulmanız gerekir, sonra cevabınızı kolayca bulabilirsiniz:

**`192.168.16.0 'dan 192.168.31.0 ağına.`**

Blok boyutu nedir? Tam olarak 16 KlasC ağı vardır, bu nedenle, net olarak 16 blok boyutuna sığar.

Şimdi, blok boyutunu biliyorsunuz. Bu ağları tek bir yayına summarize etmek için kullanılan mask'ları ve network adreslerini bulabilirsiniz. Summary adresini yayınlamak için kullanılan network adresi, bloktaki ilk network adresidir. Bu örnekte, 192.168.16.0'dır. Aynı örnekte, bir summary mask'ını çözmek için, 16 blok boyutu sağlamakta hangi mask kullanılmalıdır? 240, doğrudur. Bu 240, summarize yaptığımız oktet olan, üçüncü oktet'te yer almalıdır. Böylece, mask, 255.255.240.0 olmalıdır.

*Bu summary adreslerinin bir router'da nasıl uygulanacağını, nölüm 7'de öğreneceksiniz.*

NOT

Diğer örneğe bakalım:

172.16.32.0 'dan 172.16.50.0 'a kadar ağlar.

Bu, önceki örnekteki kadar net değil. Çünkü iki olası cevabı var ve bunun nedeni şudur: 32 ağından başladığınızdan, blok boyutu seçenekleriniz: 4, 8, 16, 32, 64 vs.dir ve 16 ile 32'nin blok boyutları, bu summary adresi gibi çalışabilir.

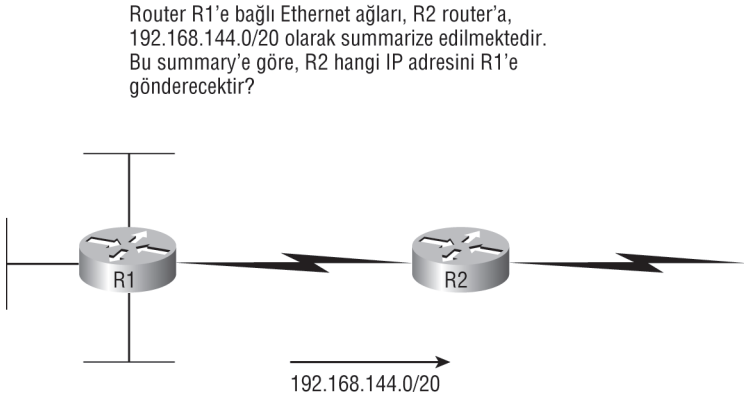
- **Cevap#1:** Şayet 16 blok boyutunu kullanıyorsanız, network adresiniz, 255.255.240.0 (240, bir 16 bloğu sağlar) bir mask'la, 172.16.32.0'dır. Ancak, bu sadece, 32'den 47'e kadarını summarize eder. 48'den 50'ye kadar ağların, tek bir network gibi yayınlanacağı anlamına gelmektedir. Bu muhtemelen en iyi cevaptır, fakat sizin network tasarımınıza bağlıdır. Gelin diğer cevaba bakalım.
- **Cevap#2:** Şayet 32 blok boyutunu kullanıyorsanız, summary adresiniz hala, 172.16.32.0, fakat mask, 255.255.224.0 olacaktır (224, bir 32 blok sağlar). Bu cevaptaki olası problem şudur; 32'den 63'e kadar ağları summarize edecektir ve yalnızca 32'den 50'ye kadar ağlara sahip oluruz. Aynı ağa, sonradan 51'den 63'e kadar ağları eklemeyi planlıyorsanız, endişelenmeye gerek yok, fakat 51'den 63'e herhangi bir ağ, sizin ağınız dışında bir ağdan yayınlanır ve görünürse ağ topluluğunuzda ciddi problemleriniz olabilir! Bir nolu cevabın en güvenli cevap olduğunu söylememin sebebi budur.

Gelin başka bir örneğe bakalım, fakat ona bir host'un perspektifinden bakalım.

Summary adresiniz, 192.168.144.0/244'dür. Bu summary adresine göre gönderilecek host adres aralığı nedir? /20, 192.168.144.0 summary adresi ve 255.255.240.0 mask'ı sağlar.

Üçüncü oktet, 16 blok boyutundadır ve 144 summary adresinden başlar. Bir sonraki 16 bloğu, 160'tır, bu nedenle summary adres aralığı, üçüncü oktet'te 144'ten 159'a kadardır. (16'ları sayabilmelisiniz!) Routing tablosunda bu summary adresine sahip bir router, 192.168.144.1'den 192.168.159.254'e kadar hedef adresi olan IP adreslerini iletecektir.

Sadece iki tane summarization örneği kaldı, ondan sonra hata gidermeye geçeceğiz.



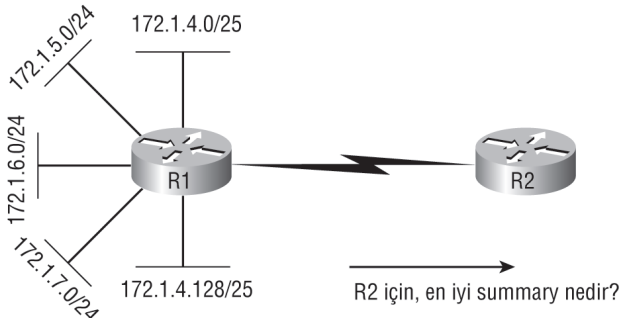

***Şekil 3.15:*** *Summarization örneği 1.*

Şekil 3.15'te, routerR1'e bağlı Ethernet ağları R2'ye 192.168.144.0/20 olarak summarize edilmiştir. Bu summary'e göre R2, hangi IP adres aralığını R1'e iletilecektir?

Router R1'e bağlı olan Ethernet network'leri R2'ye 192.168.144.0/20 olarak summarize edildi.

Endişelenmeyin, bu gerçekten göründüğünden daha kolay bir sorudur. Soru aslında, listelenmiş summary adresine sahiptir: 192.168.144.0/20. Siz zaten, /20'nin, 255.255.255.240.0 olduğunu biliyorsunuz. Bu, üçüncü oktette 16 blok boyutuna sahip olduğunuz anlamına gelir. 144 den başlayarak (aynı zamanda soruda vardır), 16'nın bir sonraki blok boyutu, 160'dır. Bu nedenle, bu oktet'te 159'un üzerine çıkamazsınız. İletilecek IP adresleri, 192.168.144.1'den 192.168.159.255'e kadardır (evet, broadcast adresi iletilecektir).

Tamam, son bir tane daha. Şekil 3.16'da, routerR1'e bağlı beş network var. R2 için en iyi summary adresi nedir?




***Şekil 3.16:*** *Summarization örneği 2.*

R2'ye en iyi summary nedir?

Açık sözlü olacağım, bu Şekil 3.15'ten daha zor bir sorudur. Cevabı bulmanız oldukça zor olacaktır. İlk yapmanız gereken, tüm ağları yazmak ve ortak herhangi bir şey bulup bulamayacağınıza bakmaktır:

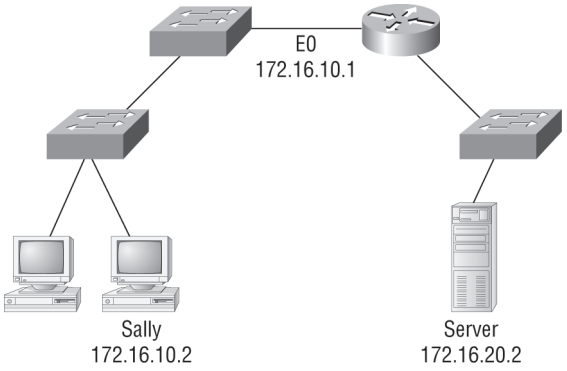
- 172.1.4.128/25
- 172.1.7.0/24
- 172.1.6.0/24
- 172.1.5.0/24
- 172.1.4.0/24

Size ilginç görünen bir oktet var mı? Ben görebiliyorum. Üçüncü oktet. 4, 5, 6, 7, evet, blok boyutu 4'tür. Böylece, 172.1.4.0'ı, 255.255.252.0 mask'ını kullanarak summarize edebilirsiniz. Bu, üçüncü oktette, 4 blok boyutu kullanacağınız anlamına gelmektedir. Bu summary ile iletilen IP adresleri, 172.1.4.1'den 172.1.7.255'e kadardır.

Şimdi, bu summarization bölümünü özetleyelim: Aslında blok boyutunuzu kesinleştirdiğinizde, summary adresini ve mask'ını bulup, uygulamak gerçekten oldukça basittir. Fakat /20'nin ne olduğunu bilmiyorsanız veya 16'nın katlarını sayamıyorsanız, kolayca zor durumda kalacaksınız.

# IP Adreslemesinde Hata Giderme

IP adreslemesinde hata giderimi, açıkça çok önemli bir beceridir. Çünkü işlem boyunca bir yerlerde sorunla karşılaşmak, hemen hemen kesindir, bu başınıza gelecektir. Hayır karamsar biri değilim, sadece gerçekçi olmaya çalışıyorum. Bu acı gerçekten dolayı evde veya işte iken, bir IP ağındaki problemi tespit edip çözebildiğiniz gün sizin için çok iyi olacaktır.




***Şekil 3.17:*** *Basit IP hata giderimi.*

Bu sebeple IP adreslemesinin hata tespitinde Cisco yöntemi'ni size göstereceğim yer burasıdır. Gelin, basit IP probleminizle ilgili örnek için Şekil 3.17'ye bakalım. Zavallı Sally, Windows sunucusuna bağlanamamaktadır. Bunu, Microsoft ekibini arayıp, sunucularının bir çöp yığını olduğunu

ve tüm problemlerinize sebep olduğundan bahsederek halledebilir misiniz? Muhtemelen çok iyi bir fikir değildir. Gelin, onun yerine ağımızı tekrar gözden geçirelim.

Gelin, Cisco'nun izlediği hata giderme adımlarını uygulayarak işe başlayalım. Onlar oldukça basit, fakat bir o kadar da önemlidir. Müşterinin makinesinde olduğunuzu ve uzak bir ağda olan sunucu ile iletişim kuramadığınızı düşünün. Aşağıda, Cisco'nun önerdiği dört hata giderme adımını bulabilirsiniz:

1. Bir DOS penceresi açın ve 127.0.0.1 'e ping atın. Bu sistem tanı ya da loopback adresidir ve şayet ping atabiliyorsanız, IP yığınınızın çalıştığı kabul edilir. Şayet atamazsanız, o zaman IP yığın probleminiz vardır ve host'taki TCP/IP'yi yeniden kurmanız gerekir.

```
C:\>ping 127.0.0.1
Pinging 127.0.0.1 with 32 bytes of data:
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Ping statistics for 127.0.0.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

2. DOS penceresinden, lokal host'un IP adresini pingleyin. Bu başarılırsa, network interface card (NIC)'ınız çalışıyor demektir. Şayet atamıyorsanız, NIC'te problem var demektir. Buradaki başarı, kablonun NIC'e takılı olduğu anlamına gelmez. Sadece host'taki IP protokol yığını, NIC ile iletişimde olabilir (LAN sürücüsü yardımıyla).

```
C:\>ping 172.16.10.2
Pinging 172.16.10.2 with 32 bytes of data:
Reply from 172.16.10.2: bytes=32 time<1ms TTL=128
Reply from 172.16.10.2: bytes=32 time<1ms TTL=128
Reply from 172.16.10.2: bytes=32 time<1ms TTL=128
Reply from 172.16.10.2: bytes=32 time<1ms TTL=128
Ping statistics for 172.16.10.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

3. DOS penceresinden, varsayılan ağ geçidini (router) pingleyin. Şayet ping çalışıyorsa, NIC'in ağa bağlı olduğu ve yerel ağ ile iletişime geçebileceği anlamına gelir. Şayet atamazsa, NIC ile router arasında herhangi bir yerde fiziksel network probleminiz vardır.

```
 C:\>ping 172.16.10.1
Pinging 172.16.10.1 with 32 bytes of data:
Reply from 172.16.10.1: bytes=32 time<1ms TTL=128
Reply from 172.16.10.1: bytes=32 time<1ms TTL=128
Reply from 172.16.10.1: bytes=32 time<1ms TTL=128
Reply from 172.16.10.1: bytes=32 time<1ms TTL=128
Ping statistics for 172.16.10.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
```

```
Approximate round trip times in milli-seconds:
  Minimum = Oms, Maximum = Oms, Average = Oms
```

4. Şayet 1'den 3'e kadarki adımlar başarılı olursa, uzaktaki sunucuyu ping'lemeye çalışın. Bu çalışırsa, lokal host ile uzaktaki sunucu arasında IP iletişimi var demektir. Ayrıca, uzak fiziksel ağında çalıştığını bilirsiniz.

```
C:\>ping 172.16.20.2
Pinging 172.16.20.2 with 32 bytes of data:
Reply from 172.16.20.2: bytes=32 time<1ms TTL=128
Reply from 172.16.20.2: bytes=32 time<1ms TTL=128
Reply from 172.16.20.2: bytes=32 time<1ms TTL=128
Reply from 172.16.20.2: bytes=32 time<1ms TTL=128
Ping statistics for 172.16.20.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
  Minimum = Oms, Maximum = Oms, Average = Oms
```

Şayet kullanıcı, 1'den 4'e kadarki adımların başarılı olmasından sonra, hala sunucuyla iletişim kuramıyorsa muhtemelen bazı isim çözümleme problemleri yaşıyorsunuzdur ve Domain Name System (DNS) ayarlarınızı kontrol etmeniz gerekir. Fakat uzak sunucuya ping atmakta problem varsa, bazı fiziksel network problemleriniz olduğunu bilirsiniz, sunucu makineye gitmeniz ve sorunu bulana kadar, 1'den 3'e kadar adımları uygulamanız gerekir.

IP adres problemlerini ve onları nasıl çözeceğimizi belirlemeden önce, hem bir PC hem de bir Cisco router'dan ağınızda hata gidermede yardımcı olması için kullanabileceğiniz bazı temel DOS komutlarını anlatmak istiyorum (komutlar aynı şeyleri yapıyor olabilir, fakat onlar farklı şekilde çalışır):

Packet InterNet Groper (`ping`)**:** Bir ağda, bir node IP yığınının başladığını ve aktif olup olmadığını test etmek için ICMP echo request ve reply kullanır.

**`traceroute`:** TTL time-out'lar (zaman aşımları) ve ICMP hata mesajları kullanarak, bir hedef ağa giden yoldaki router'ların listesini görüntüler. Bu komut, bir DOS komut sisteminden çalışmayacaktır.

**`tracert`:** `Traceroute` ile aynı komuttur, fakat bir Microsoft Window komutudur ve bir Cisco router'da çalışmayacaktır.

**`arp –a`:** Bir Windows PC'de, IP'den MAC adresi eşleşmesinde kullanılır.

**`show ip arp`:** `arp –a` ile aynı komuttur, fakat bir Cisco router'daki ARP tablosunu görüntüler. `traceroute` ve `tracert` komutları gibi, DOS ve Cisco 'da birbirinin yerine kullanılamaz.

**`ipconfig /all`:** Sadece DOS komut satırından kullanılabilir, size PC network yapılandırmasını gösterir.

Tüm bu adımları geçip, uygun DOS komutlarını kullandığınızda, şayet ihtiyaç duyarsanız, bir problem tespit ettiğinizde ne yaparsınız? Bir IP adresleme hatasını çözmek konusunda nasıl bir yöntem izlersiniz? Gelin, IP adres problemlerini nasıl belirleyeceğimize ve onları nasıl çözeceğimize geçelim.

## IP Adres Problemlerini Belirlemek

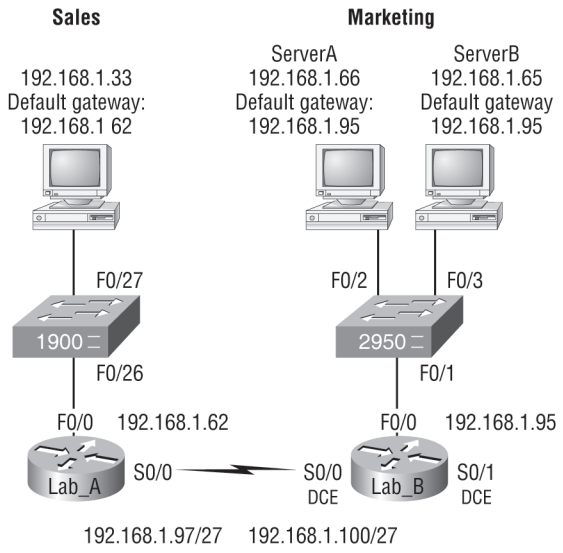
Bir host, router ya da diğer network cihazlarının yanlış IP adresi, subnet mask ve varsayılan ağ geçidi ile yapılandırılması yaygındır. Bu çok sık olduğundan, size IP adres yapılandırma hatalarını nasıl teşhis edip, çözeceğimizi öğreteceğim.

Hata tespitindeki dört basit adımı uygulayıp, problem olduğunu belirlediğinizde, onu bulup çözmeniz gerekmektedir. Bu adımlar gerçekten, network ve IP adresleme planından bilgi almaya yardımcı olur. Şayet olduysa, kendinizi şanslı kabul edin ve loto bileti almaya gidin. Çünkü öyle olması gerektiği halde, çok seyrek olur.

Ağımızı doğru şekilde genişletince, IP adresleme planı dâhil olmak üzere, problemi belirlemek için her host'un IP adresini, mask'ını ve varsayılan ağ geçidi adresini doğrulamaya ihtiyaç duyarsınız. (Fiziksel bir probleminiz yok, ya da varsa da onu çözdüğünüzü farz ediyorum).

Gelin, Şekil 3.18'de gösterilen örneğe bakalım. Satış departmanındaki bir kullanıcı sizi arıyor ve pazarlama departmanındaki SunucuA'ya erişemediğini söylüyor. Ona, pazarlama departmanındaki SunucuB'ye ulaşıp ulaşamadığını soruyorsunuz, fakat bu sunucuya erişim için gerekli haklarının olmadığını söylüyor. Ne yaparsınız?

NOT *Bölüm 5,"Cisco IOS'u Yönetmek"te CDP kullanarak ağınız hakkında nasıl bilgi alacağınızı göstereceğim.*



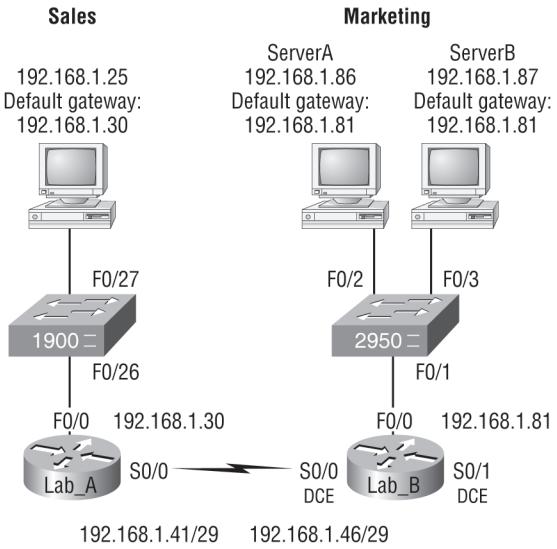

***Şekil 3.18:*** *IP adres problemi.*

İstemciden, yukarıda öğrendiğiniz dört hata giderme adımını gözden geçirmesini istiyorsunuz. 1'den 3'e kadar ki adımlar çalışıyor, fakat 4 çalışmıyor. Şekle bakarak, problemi belirleyebilir misiniz? Network çizimindeki ipuçlarına bakın. İlk olarak, LabA router ile LabB router arasındaki WAN linki, /27 mask olarak görünüyor. Bu mask'ın 255.255.255.224 olduğunu ve bu mask'ı kullanan ağları belirlemeyi zaten biliyorsunuz. Network adresi, 192.168.1.0'dır. Bizim geçerli subnet'lerimiz ve host'larımız nedir? 256-224=32, bu, subnet'lerimizi 32, 64, 96, 128, vs yapar. Şekle bakarak, 32 subnet'inin satış departmanı, 96 subnet'inin WAN linki ve 64 subnet'inin pazarlama departmanı tarafından kullanıldığını görebilirsiniz.

Şimdi, her subnet için geçerli host aralıklarının ne olduğunu belirlemelisiniz. Bu bölümün başında öğrendiklerinizden, subnet adresleri, broadcast adresleri ve geçerli host aralıklarını kolayca belirleyebilmeniz gerekir. Satış LAN'ı için geçerli host'lar, 33'den 62'ye kadardır, sonraki subnet'in, 64 olmasından dolayı, broadcast adresi 63'tür, değil mi? Pazarlama LAN'ı için geçerli host'lar 65'den 94'e kadardır (broadcast, 95'dir) ve WAN linki için 97'den 126'ya kadar (broadcast,127). Şekle bakarak, LabB'deki varsayılan ağ geçidinin, yanlış olduğunu belirleyebilirsiniz. Bu adres, 64 subnet'inin broadcast adresidir, bu nedenle onun geçerli host olmasının hiç şansı yoktur.

Hepsini anladınız mı? Emin olmak için, belki başka bir örnek denemeliyiz. Şekil 3.19, bir network problemini göstermektedir. Satış LAN'ındaki bir kullanıcı, SunucuB'ye erişememektedir. Kullanıcının uygulayacağı dört basit hata tespit adımınız var ve hostun, lokal ağ ile iletişim kurabildiğini, fakat uzak ağ ile kuramadığını tespit ettiniz. IP adresleme problemini bulup, belirleyin.

Şayet, son problemi çözmek için uyguladığınız aynı adımları kullanırsanız, ilk olarak, WAN linkinin (/29 ya da 255.255.255.248) subnet mask'ı sağladığını görebilirsiniz. Sizin geçerli subnet'lerin, broadcast adreslerinin ve geçerli host aralıklarının, bu problemi çözmek için ne olması gerektiğini belirlemeye ihtiyacınız vardır.




***Şekil 3.19:*** *IP adresleme problemi 2.*

248, bir 8 blok boyutudur (256–248=8), bu nedenle subnet'ler, hem sekizden başlar hem de 8'in katlarıyla artar. Şekle bakarak, Satış LAN'ının 24 subnet'te, WAN'ın 40'da ve Pazarlama LAN'ının, 80 subnet'inde olduğunu görürsünüz. Problemi hala anlayamadınız mı? Satış LAN'ı için geçerli host aralığı 25-30'dur ve yapılandırma doğru gözükmektedir. WAN linki için geçerli host aralığı 41-46'dır ve buda doğru görünmektedir. 80 subnet'inin geçerli host aralığı, bir sonraki subnet'in 88 olmasından dolayı 87 broadcast adresiyle, 81-86'dır. SunucuB, subnet'in broadcast adresi ile yapılandırılmıştır.

Tamam, host'lardaki yanlış yapılandırılmış IP adreslerini çözebilirsiniz. Şayet host, IP adresine sahip değilse ve bir tane tanımlamanız gerekirse ne yaparsınız? Yapmanız gereken, LAN'daki diğer host'lara bakmak ve network, mask ile varsayılan ağ geçidini anlamaktır. Gelin host'lara geçerli IP adreslerini nasıl bulup tanımlanacağıyla ilgili birkaç örneğe bakalım.

Bir LAN'da, bir sunucu ve router IP adresi tanımlamanız gerekir. Bu segment'te tanımlı subnet, 192.168.20.24/29'dur ve router'a, ilk kullanılabilir adresin, sunucuya da, son geçerli host ID'sinin tanımlanması gerekmektedir. Sunucuya tanımlanan IP adresi, mask ve varsayılan ağ geçidi nedir?

Bunu cevaplamak için /29'un, 8 blok boyutunu sağlayan, 255.255.255.248 mask'ı olduğunu bilmelisiniz. Subnet 24 olarak bilinir, bir blok 8'de sonraki subnet, 32'dir, bu nedenle 24 subnet'inin broadcast adresi, 31'dir, bu da geçerli host aralığını 25–30 yapar.

**Sunucu IP adresi:** 192.168.20.30

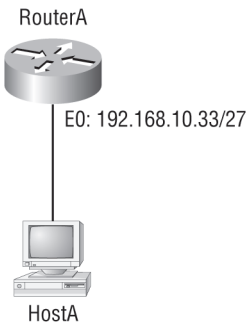
**Sunucu mask :** 255.255.255.248

**Varsayılan Ağ geçidi :** 192.168.20.25 (router'ın IP adresi)

Başka bir örnek için gelin Şekil 3.20'ye bakalım ve bu problemi çözelim.

Ethernet0'daki router'ın IP adresine bakın. Host'a, hangi IP adresi, subnet mask'ı ve geçerli host aralığı tanımlanabilir?

Router'ın Ethernet0'ının IP adresi, 192.168.10.33/27'dir. Zaten bildiğiniz gibi, /27, 32 blok boyutu ile 224 'dür. Router'ın interface'i, 32




***Şekil 3.20:*** *Geçerli host'u bulmak#1.*

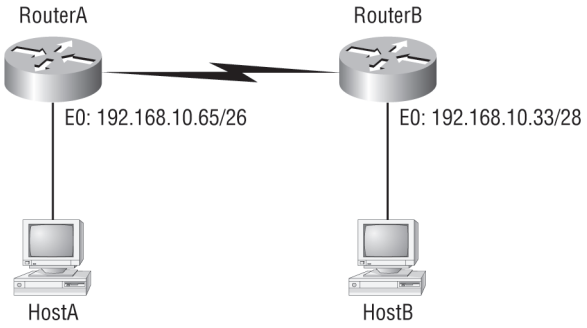
subnet'indedir. Sonraki subnet, 64'dür, bu 32 subnet'inin broadcast adresi, 63'tür ve geçerli host aralığı, 33-62'dir.

**Host IP adresi:** 192.168.10.34-62 (router'a tanımlanan 33 dışındaki herhangi bir adres)

**Mask:** 255.255.255.224

**Varsayılan Ağ geçidi:** 192.168.10.33

Şekil 3.21, Ethernet yapılandırması zaten tanımlanmış iki router göstermektedir. HostA ve HostB'nin host adresleri ve subnet mask'ları nedir?




***Şekil 3.21:*** *Geçerli host'ları bulmak#2.*

RouterA, 192.168.10.65/26 ve RouterB, 192.168.10.33/28 IP adreslerine sahipler. Host yapılandırması nasıldır? RouterA Ethernet0, 192.168.10.64 subnet'inde ve RouterB Ethernet0, 192.168.10.32 ağındadır.

**Host A IP adresi:** 192.168.10.66–126

**Host A mask'ı:** 255.255.255.192

**Host A varsayılan ağ geçidi:** 192.168.10.65

**Host B IP adresi:** 192.168.10.34–46

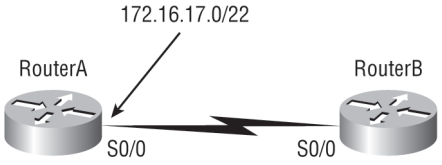
**Host B mask'ı:** 255.255.255.240

**Host B varsayılan ağ geçidi:** 192.168.10.33

Sadece birkaç örnek daha var ve sonra bu modül tarih oluyor. Orda kalın!

Şekil 3.22, iki router'ı göstermektedir; RouterA'daki S0/0 interface'ini yapılandırmanız gerekmektedir. Seri linke tanımlanan network, 172.16.17.0/22'dir. Hangi IP adresi tanımlanabilir?

İlk olarak, /22 CIDR'ın, üçüncü oktet'te 4 blok boyutu yapan, 255.255.252.0 olduğunu bilmelisiniz. 17 listelendiğinden, uygun aralık 16.1'den 19.254'e kadardır. Böylece, örneğin S0/0 IP adresi aralıkta olduğundan 172.16.18.255'dir.

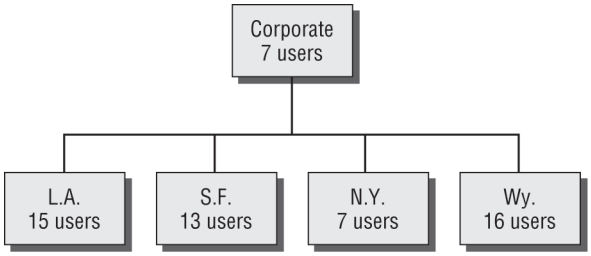



***Şekil 3.22:*** *Geçerli host adresi bulmak#3.*

Tamam, son bir tane! Bir KlasC network ID'niz var ve Şekil 3.23'te belirtilen her şehir için yeterli, uygun host adresi sağlarken, şehir başına uygun bir subnet sağlamanız gerekmektedir. Mask'ınız nedir?

Aslında, bu, muhtemelen tüm gün boyunca yaptığınız en kolay örnektir. Ben, gerekli 5 subnet saydım ve Wyoming ofisinin, 16 kullanıcıya ihtiyacı vardır.(daima, en fazla host'a ihtiyaç duyan ağa bakın). Wyoming ofis için hangi blok boyutuna gerek vardır? 32. (Daima 2 çıkartmak zorunda olduğunuzdan, 16 blok boyutunu kullanamazsınız!) 32 blok boyutunu size hangi mask sağlar? 224. Bingo! Bu, her biri 30 host ile 8 subnet sağlar.

Başardınız. Tamam, biraz dinlenin, sonra geri gelin ve yazılı lab ve gözden geçirme sorularını tamamlayın.




***Şekil 3.23:*** *Geçerli subnet mask'ı bulmak.*

# Özet

Bölüm 2 ve 3'ü okuyup ilk seferinde her şeyi anladınız mı? Şayet öyleyse, bu mükemmel, tebrikler! Muhtemelen, birkaç defa konu kaçırdınız. Söylediğim gibi, bu genelde olur, bu yüzden stres yapmayın. Gerçekten iyi olmadan önce, şayet her bölümü bir defadan fazla, hatta 10 defa okumak zorunda olsanız da, kendinizi kötü hissetmeyin.

Bu bölüm size, IP adreslemesini anlamanın önemini anlattı. Bu bölümü okuduktan sonra, IP adreslerini ezbere subnetleyebilmeniz gerekir. Ayrıca, basit VLSM ağlarını nasıl tasarlayıp kuracağınızı bilmelisiniz.

Cisco'nun hata tespiti yöntemlerini de anlamış olmalısınız. Bir network/IP adreslemesi probleminin olduğu yeri daraltmaya çalıştığınızda ve sonra onu çözmek için nasıl sistematik ilerleneceğini bildiğinizde, uyguladığınız Cisco tavsiye ettiği dört adımı hatırlamalısınız. İlave olarak, bir network çizimine bakarak, geçerli IP adreslerini ve subnet mask'larını bulabilmelisiniz.

# Sınav Gereklilikleri

**Ezberden subnet'leme adımlarını hatırlamak:** IP adreslemenin ve subnet'lemenin nasıl çalıştığını anlayın. İlk olarak, 256-subnet mask hesaplaması kullanarak blok boyutunuzu belirleyin. Sonra, subnet'lerinizi sayın ve her subnet için broadcast adreslerini belirleyin. O, daima bir sonraki subnet'ten hemen önceki numaradır. Geçerli host'larınız, subnet adresi ile broadcast adresi arasındaki numaralardır.

**Farklı blok boyutlarını anlamak.** Bu, IP adreslemeyi ve subnet'lemeyi anlamanın önemli bir bölümüdür. Geçerli blok boyutları daima, 4, 8, 16, 32, 64, 128 vs.dir. 256-subnet mask matematiği kullanarak blok boyutunuzu belirleyebilirsiniz.

**Dört sistem tanı adımını hatırlamak.** Hata tespiti için Cisco'nun tavsiye ettiği dört basit test, loopback adresini pinglemek, NIC'i pinglemek, varsayılan ağ geçidini pinglemek ve uzak cihazı pinglemektir.

**Bir IP adresleme problemini bulup düzeltebilmelisiniz.** Cisco'nun önerdiği dört hata tespiti adımını denediğinizde, ağdan bilgi alarak ve ağınızdaki geçerli ve geçersiz host'larınızı bularak, IP adresleme problemini belirleyebilmeniz gerekmektedir.

**Host'unuzdan ya da bir Cisco router'dan kullanabileceğiniz hata tespit araçlarını anlayın.** `Ping 127.0.0.1`, lokal IP yığınınızı test eder. `tracert`, bir ağ topluluğu boyunca, bir hedefe

giden bir paketin yolunu izlemek için Windows DOS komutudur. Cisco router'lar, `traceroute` ya da kısaca `trace` kullanır. Windows ve **DOS komutları ile sakın karıştırmayın.** Aynı çıktıları ürettikleri halde, aynı komut satırından çalışmazlar. `ipconfig /all`, bir DOS komut satırından, PC'nizin network yapılandırmasını görüntüleyecektir ve `arp –a` (yine DOS komut satırından), bir Windows PC'sindeki IP'den MAC adresi eşleşmesini görüntüleyecektir.

# Yazılı Lab'lar 3

Bu bölümde, anlatılan, bilgi ve kavramları anladığınızdan tamamıyla emin olmak için aşağıdaki lab'ları tamamlayacaksınız:

- **Lab 3.1:** Yazılı Subnet Uygulaması #1
- **Lab 3.2:** Yazılı Subnet Uygulaması #2
- **Lab 3.3:** Yazılı Subnet Uygulaması #3

(Yazılı lab'ın cevapları, bu bölüm için gözden geçirme sorularına cevaplardan sonra bulunabilir.)

## Yazılı Lab 3.1: Yazılı Subnet Uygulaması#1

Soru 1'den soru 6'ya kadar subnet, broadcast adresi ve geçerli host aralığını yazın:

1. 192.168.100.25/30
2. 192.168.100.37/28
3. 192.168.100.66/27
4. 192.168.100.17/29
5. 192.168.100.99/26
6. 192.168.100.99/25
7. Bir KlasB ağınız var ve 29 subnet'e ihtiyacınız var. Mask'ınız nedir?
8. 192.168.192.10/29 'un broadcast adresi nedir?
9. Bir KlasC /29 mask'ı ile kaç adet host mümkündür?
10. 10.16.3.65/23 host ID'si için subnet nedir?

## Yazılı Lab 3.2: Yazılı Subnet Uygulaması

Verilen bir KlasB ağı ve belirlenen net bit'leri (CIDR) ile her mask için uygun host adreslerinin numaraları ve subnet mask'larını tespit etmek için aşağıdaki tabloyu tamamlayın.

| Classful Adres | Subnet Mask | Subnet başına host sayısı ($2^x - 2$) |
|---|---|---|
| /16 | | |
| /17 | | |
| /18 | | |
| /19 | | |
| /20 | | |
| /21 | | |
| /22 | | |
| /23 | | |
| /24 | | |
| /25 | | |
| /26 | | |
| /27 | | |
| /28 | | |

| Classful Adres | Subnet Mask | Subnet başına host sayısı ($2^x - 2$) |
|---|---|---|
| /29 | | |
| /30 | | |

## Yazılı Lab 3.3: Yazılı Subnet Uygulaması

| Ondalıklı IP adresi | Adres sınıfı | Subnet ve Host bit'i sayıları | Subnet sayısı (2x) | Host sayısı (2x – 2) |
|---|---|---|---|---|
| 10.25.66.154/23 | | | | |
| 172.31.254.12/24 | | | | |
| 192.168.20.123/28 | | | | |
| 63.24.89.21/18 | | | | |
| 128.1.1.254/20 | | | | |
| 208.100.54.209/30 | | | | |

(Yazılı lab'ın cevapları, bu bölüm için gözden geçirme sorularına cevaplardan sonra bulunabilir.)

# Gözden Geçirme Soruları

NOT *Aşağıdaki sorular, bu modülün materyallerini anladığınızı test etmek için tasarlanmıştır. İlave sorulara nasıl sahip olacağınızla ilgili daha fazla bilgi için bu kitabın Giriş bölümüne bakın.*

1. 255.255.255.224 subnet mask'ı kullanan bir lokal subnet'teki host'lara tanımlanabilecek maksimum IP adres sayısı nedir?

   A. 14

   B. 15

   C. 16

   D. 30

   E. 31

   F. 62

2. Her subnet'te uygun host adres sayısını maksimum yaparken, 29 subnet'e ihtiyacı olan bir ağa sahipsiniz. Doğru subnet mask'ı sağlamak için host alanından kaç bit almalısınız?

   A. 2

   B. 3

   C. 4

   D. 5

   E. 6

   F. 7

3. 200.10.5.68/28 IP adresinde bir host için alt ağ adresi nedir?

   A. 200.10.5.56

   B. 200.10.5.32

   C. 200.10.5.64

   D. 200.10.5.0

4. 172.16.0.0/19 network adresi, kaç subnet ve host sağlar?

   A. 7 subnet, her biri 30 host

   B. 7 subnet, her biri 2,046 host

   C. 7 subnet, her biri 8,196 host

   D. 8 subnet, her biri 30 host

   E. 8 subnet, her biri 2,046 host

   F. 8 subnet, her biri 8,196 host

5. 10.16.3.65/23 IP adresini hangi iki ifade açıklamaktadır?(iki şık seçin.)

   A. Subnet adresi, 10.16.3.0 255.255.254.0.

   B. Subnet'teki en düşük host adresi 10.16.2.1 255.255.254.0.

   C. Subnet'teki son geçerli host adresi, 10.16.2.254 255.255.254.0

   D. Subnet'in broadcast adresi, 10.16.3.255 255.255.254.0.

   E. Network, subnetlenmemiştir.

6. Ağdaki bir host, 172.16.45.14/30 adresine sahipse, bu host'un ait olduğu alt ağ nedir?
   A. 172.16.45.0
   B. 172.16.45.4
   C. 172.16.45.8
   D. 172.16.45.12
   E. 172.16.45.16
7. Bir VLSM ağında, IP adresi israfını azaltmak için noktadan-noktaya WAN linklerinde hangi mask'ı kullanmalısınız?
   A. /27
   B. /28
   C. /29
   D. /30
   E. /31
8. 172.16.66.0/21 IP adresli bir host'un alt ağ adresi nedir?
   A. 172.16.36.0
   B. 172.16.48.0
   C. 172.16.64.0
   D. 172.16.0.0
9. IP adresi 192.168.192.10/29 olan bir router interface'iniz var. Router interface'inide içererek, router interface'ine bağlı LAN'da kaç host, IP adresi alabilir?
   A. 6
   B. 8
   C. 30
   D. 62
   E. 126
10. 192.168.19.24/29 subnet'inde olan bir sunucuyu yapılandıracaksınız. Router, ilk geçerli host adresine sahiptir. Aşağıdakilerden hangisini sunucu makineye tanımlarsınız?
    A. 192.168.19.0 255.255.255.0
    B. 192.168.19.33 255.255.255.240
    C. 192.168.19.26 255.255.255.248
    D. 192.168.19.31 255.255.255.248
    E. 192.168.19.34 255.255.255.240
11. 192.168.192.10/29 IP adresine sahip bir router interface'iniz var. Bu LAN'daki host'ların kullanacağı broadcast adresi nedir?
    A. 192.168.192.15
    B. 192.168.192.31
    C. 192.168.192.63
    D. 192.168.192.127
    E. 192.168.192.255

12. Bir ağı, her biri 16 host'a sahip 5 subnet'e bölmeniz gerekmektedir. Hangi classful subnet mask'ı kullanırsınız?

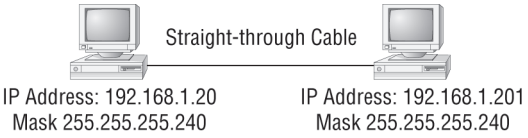
    A. 255.255.255.192

    B. 255.255.255.224

    C. 255.255.255.240

    D. 255.255.255.248

13. Bir network yöneticisi, şekilde gösterildiği gibi, hostA ile hostB'yi direk olarak Ethernet interface'lerinden bağlıyor. Host'lar arasındaki ping denemeleri başarısızdır. Host'lar arasında bağlantıyı sağlamak için ne yapılabilir? (İki şık seçin.)




    A. Düz kablo yerine, çapraz bir kablo kullanılmalıdır.

    B. Düz kablo yerine, rollover bir kablo kullanılmalıdır.

    C. Subnet mask, 255.255.255.192 olarak ayarlanmalıdır.

    D. Her host'ta, bir varsayılan ağ ayarlanmalıdır.

    E. Subnet mask'lar, 255.255.255.0 olarak ayarlanmalıdır.

14. 14. Bir router'ın Ethernet port'una, 172.16.112.1/25 IP adresi tanımlanırsa, bu host'un geçerli subnet adresi ne olur?

    A. 172.16.112.0

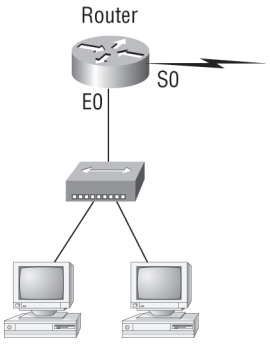
    B. 172.16.0.0

    C. 172.16.96.0

    D. 172.16.255.0

    E. 172.16.128.0

15. Aşağıdaki şekli kullanarak, sekiz subnet kullanıyorsanız, E0'ın IP adresi ne olur? Network ID, 192.168.10.0/28'dir ve aralıktaki son geçerli IP adresini kullanmalısınız. Bu soruda, zero subnet, geçerli kabul edilmeyecektir?




    A. 192.168.10.142

    B. 192.168.10.66

    C. 192.168.100.254

    D. 192.168.10.143

    E. 192.168.10.126

16. Bir önceki sorudaki şekli kullanarak, ilk subnet'i kullanıyorsanız, S0'ın IP adresi ne olur? Network ID, 192.168.10.0/28 dir ve aralıktaki son geçerli IP adresini kullanmalısınız. Bu soruda, zero subnet geçerli kabul edilmeyecektir?

    A. 192.168.10.24

    B. 192.168.10.62

    C. 192.168.10.30

    D. 192.168.10.127

17. Şayet KlasC subnet mask'ı, 255.255.255.224 ise, 8 subnet'in kullanımına izin vermek için hangi yapılandırma komutu kullanılmalıdır?

    A. Router(config)#ip classless

    B. Router(config)#ip version 6

    C. Router(config)#no ip classful

    D. Router(config)#ip unnumbered

    E. Router(config)#ip subnet-zero

18. 172.16.17.0/22 subnet'i olan bir ağa sahipsiniz. Geçerli host adresleri nelerdir?

    A. 172.16.17.1 255.255.255.252

    B. 172.16.0.1 255.255.240.0

    C. 172.16.20.1 255.255.254.0

    D. 172.16.16.1 255.255.255.240

    E. 172.16.18.255 255.255.252.0

    F. 172.16.0.1 255.255.255.0

19. Router'ınız, Ethernet0'ında şu IP adresine sahiptir: 172.16.2.1/23. Router'a bağlı LAN interface'inde aşağıdakilerden hangileri geçerli host ID'leri olabilir?

    A. 172.16.0.5

    B. 172.16.1.100

    C. 172.16.1.198

    D. 172.16.2.255

    E. 172.16.3.0

    F. 172.16.3.255

20. Lokal host'unuzdaki IP yığınını test etmek için, hangi IP adresini ping'lemelisiniz?

    A. 127.0.0.0

    B. 1.0.0.127

    C. 127.0.0.1

    D. 127.0.0.255

    E. 255.255.255.255

# Gözden Geçirme Sorularının Cevapları

1. D /27 (255.255.255.224), 3 bit kullanımda, 5 bit değil demektir. Bu, her biri 30 host'lu 8 subnet sağlar. Bu mask'ın bir KlasA, B veya C network adresleri ile kullanılmasının önemi var mıdır? Asla. Host bit sayısı değişmez.
2. D 240 mask, 4 subnet bit'idir ve her biri 14 host ile 16 subnet sağlar. Bizim daha fazla subnet'e ihtiyacımız var, bu nedenle subnet bit'i eklemeliyiz. Bir ilave subnet bit'i ile 248 mask olacaktır. Bu, 3 host bit'i (her subnet'e 6 host) ile 5 subnet bit'i (32 subnet) sağlar. Bu en iyi cevaptır.
3. C Bu oldukça basit bir sorudur. /28, 255.255.255.240'dır ve blok boyutumuzun, dördüncü oktet'te 16 olduğu anlamına gelir. 0, 16, 32, 48, 64, 80 vs. Host, 64 subnet'indedir.
4. F /19'un CIDR adresi, 25.255.224.0'dır. Bu bir klasB adresidir, bu nedenle, sadece 3 subnet bit'idir fakat o, 13 host bit'i veya her biri 8,190 host ile 8 subnet sağlar.
5. B,D Bir KlasA adresi ile kullanılan 255.255.254.0 (/23) mask'ı, 15 subnet bit'i ve 9 host bit'i olduğu anlamına gelir. Üçüncü oktetteki blok boyutu 2'dir (256-254). Bu nedenle, bu ilgili oktetteki subnet'leri, 0, 2, 4, 6 ...254 yapar. 10.16.3.65 host'u, 2.0 subnet'indedir. Bir sonraki subnet 14.0'dır, bu nedenle 2.0 subnet'i için broadcast adresi, 3.255'dir. Geçerli host adresleri, 2.1'den 3.254'e kadardır.
6. D /30, adresin klasına bakılmaksızın, dördüncü oktet'te 252'ye sahiptir. Bu, 4 blok boyutuna sahip olduğumuz ve subnet'lerimizin, 0, 4, 8, 12, 16 vs. olduğu anlamına gelir. 14 adresi, 12 subnet'indedir.
7. D Bir noktadan-noktaya link, sadece iki host kullanır. /30 veya 255.255.255.252 mask'ı, her subnet için iki host sağlar.
8. C /21, 255.255.248.0'dır, üçüncü oktette 8 blok boyutumuz var demektir, bu nedenle 66'ya ulaşana kadar 8'in katlarıyla sayarız. Bu sorudaki subnet, 64.0'dır. Sonraki subnet 72.0'dır, bu nedenle 64 subnet'inin broadcast adresi, 71,255'dir.
9. A /29 (255.255.255.248), adresin klasına bakılmaksızın, sadece 3 host bit'ine sahiptir. Altı host, bu LAN'daki maksimum host sayısıdır (router interface'inide içerir).
10. C /29, 255.255.255.248 'dir ve dördünü oktette 8 blok boyutundadır. Subnet'ler, 0, 8, 16, 24, 32, 40,vs. dir. 192.168.19.24, 24 subnet'tir ve 32, sonraki subnet olduğundan, 24 subnet için broadcast adresi, 31'dir. 192.168.19.26, tek doğru cevaptır.
11. A /29 (255.255.255.248), dördüncü oktet'te 8 blok boyutuna sahiptir. Bu, subnet'lerin 0, 8, 16, 24,vs. olduğu anlamına gelir. 10, 8 subnet'indedir. Sonraki subnet, 16'dır, bu nedenle 15, broadcast adresidir.
12. B Her biri en az 16 host ile 5 subnet'e ihtiyacınız var. 255.255.255.240 mask'ı, 14 host'la, 16 subnet sağlar. Bu çalışmayacaktır. 255.255.255.224, her biri 30 host'la 8 subnet sağlar. En iyi cevap budur.
13. A,E İlk olarak, şayet şekilde görüldüğü gibi, direkt bağlı iki host'a sahipseniz, bir çapraz kabloya ihtiyacınız vardır. Düz kablo çalışmayacaktır. İkincisi, host'lar, kendilerini farklı subnet'lere koyan, farklı mask'lara sahiptir. Kolay çözüm, her iki mask'ı da 255.255.255.0 (/24) olarak ayarlamaktır.
14. A /25, 255.255.255.128'dir. Bir KlasB ağı ile kullanıldığında, üçüncü ve dördüncü oktetler, 8 bit'i üçüncü, 1 bit'i dördüncü oktette, toplam 9 subnet bit'i ile subnet'leme için kullanılmaktadır. Dördüncü oktette sadece bir bit olduğundan bit, bir ya da sıfırdır (0 veya 128 değerindedir). Sorudaki host, 0 subnet'indedir. 128, bir sonraki subnet olduğundan, 127 broadcast adresine sahiptir.
15. A /28, 255.255.255.240 mask'ıdır. Gelin dokuzuncu subnet'e kadar sayalım (sekizinci subnet'in broadcast adresini bulmamız gerekmektedir, bu yüzden dokuzuncu subnet'e kadar sayıyoruz). 16'dan başlayarak (hatırlayın, soru, zero(0) subnet'ini kullanmayacağımızı

belirtmektedir, bu yüzden 0'dan değil, 16'dan başlıyoruz.) 16, 32, 48, 64, 80, 96, 112, 128, 144. Sekizinci subnet, 128'dir ve sonraki subnet, 144'tür. Böylece, 128 subnet'inin broadcast adresi, 143'tür. Bu, host aralığını, 129–142 yapar. 142, son geçerli host'tur.

16. C /28, 255.255.255.240 mask'ıdır. İlk subnet, 16'dır (hatırlayın, soru, zero(0) subnet'ini kullanmayacağımızı belirtmektedir) ve sonraki subnet, 32'dir, bu sebeple broadcast adresimiz, 31'dir. Bu, host aralığımızı, 17–30 yapar. 30, son geçerli hosttur.
17. E 255.255.255.224 KlasC subnet mask'ı, 3 bit'i 1 ve 5 bit'i 0'dır (11100000) ve her biri 30 host ile 8 subnet sağlamaktadır. Bununla birlikte, `ip subnet-zero` komutu kullanılmazsa, o zaman kullanım için sadece 6 subnet uygun olabilir.
18. E Bir /22 mask ile KlasB network ID'si, 255.255.252.0'dır, buda üçüncü oktet'te 4 blok boyutu demektir. Sorudaki subnet adresi, 172.16.19.255 broadcast adresi ile 172.16.16.0 subnet'indedir. Sadece E şıkkı, doğru subnet mask'ına sahiptir ve 172.16.18.22, geçerli bir host adresidir.
19. D,E E0 interfce'indeki router'un IP adresi 172.16.2.1/23'tür (255.255.254.0). Bu, üçüncü oktet'te blok boyutunu 2 yapar. Router'un interface'i, 2.0 subnet'indedir ve sonraki subnet, 4.0 olduğundan, broadcast adresi, 3.254'tür. Geçerli host aralığı, 2.1'den 3.254'e kadardır. Router, aralıktaki ilk geçerli host adresini kullanıyor.
20. C Host'unuzdaki lokal yığını test etmek için 127.0.0.1 loopback interface'ini ping'leyin.

# Yazılı Lab 3.1 Cevapları

1. 192.168.100.25/30. 30, 255.255.255.252'dir. Geçerli subnet, 192.168.100.24, broadcast, 192.168.100.27 ve geçerli host'lar, 192.168.100.25 ve 26'dır.
2. 192.168.100.37/28'dir. /28, 255.255.255.240'dır. Dördüncü oktet, 16 blok boyutundadır. 37'yi geçene kadar 16'nın katlarını sayın. 0, 16, 32, 48. Host, 47 broadcast adresi ile 32 subnet'indedir. Geçerli host'lar, 33-46'dır.
3. 192.168.100.66/27. /27, 255.255.255.224'tür. Dördüncü oktet, 32 blok boyutundadır. 66 host adresini geçene kadar, 32'nin katlarını sayın. 0, 32, 64, 96. Host, 64 subnet'indedir, broadcast adresi, 95'tir. Geçerli host aralığı 65-94'tür.
4. 192.168.100.17/29. /29, 255.255.255.248'dir. Dördüncü oktet, 8 blok boyutundadır. 0, 64, 128. Host, 64 subnet'indedir, broadcast 23'tür. Geçerli host'lar, 17-22'dir.
5. 192.168.100.99/26. /29, 255.255.255.192'dir. Dördüncü oktet, 64 blok boyutundadır. 0, 8, 16, 24. Host, 16 subnet'indedir, broadcast,127 dir. Geçerli host'lar, 65-126'dır.
6. 192.168.100.99/25. /29, 255.255.255.128'dir. Dördüncü oktet, 128 blok boyutundadır. 0, 128. Host, 0 subnet'indedir, broadcast 127'dir. Geçerli host'lar, 1-126'dır.
7. Varsayılan KlasB, 255.255.0.0'dır. Bir KlasB 255.255.255.0 mask'ı, her biri 254 host ile 256 subnet'tir. Bizim daha az subnet'e ihtiyacımız var. Şayet 255.255.240.0 kullanırsak, 16 subnet sağlar. Gelin bir subnet bit'i daha ekleyelim. 255.255.248.0. Bu, subnet'leme için 5 bit'tir ve 32 subnet sağlar. Bu, bizim en iyi cevabımızdır (/21).
8. /29, 255.255.255.248'dir. Bu, dördüncü oktet'te 8 blok boyutudur. 0, 8, 16. Host, 8 subnet'indedir, broadcast,15'tir.
9. /29, 5 subnet bit'i ve 3 host bit'i olan, 255.255.255.248'dir. Bu, her subnet için sadece 6 host'tur.
10. /23, 255.255.254.0'dır. Üçüncü oktet, 2 blok boyutundadır. 0, 2, 4. Subnet, 16.2.0 subnet'indedir, broadcast adresi, 16.3.255'tir.

# Yazılı Lab 3.2 Cevapları

| Classful Adres | Subnet Mask | Subnet başına Host sayısı ($2^h - 2$) |
|---|---|---|
| /16 | 255.255.0.0 | 65,534 |
| /17 | 255.255.128.0 | 32,766 |
| /18 | 255.255.192.0 | 16,382 |
| /19 | 255.255.224.0 | 8,190 |
| /20 | 255.255.240.0 | 4,094 |
| /21 | 255.255.248.0 | 2,046 |
| /22 | 255.255.252.0 | 1,022 |
| /23 | 255.255.254.0 | 510 |
| /24 | 255.255.255.0 | 254 |
| /25 | 255.255.255.128 | 126 |
| /26 | 255.255.255.192 | 62 |
| /27 | 255.255.255.224 | 30 |
| /28 | 255.255.255.240 | 14 |
| /29 | 255.255.255.248 | 6 |
| /30 | 255.255.255.252 | 2 |

# Yazılı Lab 3.3 Cevapları

| Ondalıklı IP Adresi | Adres Sınıfı | Subnet ve Host Bit sayısı | Subnet Sayısı (2x) | Host Sayısı (2x – 2) |
|---|---|---|---|---|
| 10.25.66.154/23 | A | 15/9 | 32768 | 510 |
| 172.31.254.12/24 | B | 8/8 | 256 | 254 |
| 192.168.20.123/28 | C | 4/4 | 16 | 14 |
| 63.24.89.21/18 | A | 10/14 | 1,024 | 16,384 |
| 128.1.1.254/20 | B | 4/12 | 16 | 4094 |
| 208.100.54.209/30 | C | 6/2 | 64 | 2 |

# 4 Cisco Internetworking Operating System (IOS) ve Security Device Manager (SDM)

# 4 Cisco Internetworking Operating System (IOS) ve Security Device Manager (SDM)

- IOS Kullanıcı Arayüzü
- Command-Line Interface (CLI)
- Router ve Switch Yönetimsel Konfigürasyonları
- Router Interface'leri
- Konfigürasyonlara Bakmak, Kaydetmek ve Silmek
- Cisco Security Device Manager (SDM)
- Özet
- Sınav Gereklilikleri
- Yazılı Lab 4
- Pratik Lab'lar
- Gözden Geçirme Soruları
- Gözden Geçirme Sorularının Cevapları
- Yazılı Lab 4 Cevapları

# Cisco Internetworking Operating System (IOS) ve Security Device Manager (SDM)

Sizi, Cisco Internetwork Operating System (IOS) ile tanıştırmanın zamanı geldi. IOS, hem Cisco router'ları ve switch'leri çalıştırır hem de cihazların konfigürasyonunu sağlar.

İşte bu bölümde öğreneceksiniz bunlardır. Size, Cisco IOS command-line interface (CLI) kullanarak, bir Cisco router'ı nasıl yapılandıracağınızı göstereceğim. Bu arayüzde uzmanlaştığınızda, hem hostname'leri, banner'ları, şifreleri v.s yapılandırabileceksiniz hem de Cisco IOS kullanarak hata tespiti yapabileceksiniz.

Buradan, Cisco'nun SDM'ine (Security Device Manager) bir göz atacağız ve aynı yapılandırmaları sağlaması için bir router'a, HTTPS oturumunu nasıl kuracağımızı öğreneceksiniz. SDM, bölümün ilerisinde gerçekten çok daha etkili bir araç olmaktadır. Çünkü access list'leri, VPN'leri ve kolayca IPSec yapılandırmayı mümkün kılar, fakat ilk olarak Cisco IOS'un temellerini öğrenmeniz gerekmektedir.

Ayrıca sizin, router yapılandırmaları ve komut doğrulamalarında hızlı olmanızı sağlayacağım. Aşağıda, bu bölümün içereceği konuların listesini bulabilirsiniz:

- Cisco IOS'u (Internetwork Operating System) anlamak ve yapılandırmak.
- Bir router'ı ayağa kaldırmak.
- Bir router'da oturum açmak.
- Router komut satırlarını anlamak.
- CLI komut satırını anlamak.
- Düzenleme (editing) ve yardım (help) özelliklerini çalıştırmak.
- Temel routing bilgilerini toplamak.
- Yönetimsel fonksiyonları ayarlamak.
- Hostname'leri ayarlamak.
- Banner'ları ayarlamak.
- Şifreleri ayarlamak.
- Interface açıklamalarını ayarlamak.
- Interface ayarlamalarını çalıştırmak.
- Konfigürasyonları gözden geçirmek, kaydetmek ve silmek.
- Routing ayarlarını doğrulamak.

Daha önceki modüllerde olduğu gibi, bu modülde öğreneceğiniz prensipler, kitapta sonraki modüllere geçmeden önce oturması gereken temel bölümlerdir.

*Bu bölümün son güncellemeleri için www.lammle.com ve/veya www.sybex.com adreslerine bakınız.*

NOT

## IOS Kullanıcı Arayüzü

Cisco Internetwork Operating System (IOS), Cisco router ve birçok switch'in çekirdeğidir. Bilmiyorsanız, bir kernel, düşük-seviye donanım arayüzleri ve güvenlik gibi yönetimsel yetileri ve kaynakları sağlayan, bir işletim sisteminin vazgeçilmez temel parçasıdır.

*Cisco switch konfigürasyonlarını, bölüm 8, "LAN Switching ve Spanning Tree Protocol(STP)" için saklayacağım.*

NOT

Sonraki bölümlerde, size, Cisco IOS ve command-line interface (CLI) kullanarak bir Cisco router'ın nasıl yapılandırılacağını göstereceğim. Bu bölümün sonuna doğru, Cisco SDM kullanacağız.

## Cisco Router IOS

Cisco IOS; routing, switching, ağlararası iletişim ve telekomünikasyon özellikleri sağlayan tescilli bir kernel'dır. İlk IOS, 1986'da William Yeager tarafından yazılmıştır ve network uygulamalarını mümkün kılmıştır. IOS, hem birçok Cisco router'da hem de Catalyst2950/2960 ve 3550/3560 serisi switch'ler gibi, çok sayıda Cisco Catalyst switch'te çalışmaktadır.

Aşağıdakiler, Cisco router IOS yazılımının sorumlu olduğu bazı özelliklerdir:

- Network protokol ve fonksiyonlarını taşımak
- Cihazlar arasındaki yüksek hızda trafiği bağlamak.
- Erişimi kontrol etmek için güvenlik sağlamak ve izinsiz network kullanımını engellemek.
- Ağın büyümesini ve kullanılabilirliğini kolaylaştırmak için ölçeklenebilirlik sağlamak.
- Network kaynaklarına bağlanmak için güvenliği sağlamak.

Cisco IOS'a, router'ın konsol port'undan, auxiliary (veya Aux) port'una bir modem bağlayarak ya da Telnet üzerinden erişebilirsiniz. IOS komut satırına girmek, bir EXEC oturumu olarak belirtilir.

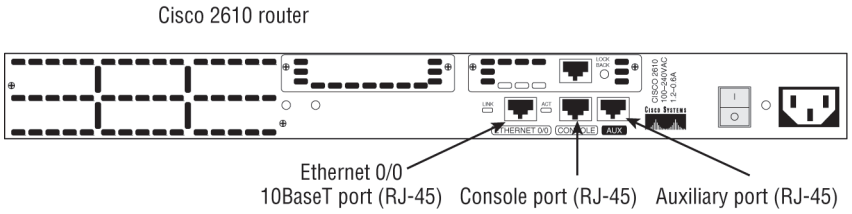
## Cisco Router'a Bağlanmak

Bir Cisco router'a, onu yapılandırmak, yapılandırmayı doğrulamak ve istatistikleri kontrol etmek için bağlanabilirsiniz. Bunu yapmanın farklı yolları vardır, fakat en yaygını ilk olarak ona konsol port'u ile bağlanmaktır. Konsol port'u genelde router'ın arkasında bulunan RJ-45 (8 pin modüler) bir bağlantıdır. Varsayılan olarak bir şifre ayarlanmış veya ayarlanmamış olabilir. Yeni ISR router'lar, varsayılan olarak kullanıcı adı ve şifre olarak Cisco kullanır.

Cisco router'a ayrıca bir auxiliary port yardımıyla da bağlanabilirsiniz. O aslında, bir konsol port'uyla aynı şeydir, bu nedenle onu da kullanabilirsiniz. Fakat bir auxiliary port, aynı zamanda modem komutlarını yapılandırmanıza da izin verir, böylece modem router'a bağlanabilir. Bu oldukça güzel bir özelliktir, şayet router down ise ve onu network dışından yapılandırmanız gerekiyorsa, uzak router'a dial-up yapmanıza ve auxiliary port'a bağlanmanıza izin verir.

**NOT** *Bir PC'nin, bir router konsol port'una nasıl bağlanacağının açıklaması için "Ağlar arası iletişim" başlıklı bölüm 1'e bakınız.*

Bir Cisco router'a bağlanmanın üçüncü yolu, ağdan Telnet programıdır. Telnet, aptal terminal olarak davranan bir terminal emülasyon programıdır. Telnet'i, Ethernet ya da seri port gibi router üzerindeki aktif bir interface'e bağlanmak için kullanabilirsiniz.

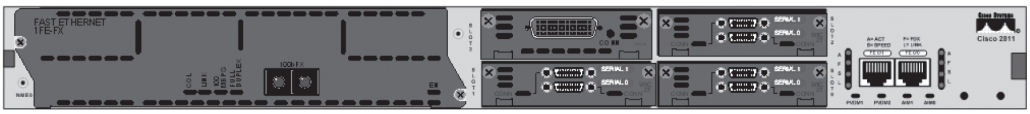



***Şekil 4.1:*** *Bir Cisco 2660 router.*

Şekil 4.1, bize Cisco 2600 serisi modüler router'ı göstermektedir. 2500 serisinden bir gömlek üstündür, çünkü daha hızlı bir işlemciye sahiptir ve daha fazla interface kullanabilir. 2500 ve 2600 serisi router'ların her ikisi de, miadını doldurmuştur. Siz onları sadece ikinci el satın alabilirsiniz. Bununla birlikte, birçok 2600 serisi router, hala kullanımdadır, bu nedenle onlardan anlamak çok önemlidir. Tüm farklı interface ve bağlantı türlerini dikkatle inceleyin.

2600 serisi router, birçok seri interface'e sahip olabilir. Seri V.35 WAN bağlantılarını, T1 veya Frame Relay bağlantısı için kullanılabilir. Modeline bağlı olarak çok sayıda Ethernet veya FastEt-

hernet port'u, router'da kullanılabilir. Bu router ayrıca RJ-45 konnektörleri yardımıyla bir konsol ve auxiliary bağlantısına sahiptir.

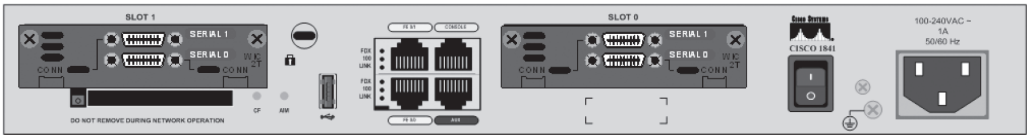


***Şekil 4.2:*** *Bir Cisco 2800 router.*

Bahsetmek istediğim diğer router 2800 serisidir (Şekil 4.2'de gösterilmektedir). Bu router, 2600 serisi router'ların yerini almaktadır ve bir Integrated Services Router (ISR) olarak belirtilir. İsmini, güvenlik gibi yerleşik bazı servislerden alır. 2600 gibi modüler bir cihazdır, fakat daha hızlı ve düzgündür. Daha geniş interface seçeneklerini desteklemesi için özenle tasarlanmıştır.

Güvenlik özelliğinin yerleşik olduğunu belirtmiştim. 2800, önceden yüklenmiş Device Manager'a (SDM) sahiptir. SDM, Cisco router'lar için Web-tabanlı cihaz-yönetim aracıdır. Bir web konsolu üzerinden router'ı yapılandırmanıza yardımcı olabilir. Çoğunu aklınızda tutmanız gerekir ki, birçok interface ekleyerek başlamazsanız, 2800 almakla oldukça karlı çıkarsınız. Bu küçük güzelliklerin her biri için para ödemek zorundasınız ve maliyetler gerçekten hızla artabilir.

2800 serisinden daha ucuz bazı router serileri vardır: 1800 ve 800 serileri. Şayet daha ucuz, 2800'e alternatif router bakıyorsanız ve hala aynı 12.4 IOS ile en güncel SDM'i çalıştırmak istiyorsanız, bu router'ları inceleyebilirsiniz.



***Şekil 4.3:*** *Bir Cisco 1841 router.*

Şekil 4.3, 2800 ile aynı interfaceleri bulunduran, fakat daha küçük ve ucuz olan 1841 router'u göstermektedir. Sizin 1800 serisi bir router yerine 2800 seçmenizin gerçek sebebi, 2800'de çalıştırabileceğiniz wireless denetleyicisi ve switching modülleri gibi daha gelişmiş interfaceleri olmasıdır.

Router konfigürasyon örneklerini göstermek için bu kitap boyunca, tüm yeni 2800, 1800 ve 800 serisi router'ları kullanacağım. Routing prensiplerini uygulamak için 2600, hatta 2500 router'ları bile kullanabileceğinizi kavramalısınız.

*Tüm Cisco router'lar hakkında www.cisco.com/en/US/products/hw/routers/index.html adresinde daha fazla bilgi bulabilirsiniz.*

NOT

## Bir Router'ı Çalıştırmak

Bir Cisco router'ı ilk açtığınızda power-on self-test (POST) çalıştırır. Şayet geçerse, flash bellekten Cisco IOS arar ve yükler (şayet bir IOS dosyası varsa). (Bu arada bilmiyorsanız flash bellek, elektronik olarak silinebilen, programlanabilir ve salt-okunur bir bellektir-EEPROM.) Bundan sonra, IOS yüklenir ve geçerli bir konfigürasyon (startup-config) arar. Bu kalıcı RAM (NVRAM)'de saklanır.

Aşağıdaki mesajlar, bir router'ı ilk boot ettiğinizde ya da reload ettiğinizde görünen mesajlardır (2822 router kullanıyorum):

```
System Bootstrap, Version 12.4(13r)T, RELEASE SOFTWARE (fc1)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 2006 by cisco Systems, Inc.
Initializing memory for ECC
c2811 platform with 262144 Kbytes of main memory
```

```
Main memory is configured to 64 bit mode with ECC enabled
Upgrade ROMMON initialized
program load complete, entry point: 0x8000f000, size: 0xcb80
program load complete, entry point: 0x8000f000, size: 0xcb80
```

Bu, router boot proses çıktısının ilk parçasıdır. İlk olarak POST çalıştığında, bootstrap programı hakkındaki bilgidir. Sonra router'a nasıl yükleme yapacağını söyler (varsayılanı, flash bellekteki IOS'u bulmaktır). Aynı zamanda, router'daki RAM boyutunu listeler.

Sonraki bölüm bize, RAM'e, sıkıştırılmış IOS'un açılmasını gösterir:

```
program load complete, entry point: 0x8000f000, size: 0x14b45f8
Self decompressing the image :
   ########################################################
######
   ########################################## [OK]
```

Pound işaretleri bize, IOS'un RAM'e yüklenmekte olduğunu belirtir. Sıkıştırılmış IOS RAM'e açıldıktan sonra, IOS, yüklenir ve aşağıda görüleceği gibi router çalışmaya başlar. IOS versiyonunun, gelişmiş güvenlik versiyonu 12.4.(12) olduğuna dikkat edin:

```
[some output cut]
Cisco IOS Software, 2800 Software (C2800NM-ADVSECURITYK9-M),
Version
     12.4(12), RELEASE SOFTWARE (fc1)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2006 by Cisco Systems, Inc.
Compiled Fri 17-Nov-06 12:02 by prod_rel_team
Image text-base: 0x40093160, data-base: 0x41AA0000
```

ISR router'ların yeni güzel özelliklerinden biri, IOS isminin şifreli olmamasıdır. Dosya adı aslında size, Advanced Security'de olduğu gibi IOS'un neler yapabileceğini söyler. IOS yüklenince, POST'tan öğrenilen bilgi görüntülenecektir. Aşağıda görebilirsiniz:

```
[some output cut]
Cisco 2811 (revision 49.46) with 249856K/12288K bytes of memory.
Processor board ID FTX1049A1AB
2 FastEthernet interfaces
4 Serial(sync/async) interfaces
1 Virtual Private Network (VPN) Module
DRAM configuration is 64 bits wide with parity enabled.
239K bytes of non-volatile configuration memory.
62720K bytes of ATA CompactFlash (Read/Write)
```

İki FastEthernet, dört seri interface ve bir VPN modül vardır. RAM'in boyutu, NVRAM ve flash, ayrıca görüntülenmektedir. Yukarıdaki router çıktısı bize, 256MB RAM, 239K NVRAM ve 64MB flash olduğunu göstermektedir.

NOT *1841 ve 871W router'lar, 2811 ile tamamen aynı şekilde boot olur. 1841 ve 871W, daha az hafıza ve farklı interfaceler sağlar, fakat bunun dışında aynı bootup prosedürü ve aynı start-up dosyasına sahiptir.*

IOS yüklenip, çalışmaya başladığında, bir önyapılandırma (startup-config denilir), NVRAM'den RAM'e kopyalanır. Bu dosyanın kopyası, RAM'e yerleştirilir ve running-config olarak belirtilir.

## ISR Olmayan Bir Router'ı Çalıştırmak (2600)

Gördüğünüz gibi boot işleyişi, ISR olmayan router'larda ISR router'larla neredeyse aynıdır. Bir 2600 router'ı ilk olarak boot ettiğinizde ya da yüklediğinizde, aşağıdaki mesajlar görünür:

```
System Bootstrap, Version 11.3(2)XA4, RELEASE SOFTWARE (fc1)
Copyright (c) 1999 by cisco Systems, Inc.
TAC:Home:SW:IOS:Specials for info
C2600 platform with 65536 Kbytes of main memory
```

Sonraki bölüm bize, sıkıştırılmış IOS'un RAM'e açılmasını göstermektedir:

```
program load complete, entry point:0x80008000, size:0x43b7fc
Self decompressing the image :
################################################################
######
################################################################
######
################################################################
######
################################################################
######
################################################################
######
################################################################
######
## [OK]
```

Şimdiye kadar herşey nerdeyse aynı. Aşağıda, IOS versiyonunun, 12.3(20) olarak göründüğüne dikkat edin:

```
Cisco Internetwork Operating System Software
IOS (tm) C2600 Software (C2600-IK9O3S3-M), Version 12.3(20),
RELEASE
   SOFTWARE (fc2)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2006 by cisco Systems, Inc.
Compiled Tue 08-Aug-06 20:50 by kesnyder
Image text-base: 0x80008098, data-base: 0x81A0E7A8
```

2800 serisinde olduğu gibi, IOS yüklendiğinde POST'tan öğrenilen bilgi görüntülenecektir:

```
cisco 2610 (MPC860) processor (revision 0x202) with 61440K/4096K
bytes
   of memory.
Processor board ID JAD03348593 (1529298102)
M860 processor: part number 0, mask 49
Bridging software.
X.25 software, Version 3.0.0.
1 Ethernet/IEEE 802.3 interface(s)
```

```
1 Serial network interface(s)
2 Serial(sync/async) network interface(s)
32K bytes of non-volatile configuration memory.
16384K bytes of processor board System flash (Read/Write)
```

Son olarak, burada bir Ethernet ve üç seri interface görürüz. RAM boyutu ve flash ayrıca görüntülenmektedir ve yukarıdaki router çıktısı, 64MB RAM ve 16MB flash olduğunu göstermektedir.

Ve belirttiğim gibi, IOS yüklenip, up olduğunda ve çalıştığında, startup-config olarak bilinen geçerli bir konfigürasyon, NVRAM'den yüklenmektedir. Fakat ISR router'ın varsayılan bootup'ından farklılık buradadır. Şayet NVRAM'de bir yapılandırma ayarı yoksa router, TFTP host'unda geçerli bir konfigürasyon bulmak için broadcast gönderir. (Bu sadece, router, herhangi bir interface'inde, CD (carrier detect) algılarsa olur.) Şayet broadcast başarısız olursa, setup-mode olarak bilinen bir duruma geçer. Bu router'ı, adım adım yapılandırmanıza yardımcı olan bir işlemdir. Şayet router'ınızın herhangi bir interface'ini ağınıza bağlarsanız ve router'ınızı boot ederseniz, router konfigürasyonu ararken, birkaç dakika beklemek zorunda kalabileceğinizi hatırlamanız gerekir.

NOT *ISR router'larınızında bu boot işleyişine, startup-config'i silerek ve router'ı reload ederek sahip olabilirsiniz. Bu size, varsayılan bir konfigürasyon olmaksızın, temiz bir router sağlar. Bunun nasıl yapıldığını size, bu bölümün ilerleyen kısımlarında göstereceğim.*

Aynı zamanda setup moduna, privileged olarak bilinen moddan, `setup` komutunu yazarak komut satırından girebilirsiniz. Setup modu, bazı genel komutları içerir ve genelde çok kullanışlı değildir. İşte bir örnek:

```
Would you like to enter the initial configuration dialog? [yes/
no]: y

At any point you may enter a question mark '?' for help.
Use ctrl-c to abort configuration dialog at any prompt.
Default settings are in square brackets '[]'.

Basic management setup configures only enough connectivity
for management of the system, extended setup will ask you
to configure each interface on the system

Would you like to enter basic management setup? [yes/no]: y
Configuring global parameters:

  Enter host name [Router]:Ctrl+C
Configuration aborted, no changes made.
```

NOT *Setup moddan, Ctrl+C'ye basarak istediğiniz zaman çıkabilirsiniz.*

Ben, ısrarla, setup moduna bir defa girmenizi ve bir daha uğramamanızı öneririm. Daima CLI ya da SDM kullanmalısınız.

# Command-Line Interface (CLI)

Bazen CLI'yı "Cash Line Interface" olarak belirtirim. Çünkü CLI kullanarak Cisco router ya da switch'lerde gelişmiş konfigürasyonlar oluşturabilirseniz, yani cash (nakit) kazanırsınız!

CLI kullanmak için router boot etmeyi bitirince Enter tuşuna basın. Bunu yaptıktan sonra, router, interface'lerinin her birinin durumu hakkında tüm bilgiyi veren mesajlar gönderir ve sonra bir banner gelerek size login olup olmayacağınızı sorar. Örneğe bakalım:

```
[some output cut]
*Feb 28 16:42:OO.967: %VPN_HW-6-INFO_LOC: Crypto engine: onboard
O
   State changed to: Initialized
*Feb 28 16:42:OO.971: %VPN_HW-6-INFO_LOC: Crypto engine: onboard
O
   State changed to: Enabled
*Feb 28 16:42:O1.471: %LINK-3-UPDOWN: Interface FastEthernetO/O,
   changed state to up
*Feb 28 16:42:O1.471: %LINK-3-UPDOWN: Interface FastEthernetO/1,
   changed state to up
*Feb 28 16:42:O1.471: %LINK-3-UPDOWN: Interface SerialO/O/O,
changed
   state to down
*Feb 28 16:42:O1.471: %LINK-3-UPDOWN: Interface SerialO/O/1,
changed
   state to down
*Feb 28 16:42:O1.471: %LINK-3-UPDOWN: Interface SerialO/1/O,
changed
   state to down
*Feb 28 16:42:O1.471: %LINK-3-UPDOWN: Interface SerialO/2/O,
changed
   state to down
[some output cut]
————————————————————
Cisco Router and Security Device Manager (SDM) is installed
on this device. This feature requires the one-time use of the
username
      “cisco” with the password “cisco”. The default username and
password
   have a privilege level of 15.
Please change these publicly known initial credentials using SDM
or the
   IOS CLI. Here are the Cisco IOS commands.
username <myuser>  privilege 15 secret O <mypassword>
no username cisco
Replace <myuser> and <mypassword> with the username and password
you
   want to use.
For more information about SDM please follow the instructions in
the
  QUICK START GUIDE for your router or go to http://www.cisco.
com/go/sdm
————————————————————
User Access Verification
Username: cisco
Password: cisco [this won’t show on your screen]
yourname#
```

Buradan, kullanıcı adı/şifre olarak, cisco/cisco kullanarak oturum açın ve artık priviledge moddasınız (bundan daha sonra bahsedeceğim).

Router'da zaten bir konfigürasyon olmasının sebebi, router'ı yapılandırmak zorunda olmaksızın, HTTPS üzerinden SDM ile bağlanabilmenizdir. Tekrar etmem gerekirse, önceden yapılandırılmış startup-config'ten, bu bölümde yakında bahsedeceğim.

## Bir ISR Olmayan Router'dan CLI Girişi

Interface durum mesajları göründükten ve Enter'a bastıktan sonra `Router>` komut satırı görünecektir. Bu, user exec mode (user mod) olarak belirtilir ve yaygın olarak istatistiklere bakmak için kullanılır, fakat aynı zamanda privileged moda bağlanmak için bir basamaktır.

Sadece, `enable` komutu ile girebileceğiniz, privileged exec modda (privileged mode) sadece, bir Cisco router konfigürasyonuna bakabilir ve değiştirebilirsiniz.

Şöyle yapabilirsiniz:

```
Router>enable
Router#
```

Şimdi `Router#` komut satırına ulaşabilirsiniz. Bu, router konfigürasyonunu bakıp değiştirebileceğiniz privileged modda bulunduğunuzu belirtir. Aşağıda görüldüğü gibi `disable` komutunu kullanarak, tekrar user moda dönebilirsiniz:

```
Router#disable
Router>
```

Bu noktada, herhangi bir moddan, konsola çıkmak için `logout` yazabilirsiniz:

```
Router>logout


Router con0 is now available
Press RETURN to get started.
```

**NOT** *Bir ISR router'ın varsayılan konfigürasyonunu silip, reload ederek, aynı komut satırlarına ulaşacağınızı ve bir kullanıcı adı ve şifre istemi gelmeyeceğini hatırlayın.*

Sonraki bölümlerde, bazı temel yönetimsel konfigürasyonların nasıl yapıldığını göstereceğim.

## Router Modlarını Gözden Geçirmek

CLI'dan konfigürasyon için `configure terminal` (ya da kısaca `config t`) yazarak router'da genel değişiklikler yapabilirsiniz. Bu sizi, global configuration moda götürür ve running-config olarak bilinen ayarların değiştirilmesini sağlar. Bir global komut (global config'den çalışan bir komut), sadece bir defa ayarlanır ve router'ın tamamını etkiler.

Privileged-mode komut satırından `config` yazabilir ve sonra `terminal`'in varsayılanına ulaşmak için sadece Enter tuşuna basabilirsiniz. Aşağıda göründüğü gibi:

```
yourname#config
Configuring from terminal, memory, or network [terminal]? [press
enter]
Enter configuration commands, one per line.  End with CNTL/Z.
yourname(config)#
```

Bu noktada, yaptığınız değişiklikler router'ın tamamını etkiler, bu nedenle global configuration mode terimi kullanılır. Dinamik RAM (DRAM)'de çalışan mevcut konfigürasyon olan running-config'i değiştirmek için, aynı gösterdiğim gibi, `configure terminal` komutunu kullanın.

NVRAM'de saklanan konfigürasyon olan startup-config'i değiştirmek için `configure memory` (ya da kısaca `config mem`) komutunu kullanın. Bu, startup-config dosyasını, RAM'deki running-

config dosyası ile birleştirir. Şayet, bir TFTP host'unda saklanan router konfigürasyonunu değiştirmek isterseniz (Modül5,"Cisco IOS yönetimi"nde işlenecektir) , `configure network` (ya da kısaca `config net`) komutunu kullanın. Bu ayrıca, RAM'deki running-config dosyası ile dosyayı birleştirir.

`Configure terminal`, `configure memory` ve `configure network` komutlarının hepsi, bir router'ın RAM'indeki bilgileri yapılandırmak için kullanılmaktadır. Bununla birlikte, genelde sadece `configure terminal` komutu kullanılmaktadır. Yine de, şayet running-config dosyanızı berbat ettiyseniz ve router'ınızı tekrar çalıştırmak istemiyorsanız, `config mem` ve `config net` komutları kullanışlı olabilecektir.

Aşağıda, configure komutu altındaki diğer seçeneklerden bazılarını bulabilirsiniz:

```
yourname(config)#exit veya  cntl-z 'e bas
yourname#config ?
  confirm                   Yeni bir config dosyası ile running-
config yerine geçmesini onaylar
  memory                  NV bellekten yapılandırmak için.
  network                  Bir TFTP network host'undan
yapılandırma
  overwrite-network  TFTP network host'undan NV belleğin üzerine
yazma
  replace                  Running-configure 'ü yeni bir config
dosyası ile değiştirme
  terminal                Terminalden yapılandırma
  <cr>
```

Görebileceğiniz gibi, Cisco, 12.4 IOS 'da ilave bazı komutlara sahiptir. Bu komutlara, bölüm 5'te bakacağız.

## CLI İstemcileri

Bir router'ı yapılandırırken bulabileceğiniz farklı komut satırlarını anlamak gerçekten önemlidir. Bunları iyi bilmek, configuration modda herhangi bir zamanda nerede olduğunuza bakmak ve farkına varmak için size yardımcı olacaktır. Bu bölümde, bir Cisco router'da kullanılan komut satırlarını ve kullanılan bazı terimleri göstereceğim. (bir router konfigürasyonunda değişiklik yapmadan önce, daima komut istemcinizi kontrol edin!)

Sunulan her farklı komut istemcisine girmeyeceğim, çünkü bunu yapmak, bu kitabın kapsamını aşacaktır. Yerine, bu bölüm ve kitabın kalanı boyunca, göreceğiniz tüm farklı komut istemcilerini açıklayacağım. Bu komut istemcileri esasen, gerçek hayatta en sık kullanacaklarınızdır ve sınav için bilmeye ihtiyacınız olanlardır.

NOT

*Sakın korkmayın! Bu komut istemcilerinin her birinin neyi başardığını bilmeniz önemli değil. Çünkü en kısa sürede tamamıyla bu bilgileri alacaksınız. Bu nedenle şimdi rahatlayın, farklı komut istemcilerini öğrenmeye odaklanın!*

### Interface'ler

Bir interface'de değişiklik yapmak için global configuration moddan `interface` komutunu kullanırsınız:

```
Yourname(config)#interface ?
  Async              Async interface
  BVI                Bridge-Group Virtual Interface
  CDMA-Ix            CDMA Ix interface
  CTunnel            CTunnel interface
```

```
  Dialer              Dialer interface
  FastEthernet        FastEthernet IEEE 802.3
  Group-Async         Async Group interface
  Lex                 Lex interface
  Loopback            Loopback interface
  MFR                 Multilink Frame Relay bundle interface
  Multilink           Multilink-group interface
  Null                Null interface
  Port-channel        Ethernet Channel of interfaces
  Serial              Serial
  Tunnel              Tunnel interface
  Vif                 PGM Multicast Host interface
  Virtual-PPP         Virtual PPP interface
  Virtual-Template    Virtual Template interface
  Virtual-TokenRing   Virtual TokenRing
  range               interface range command
Yourname(config)#interface fastEthernet 0/0
Yourname(config-if)#
```

Komut satırınızın, `yourname(config-if)#` olarak değiştiğini fark ettiniz mi? Bu, interface configuration modda olduğunuzu belirtir. Şayet komut satırı size ayrıca hangi interface'i yapılandırdığınız bilgisini verseydi iyi olmaz mıydı? En azından, bu bilgi olmaksızın yaşamayı bilmeliyiz, çünkü bu özellik yok. Emin olduğumuz bir şey var ki, bir router'ı yapılandırırken dikkatli olmak zorundasınız!

## Subinterface'ler

Subinterface'ler, router'da mantıksal interface'ler oluşturmanızı sağlar. Komut satırı, `yourname(config-subif)#` olarak değişir:

NOT *Subinterface'ler hakkında daha fazla bilgiyi, bölüm 9 "Virtual LAN'lar" ve bölüm 14 "Wide Area Network'ler"de okuyabilirsiniz, fakat henüz oralara geçmeyin!*

```
yourname(config-if)#interface
f0/0.1
yourname(config-subif)#
```

## Line Komutları

user-mode şifrelerini ayarlamak için `line` komutunu kullanın.

Sonra, komut satırı `yourname(config-line)#` olur:

```
yourname#config t
Enter configuration commands, one per line.  End with CNTL/Z.
yourname(config)#line ?
  <0-337>  First Line number
  aux      Auxiliary line
  console  Primary terminal line
  tty      Terminal controller
  vty      Virtual terminal
  x/y      Slot/Port for Modems
  x/y/z    Slot/Subslot/Port for Modems
```

`line console 0` komutu, ana bir komut olarak bilinir (ayrıca global command olarak da bilinir) ve (`config-line`) satırından yazılan herhangi bir komut, alt komut olarak bilinir.

## Routing Protokol Konfigürasyonları

RIP ve EIGRP gibi Roting protokollerini yapılandırmak için,

`yourname(config-router)#` komut satırını kullanın:

```
yourname#config t
Enter configuration commands, one per line.  End with CNTL/Z.
yourname(config)#router rip
yourname(config-router)#version 2
yourname(config-router)#
```

## Router Terimlerini Açıklamak

Tablo 4.1, şimdiye kadar kullandığımız bazı terimleri açıklamaktadır.

**Tablo 4.1:** Router Terimleri

| Mod | Açıklama |
|---|---|
| User EXEC mod | Temel görüntüleme komutlarıyla sınırlıdır |
| Privileged EXEC mod | Diğer tüm router komutlarına erişim sağlar |
| Global configuration modu | Tüm sistemi etkileyen komutlar |
| Spesifik configuration modları | Sadece interfaceleri / prosesleri etkileyen komutlar |
| Setup mod | İnteraktif konfigürasyon diyalogu |

## Editing ve Help Özellikleri

Cisco'nun gelişmiş düzenleme özelliklerini, router'ınızı yapılandırmada size yardımcı olması için kullanabilirsiniz. Herhangi bir komut istemcisinde soru işareti (?) kullanırsanız, bu satırda kullanabileceğiniz tüm komutlar listelenecektir:

```
yourname#?
Exec commands:
  access-enable    Create a temporary Access-List entry
  access-profile   Apply user-profile to interface
  access-template  Create a temporary Access-List entry
  archive          manage archive files
  auto             Exec level Automation
  bfe              For manual emergency modes setting
  calendar         Manage the hardware calendar
  cd               Change current directory
  clear            Reset functions
  clock            Manage the system clock
  cns              CNS agents
  configure        Enter configuration mode
  connect          Open a terminal connection
  copy             Copy from one file to another
  crypto           Encryption related commands.
  ct-isdn          Run an ISDN component test command
  debug            Debugging functions (see also 'undebug')
```

```
  delete           Delete a file
  dir              List files on a filesystem
  disable          Turn off privileged commands
  disconnect       Disconnect an existing network connection
 —More—
```

Artı bu noktada, spacebar'a basarak diğer bilgilendirme sayfasına geçebilir veya Enter tuşuna basarak her seferinde bir komut ilerleyebilirsiniz. Ayrıca, çıkmak için Q'ya (ya da bu amaçla başka bir tuşa) basabilir ve komut satırına dönebilirsiniz.

İşte bir kısa yol: Belirli bir harfle başlayan komutları bulmak için harfi ve aralarında boşluk olmadan soru işaretini kullanın:

```
yourname#c?
calendar  cd         clear    clock
cns       configure  connect  copy
crypto    ct-isdn

yourname#c
```

c? yazarak, c ile başlayan tüm komutların listelendiği bir çıktı aldık. Ayrıca, `yourname#c` istemcisinin, komutların listesi görüntülendikten sonra tekrar geldiğine dikkat edin. Bu, çok uzun komutlar listelendiği ve sonraki olası komuta ihtiyacınız olduğunda, faydalı olabilir. Soru işareti kullandığınız her sefer tüm komutu tekrar yazmak zorunda olsaydınız, bu tuhaf olurdu!

Satırdaki, bir sonraki komutu bulmak için ilk komutu yazın ve sonra soru işareti koyun:

```
yourname#clock ?
  read-calendar    Read the hardware calendar into the clock
  set              Set the time and date
  update-calendar  Update the hardware calendar from the clock
yourname#clock set ?
  hh:mm:ss  Current Time
yourname#clock set 11:15:11 ?
  <1-31>  Day of the month
  MONTH   Month of the year
yourname#clock set 11:15:11 25 aug ?
  <1993-2035>  Year
yourname#clock set 11:15:11 25 aug 2007 ?
  <cr>
yourname#clock set 11:15:11 25 aug 2007
*Aug 25 11:15:11.000: %SYS-6-CLOCKUPDATE: System clock has
been updated from 18:52:53 UTC Wed Feb 28 2007 to 11:15:11
UTC Sat Aug 25 2007, configured from console by cisco on console.
```

clock? komutunu yazarak, bir sonraki geçerli parametreleri ve onların ne yaptıklarının bir listesine ulaşabilirsiniz. `<cr>` (carriage return), tek seçeneğiniz oluncaya kadar bir komut yazıp, boşluk bırakıp, soru işareti kullanmaya devam edeceğinize dikkat edin.

Şayet komutları yazıp şunu alıyorsanız:

```
yourname#clock set 11:15:11
% Incomplete command.
```

Komut serisinin tamamlanmadığını bilirsiniz. Sadece, yukarı ok tuşuna basarak, girilen son komutu tekrar görüntüleyebilirsiniz ve sonra soru işaretini kullanarak komuta devam edebilirsiniz.

Şayet hata alırsanız;

```
yourname(config)#access-list 110 permit host 1.1.1.1
                                          ^
% Invalid input detected at '^' marker.
```

Bir komutu yanlış girmişsiniz. Küçük düzeltme işaretini (^) gördünüz mü? O, hata yaptığınız ve komutu hatalı girdiğiniz kesin noktayı gösteren çok faydalı bir araçtır. Aşağıda bu küçük düzeltme işaretini görebileceğiniz başka bir örnek vardır:

```
yourname#sh serial 0/0/0
              ^
% Invalid input detected at '^' marker.
```

Bu komut doğru görünmektedir, fakat dikkat edin! Problem komutun tam olarak `show interface serial 0/0/0` şeklinde olmamasıdır.

Şayet şöyle bir hata alırsanız:

```
yourname#sh ru
% Ambiguous command:  "sh ru"
```

Bunun manası, sizin girdiğiniz seri ile başlayan ve eşsiz olmayan çok sayıda komutun olmasıdır. İhtiyacınız olan komutu bulmak için soru işaretini kullanın:

```
yourname#sh ru?
rudpv1  running-config
```

Görebileceğiniz gibi, `show ru` ile başlayan iki komut vardır.

Tablo 4.2, bir Cisco router'da olan geliştirilmiş düzenleme komutlarını göstermektedir.

**Tablo 4.2:** Geliştirilmiş Düzenleme Komutları

| Komut | Anlamı |
|---|---|
| Ctrl+A | İmlecinizi satır başına götürür |
| Ctrl+E | İmlecinizi satırın sonuna götürür |
| Esc+B | Bir kelime geriye gider |
| Ctrl+B | Bir karakter geriye gider |
| Ctrl+F | Bir karakter ileriye gider |
| Esc+F | Bir kelime ileriye gider |
| Ctrl+D | Tek bir karakteri siler |
| Backspace | Tek bir karakteri siler |
| Ctrl+R | Bir satırı tekrar görüntüler |
| Ctrl+U | Bir satırı siler |
| Ctrl+W | Bir kelimeyi siler |
| Ctrl+Z | Konfigürasyon modunu sonlandırır ve EXEC'e geri döner |
| Tab | Sizin için bir komut yazmayı sonlandırır. |

Size göstermek istediğim diğer güzel editing özelliği, uzun satırların otomatik akışıdır. Aşağıdaki örnekte yazılan komut, sağ satır boşluğuna ulaşmıştır ve otomatik olarak sola doğru 11 boşluk gitmiştir (dolar işareti [$], satırın sola kaydırıldığını belirtir):

```
yourname#config t
Enter configuration commands, one per line. End with CNTL/Z.
yourname(config)#$110 permit host 171.10.10.10 0.0.0.0 eq 23
```

Router-komut geçmişini, Tablo 4.3'te gösterilen komutlarla gözden geçirebilirsiniz.

**Tablo 4.3:** Router-Komut Geçmişi

| Komut | Anlamı |
|---|---|
| Ctrl+P veya up arrow | Girilen son komutu gösterir |
| Ctrl+N veya down arrow | Girilen bir önceki komutu gösterir |
| `show history` | Varsayılan olarak, girilen son 10 komutu gösterir |
| `show terminal` | Terminal konfigürasyonlarını ve history arabellek boyutunu gösterir |
| `terminal history size` | Arabellek boyutunu değiştirir (max 256) |

Aşağıdaki örnek, hem `show history` komutunu, history boyutunu nasıl değiştirileceğini hem de `show terminal` komutu ile nasıl doğrulanacağını göstermektedir. İlk olarak, router'a girilen son 20 komutu görmek için `show history` komutunu kullanın:

```
yourname#show history
 en
 sh history
 show terminal
 sh cdp neig
 sh ver
 sh flash
 sh int fa0
 sh history
 sh int s0/0
 sh int s0/1
```

Şimdi de, terminal history boyutunu doğrulamak için `show terminal` komutunu kullanın

```
yourname#show terminal
Line 0, Location: ““, Type: ““
[output cut]
Modem type is unknown.
Session limit is not set.
Time since activation: 00:21:41
Editing is enabled.
History is enabled, history size is 20.
DNS resolution in show commands is enabled
Full user help is disabled
Allowed input transports are none.
Allowed output transports are pad telnet rlogin lapb-ta mop
v120 ssh.
```

```
  Preferred transport is telnet.
  No output characters are padded
  No special data dispatching characters
```

Privileged modda kullanılan, `terminal history` komutu, history arabellek boyutunu değiştirebilir:

```
yourname#terminal history size ?
 <0-256> Size of history buffer
yourname#terminal history size 25

show terminal komutu ile değişikliği doğrulayabilirsiniz:
yourname#show terminal
Line 0, Location: ““, Type: ““
[output cut]
Editing is enabled.
History is enabled, history size is 25.
Full user help is disabled
Allowed transports are lat pad v120 telnet mop rlogin
  nasi. Preferred is lat.
No output characters are padded
No special data dispatching characters
Group codes:  0
```

### Cisco Düzenleme Özelliklerini Ne Zaman Kullanırsınız?

*Editing özelliklerinin birkaçı sıkça kullanılmakta, bazıları daha az kullanılmaktadır,hatta kullanılmayabilirde. Bunları Cisco'nun düzenlemediğini bilin, tamamıyla eski Unix komutlarıdır. Bununla beraber, Ctr+A, bir komutu olumsuz yapmak için gerçekten faydalıdır.*

*Örneğin, uzun bir komut yazdığınızda ve sonra konfigürasyonunuzda bu komutu kullanmak istemediğinize karar verdiğinizde veya komut çalışmadığında, girilen son komutu göstermesi için, sadece üst ok tuşuna basabilirsiniz. Ctrl+A tuş bileşimine basın, no yazın ve bir boşluk bırakıp Enter tuşuna basın. Komut iptal edilmiştir. Bu, her komutta çalışmaz, fakat birçoğunda çalışır.*

## Temel Routing Bilgilerini Toplamak

`show version` komutu, hem sistem donanımı hem de yazılım versiyonu ve boot imajları için temel bilgi sağlar. İşte bir örnek:

```
yourname#show version
Cisco IOS Software, 2800 Software (C2800NM-ADVSECURITYK9-M),
Version
   12.4(12), RELEASE SOFTWARE (fc1)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2006 by Cisco Systems, Inc.
Compiled Fri 17-Nov-06 12:02 by prod_rel_team
```

Çıktının yukarıdaki bölümü, router'da çalışan Cisco IOS'u açıklar. Aşağıdaki bölüm, kullanılan read-only memory (ROM)'u tanımlar. ROM, router'ı boot etmek ve POST'u tutması için kullanılmaktadır:

```
ROM: System Bootstrap, Version 12.4(13r)T, RELEASE SOFTWARE (fc1)
```

Sonraki bölüm, router'ın ne kadar zamandır çalıştığını, nasıl başlatıldığını (şayet bir system restarted by bus hatası görürseniz, bu oldukça kötü bir şeydir), Cisco IOS'un hangi lokasyondan yüklendiğini ve IOS'un ismini gösterir. Flash varsayılandır:

```
yourname uptime is 2 hours, 30 minutes
System returned to ROM by power-on
System restarted at 09:04:07 UTC Sat Aug 25 2007
System image file is "flash:c2800nm-advsecurityk9-mz.124-12.bin"
```

Sonraki bölüm, işlemciyi, DRAM ve flash bellek boyutunu ve router'da bulunan POST interface'leri görüntüler.

```
[some output cut]
Cisco 2811 (revision 53.50) with 249856K/12288K bytes of memory.
Processor board ID FTX1049A1AB
2 FastEthernet interfaces
4 Serial(sync/async) interfaces
1 Virtual Private Network (VPN) Module
DRAM configuration is 64 bits wide with parity enabled.
239K bytes of non-volatile configuration memory.
62720K bytes of ATA CompactFlash (Read/Write)
Configuration register is 0x2102
```

Son olarak, configuration register değeri listelenmiştir (Modül5 'te bahsedeceğim).

İlave olarak, show interfaces ve `ip interface brief` komutları, hem bir router'ı doğrulama ve hata tespitinde hem de network hatalarında çok kullanışlıdır. Bu komutlar, bu bölümde daha sonra ele alınacaktır. Sakın kaçırmayın!

# Router ve Switch Yönetimsel Konfigürasyonları

Bu bölüm, bir ağda çalışan router ve switch'i çalıştırmak açısından çok kritik olmasa da, gerçekten önemlidir. Ağınızı yönetmede size yardımcı olacak konfigürasyon komutlarını gözden geçireceğim.

Bir router ya da switch'te konfigürasyon yapabileceğiniz yönetimsel fonksiyonlar şunlardır:

- Hostname'ler
- Banner 'lar
- Şifreler
- Interface açıklamaları

Hatırlayın, bunların hiçbiri router ya da switch'inizin daha iyi ya da hızlı çalışmasını sağlamaz, fakat bana inanın, şayet network cihazlarının her birinde bu konfigürayonları ayarlamak için azda olsa bir zaman harcarsanız, işleriniz daha kolaylaşacaktır. Bu nedenle bunu yapmak, hata tespitini ve ağınızı devam ettirmenizi cidden çok daha kolaylaşacaktır. Sonraki bölümde, bir Cisco router'daki komutları göstereceğim, fakat bu komutlar, tamamıyla bir Cisco switch'tekiyle aynıdır.

## Hostname'ler

Router'ın isimlendirilmesini, `hostname` komutuyla ayarlayabilirsiniz. Bu, tamamen lokal olarak önemlidir. Yani onun, router'un nasıl isim araması kullanacağı ya da ağ topluluğunda router'ın nasıl çalışacağıyla ilgisi yoktur. Bununla birlikte bölüm 14'te, PPP'den bahsederken, kimlik doğrulama amacıyla hostname'i kullanacağım.

İşte bir örnek:

```
yourname#config t
Enter configuration commands, one per line. End with
  CNTL/Z.
yourname(config)#hostname Todd
Todd(config)#hostname Atlanta
Atlanta(config)#hostname Todd
Todd(config)#
```

Hostname'i kendi adınızla yapılandırmak ne kadar cazip gelse de, router'a lokasyona uygun bir isim vermek kesinlikle çok daha iyi bir fikirdir. Bu sebeple, cihazın gerçekte nerede bulunduğuyla ilgili bir hostname vermek, onun çok kolay bulunmasını sağlayacaktır. Ayrıca, sizin doğru cihazı yapılandırdığınızı doğrulamanıza da yardımcı olacaktır. Bu modül için şimdilik onu, Todd olarak bırakacağız.

## Banner'lar

Bir banner, küçük, hoş bir araç olmasından çok daha fazla şey ifade eder. Bir banner'a sahip olmanın iyi bir nedeni, ağ topluluğunuza modemle ya da telnet ile bağlanmak isteyenlere küçük bir güvenlik uyarısı göndermektir. Router'da görünen herhangi bir kullanıcıya, sahip olmasını istediğiniz kesin bilgileri iletmek için, bir banner oluşturabilirsiniz.

Şu dört banner tipine aşina olduğunuza emin olun: exec process creation banner, incoming terminal line banner, login banner ve message of the day banner (hepsi aşağıdaki kodda gösterilmektedir):

```
Todd(config)#banner ?
  LINE            c banner-text c, where 'c' is a delimiting
character
  exec            Set EXEC process creation banner
  incoming        Set incoming terminal line banner
  login           Set login banner
  motd            Set Message of the Day banner
  prompt-timeout  Set Message for login authentication timeout
  slip-ppp        Set Message for SLIP/PPP
```

Message of the day (MOTD), en yaygın kullanılan banner'dır. O, router'a Telnet ya da bir auxiliary port ya da konsol port'u yardımıyla bağlanan herkese bir mesaj iletir:

```
Todd(config)#banner motd ?
LINE c banner-text c, where 'c' is a delimiting character
Todd(config)#banner motd #
Enter TEXT message. End with the character '#'.
$ Acme.com network, then you must disconnect immediately.
#
Todd(config)#^Z
Todd#
00:25:12: %SYS-5-CONFIG_I: Configured from console by
  console
Todd#exit
```

```
Router con0 is now available

Press RETURN to get started.

If you are not authorized to be in Acme.com network, then you
must disconnect immediately.
Todd#
```

Yukarıdaki MOTD banner aslında router'a bağlanan birine, şayet konuk listesinde değilse, gitmek zorunda kalacağını söylemektedir! Anlamanız gereken bölüm, sınırlayıcı karakterdir (bu, mesajın tamamlandığını, router'a söylemek için kullanılmaktadır). Onun için istediğiniz herhangi bir karakteri kullanabilirsiniz. Fakat (bunun açık olduğunu umuyorum), ayırıcı karakter, mesajın kendisinde kullanılamaz. Ayrıca, mesaj tamamlanınca, Enter tuşuna basın, sonra ayırıcı karakteri yazın ve tekrar Enter tuşuna basın. Bunu yapmazsanız, yine çalışacaktır, fakat birden fazla banner varsa tek bir mesaj gibi birleşerek, tek bir satırda görüneceklerdir.

Örneğin, bir banner'ı tek satırda şu şekilde ayarlayabilirsiniz:

```
Todd(config)#banner motd x Unauthorized access prohibited! X
```

Bu örnek iyi çalışacaktır, fakat başka bir MOTD banner mesajı eklerseniz, onlar tek bir satırda görüneceklerdir.

Aşağıda, bahsettiğim diğer banner'lar hakkında bazı detayları bulabilirsiniz:

**Exec banner:** Bir EXEC prosesi (bir VTY hattına gelen bir bağlantı veya line aktivasyonu gibi) oluşturulduğunda, görüntülenmesi için bir exec banner oluşturabilirsiniz. Bir konsol port'undan kullanıcı exec oturumu başlatarak, exec banner'ı etkin kılabilirsiniz.

**Incoming banner:** Reverse Telnet hatlarına bağlı terminallerde görüntülemek için bir banner oluşturabilirsiniz. Bu banner, reverse Telnet kullanıcılarına açıklamalarda bulunmak için kullanılmaktadır.

**Login banner:** Bağlı bütün terminallerde, görüntülenmesi için bir login banner oluşturabilirsiniz. Bu banner, MOTD banner'dan sonra, login satırından önce görüntülenmektedir. Login banner, satır satır devre dışı bırakılamaz, bu nedenle onu tamamıyla devreden çıkarmak için `no banner login` komutu ile silersiniz.

İşte bir login banner örneği:

```
!
banner login ^C
_____________________________

Cisco Router and Security Device Manager (SDM) is installed on
this device.
This feature requires the one-time use of the username "cisco"
with the password "cisco". The default username and password have
a privilege level of 15.
Please change these publicly known initial credentials using SDM
or the IOS CLI.
Here are the Cisco IOS commands.
username <myuser>  privilege 15 secret 0 <mypassword>
no username cisco
```

```
Replace <myuser> and <mypassword> with the username and password
you want to use.
For more information about SDM please follow the instructions in
the QUICK START
GUIDE for your router or go to http://www.cisco.com/go/sdm
————————————————————
^C
!
```

Yukarıdaki login banner oldukça tanıdık gelmeli. ISR router'ları için varsayılan konfigürasyonunda, Cisco'nun sahip olduğu banner'dır. Bu banner, login satırından önce, MOTD banner'dan sonra görüntülenmektedir.

## Şifreleri Ayarlamak

Cisco router'larınızı güvenli kılmak için beş şifre kullanılmaktadır: konsol, auxiliary, telnet (VTY), enable password ve enable secret. Enable secret ve enable password, privileged modu güvenli kılmak için kullanılan şifreleri oluşturmakta kullanılmaktadır. `enable` komutu kullanıldığında bu, bir kullanıcı için şifre istenen komut satırı olacaktır. Diğer üçü, user moduna, konsol port'undan, auxiliary port'undan veya Telnet üzerinden erişildiğinde, bir şifre yapılandırmak için kullanılmaktadır.

Gelin şimdi bunların hepsine bir bakalım:

### Enable Password'ler

Enable password'leri, global configuration modda şu şekilde ayarlarsınız:

```
Todd(config)#enable ?
 last-resort Define enable action if no TACACS servers
             respond
 password    Assign the privileged level password
 secret      Assign the privileged level secret
 use-tacacs  Use TACACS to check enable passwords
```

Aşağıdakiler, `enable password` parametrelerini açıklamaktadır:

**`last-resort`:** Bir TACACS üzerinden kimlik denetimi oluşturduysanız ve artık o kullanılmıyorsa, hala router'a girmenizi sağlar. Şayet TACACS sunucusu çalışıyorsa, kullanılmaz.

**`password`:** Enable password, eski 10.3 öncesi sistemlerde ayarlanır ve şayet enable secret ayarlıysa kullanılmaz.

**`secret`:** Bu, ayarlandığında enable password'ü geçersiz kılan, daha yeni, şifrelenmiş password'dür.

**`use-tacacs`:** Bu router'a, bir TACACS sunucusu üzerinden kimlik denetimi yapmasını söyler. Şayet çok sayıda router'a sahipseniz, bu kullanışlıdır. Çünkü tüm bu router'lardaki şifre değiştirme işlemiyle uğraşmak istemezsiniz. Onun yerine, sadece TACACS sunucusuna gidin ve şifreyi bir defa değiştirmek zorunda kalın.

Aşağıda, enable password oluşturma ile ilgili örnek bulabilirsiniz:

```
Todd(config)#line ?
  <0-337>  First Line number
```

```
aux      Auxiliary line
console  Primary terminal line
tty      Terminal controller
vty      Virtual terminal
x/y      Slot/Port for Modems
x/y/z    Slot/Subslot/Port for Modems
```

Şunlar, bilmeniz gereken line 'lardır:

**aux:** user-mode şifreleri, auxiliary port'u için ayarlanır. O, genellikle, bir modemi router'a bağlamak için kullanılır. Fakat aynı zamanda bir konsol olarak da kullanılabilmektedir.

**console:** Bir konsol user-mod şifresi ayarlar.

**vty:** Router'da bir Telnet şifresi oluşturur. Şayet şifre ayarlanmamışsa, varsayılan olarak Telnet kullanılamaz.

User-mode şifreleri ayarlamak için, istediğiniz line'ı yapılandırın ve router'a, kimlik denetimi isteğinde bulunması için ya `login` ya da `no login` komutlarını kullanın. Sonraki bölümler size, her line için satır satır yapılandırma örneği verecektir.

## Auxiliary Şifresi

Auxiliary şifresi yapılandırmak için global configuration moda gidin ve `line aux ?` yazın. Burada sadece 0-0 seçeneği görebilirsiniz (çünkü sadece bir port vardır):

```
Todd#config t
Enter configuration commands, one per line.  End with CNTL/Z.
Todd(config)#line aux ?
  <0-0>  First Line number
Todd(config)#line aux 0
Todd(config-line)#login
% Login disabled on line 1, until 'password' is set
Todd(config-line)#password aux
Todd(config-line)#login
```

NOT *Cisco'nun, (12.2 ve daha yukarısı) yeni IOS'ları bulunan router'lardaki yeni şifre özelliğine sahip olsa da, bunun,tüm IOS'larında geçerli olmadığını kesinlikle hatırlamalısınız.*

`login` komutunu ya da auxiliary portunun kimlik denetimi istemeyeceğini hatırlamanız önemlidir.

Cisco bu prosese, line'da bir şifre oluşturmadan önce login komutunu ayarlamanıza izin vermeyerek başladı. Çünkü şayet bir line içinde login komutunu çalıştırır ve sonra bir şifre ayarlamazsanız, line kullanılmaz olacaktır. O, olmayan bir şifre isteğinde bulunacaktır. Bu nedenle güzel bir özelliktir, problem değildir!

## Konsol Şifresi

Konsol şifresini oluşturmak için, `line console 0` komutunu kullanın. Fakat (config-line)# satırından, `line console 0 ?` yazmayı denediğimde ne olduğuna bakın. Bir hata aldım. Hala, `line console 0` yazabilirsiniz ve onu kabul edecektir. Fakat help ekranı bu satırdan çalışmayacaktır. Bir önceki seviyeye geçmek için `exit` yazın ve help ekranınızın şimdi çalıştığını göreceksiniz. Bu bir özelliktir.

İşte bir örnek:

```
Todd(config-line)#line console ?
% Unrecognized command
Todd(config-line)#exit
Todd(config)#line console ?
  <0-0>  First Line number
Todd(config-line)#password console
Todd(config-line)#login
```

Sadece bir konsol port'u olduğundan, yalnız line console 0'ı seçebilirim. Tüm line şifrelerinizi aynı ayarlayabilirsiniz. Fakat güvenlik nedenleriyle, onları farklı yapmanızı öneririm.

Konsol port'u için bilmeniz gereken birkaç önemli komut daha vardır.

Bunlardan birisi olan `exec-timeout 0 0` komutu, konsol EXEC oturumu için zaman-aşımı süresini 0'a ayarlar. Bu basit olarak, hiçbir zaman zaman-aşımına uğramayacağı anlamına gelir. Varsayılan olarak zaman-aşımı süresi 10 dakikadır. (İşteki insanların yaramaz olduklarına inanıyorsanız: Onu 0 1'e ayarlayın. Bu konsol zaman-aşımı süresini 1 saniyeye ayarlayacaktır. Onu düzeltmek için diğer elinizle zaman-aşımı süresini değiştirirken, sürekli aşağı ok tuşuna basmalısınız.)

`logging synchronous`, çok hoş bir komuttur. Varsayılan bir komut olmalıdır, fakat değildir. Bu komut, aniden çıkan ve yazmaya çalıştığınız girdilerinizi bölen konsol mesajlarının oluşturduğu sinir bozucu durumu engeller. Mesajlar hala görünmektedir, fakat router komut satırınıza, girişleriniz bölünmeksizin geri döndürülürsünüz. Bu, komutlarınızın okunmasını çok kolaylaştırır.

NOT

*Konsolunuzu, asla zaman aşımına uğramaması (0 0) için ve 35,791 dakikada (2,147,483 saniye) zaman aşımı olması için ayarlayabilirsiniz. Varsayılan 10 dakikadır.*

Aşağıda her iki komutun nasıl yapılandırılacağıyla ilgili örnek vardır:

```
Todd(config-line)#line con 0
Todd(config-line)#exec-timeout ?
  <0-35791>  Timeout in minutes
Todd(config-line)#exec-timeout 0 ?
  <0-2147483>  Timeout in seconds
  <cr>
Todd(config-line)#exec-timeout 0 0
Todd(config-line)#logging synchronous
```

## Telnet Şifresi

Bir router'a Telnet erişimi için user-mode şifresi ayarlamak için `line vty` komutunu kullanın. Cisco IOS Enterprise edition çalışmayan router'lar, beş VTY line sağlar (0'dan 4'e kadar). Şayet Enterprise edition kullanıyorsanız, çok daha fazlasına sahip olursunuz. Kaç tane line'a sahip olduğunuzu öğrenmenin en iyi yolu, soru işareti kullanılmaktır:

```
Todd(config-line)#line vty 0 ?
% Unrecognized command
Todd(config-line)#exit
Todd(config)#line vty 0 ?
  <1-1180>  Last Line number
  <cr>
Todd(config)#line vty 0 1180
```

```
Todd(config-line)#password telnet
Todd(config-line)#login
```

NOT *VTY line'da şifreden önce, login komutunu çalıştırmak zorunda olabilirsiniz de olmayabilirsiniz de. Bu IOS versiyonuna bağlıdır. Netice her iki yolda da aynıdır.*

`(config-line)#` satırından yardım bilgilerine ulaşamayacağınızı hatırlayın. Soru işaretini(?) kullanmak için, privileged moda geri dönmelisiniz.

Şayet, bir VTY şifresi ayarlanmamış router'a telnet yapmayı denerseniz, ne olacaktır? Bağlantının kabul edilmediğiyle ilgili bir mesaj alırsınız, çünkü şifre oluşturulmamıştır. Bundan dolayı, bir router'a telnet yapar ve şu mesajı alırsanız:

```
Todd#telnet SFRouter
Trying SFRouter (10.0.0.1)…Open

Password required, but none set
[Connection to SFRouter closed by foreign host]
Todd#
```

Uzak router (bu örnekte SFRouter) , ayarlanmış bir VTY (Telnet şifresine) sahip değildir. `no login` komutunu kullanarak, şifresiz Telnet bağlantısına izin verebilirsiniz.

```
SFRouter(config-line)#line vty 0 4
SFRouter(config-line)#no login
```

Router'larınız bir IP adresi ile yapılandırıldığında, bir konsol kablosu kullanmak zorunda olmak yerine router'larınızı yapılandırıp kontrol etmek için Telnet programını kullanabilirsiniz. Telnet programını, herhangi bir komut satırından (DOS veya Cisco), `telnet` yazarak kullanabilirsiniz. Telnet ile ilgili daha fazla konu, Modül5'te işlenecektir.

NOT *Bir test ya da ders ortamında bulunmadıkça, şifresiz olarak, no login komutunu kullanarak Telnet bağlantılarına izin vermenizi tavsiye etmem. Çalışan ağda, VTY şifresini daima ayarlamalısınız.*

## Secure Shell (SSH) Kurmak

Telnet yerine, Secure Shell kullanabilirsiniz. SSH, şifrelenmemiş veri akışı kullanan Telnet uygulamalarından daha güvenli bir oturum oluşturur. Secure Shell (SSH), veri göndermek için şifreli anahtarlar kullanır, böylece kullanıcı adı ve şifreniz açık olarak gönderilmez.

Aşağıda, SSH kurmanın adımlarını bulabilirsiniz:

1. Hostname ayarlanır:

```
Router(config)#hostname Todd
```

2. Domain ismi ayarlanır (hem hostname hem de domain adı, şifreli anahtarların üretilmesi için gerekmektedir):

```
Todd(config)#ip domain-name Lammle.com
```

3. Oturumu güvenli kılmak için şifreli anahtarlar üretilir:

```
Todd(config)#crypto key generate rsa general-keys modulus ?
<360-2048>  size of the key modulus [360-2048]
Todd(config)#crypto key generate rsa general-keys modulus 1024
The name for the keys will be: Todd.Lammle.com
% The key modulus size is 1024 bits
% Generating 1024 bit RSA keys, keys will be non-exportable...[OK]
 *June 24 19:25:30.035: %SSH-5-ENABLED: SSH 1.99 has been enabled
```

4. SSH oturumu için maksimum idle (boşta bekleme) zamanlayıcısı ayarlanır:

```
Todd(config)#ip ssh time-out ?
<1-120>  SSH time-out interval (secs)
Todd(config)#ip ssh time-out 60
```

5. Bir SSH bağlantısı için maksimum hatalı girişim sayısı ayarlanır:

```
Todd(config)#ip ssh authentication-retries ?
<0-5>  Number of authentication retries
Todd(config)#ip ssh authentication-retries 2
```

6. Router'ın vty line'larına bağlanılır:

```
Todd(config)#line vty 0 1180
```

7. Son olarak, SSH ve sonrada Telnet'i erişim protokolleri olarak yapılandırılır:

```
Todd(config-line)#transport input ssh telnet
```

Şayet komut sırasının sonunda telnet kullanmazsanız, router'da sadece SSH çalışacaktır. Her ikisinin birlikte kullanılmasını tavsiye etmem, fakat SSH'ın Telnet'ten daha güvenli olduğunu anlayın.

## Password'lerinizi Şifrelemek

Varsayılan olarak sadece enable secret password'ü şifrelendiğinden, user-mode ve enable password'lerini şifrelemek için manuel kofigürasyona ihtiyacınız olacaktır.

Bir router'da, `show running-config` çalıştırdığınızda, enable secret dışında tüm password'lerin görünebildiğine dikkat edin:

```
Todd#sh running-config
Building configuration...
[output cut]
!
enable secret 5 $1$2R.r$DcRaVoOyBnUJBf7dbG9XE0
enable password todd
!
[output cut]
!
line con 0
 exec-timeout 0 0
 password console
 logging synchronous
 login
line aux 0
 password aux
 login
line vty 0 4
 access-class 23 in
 privilege level 15
 password telnet
 login
```

```
 transport input telnet ssh
line vty 5 15
 access-class 23 in
 privilege level 15
 password telnet
 login
 transport input telnet ssh
line vty 16 1180
 password telnet
 login
!
end
```

Password'lerinizi manuel olarak şifrelemek için `service password-encryption` komutunu kullanın. Bunu nasıl yapacağınızla ilgili örnek aşağıdadır:

```
Todd#config t
Enter configuration commands, one per line.  End with CNTL/Z.
Todd(config)#service password-encryption
Todd(config)#exit
Todd#sh run
Building configuration...
[output cut]
!
enable secret 5 $1$2R.r$DcRaVoOyBnUJBf7dbG9XE0
enable password 7 131118160F
!
[output cut]
!
line con 0
 exec-timeout 0 0
 password 7 060500 2F5F41051C
 logging synchronous
 login
line aux 0
 password 7 03054E13
 login
line vty 0 4
 access-class 23 in
 privilege level 15
 password 7 010703085500E12
 login
 transport input telnet ssh
line vty 5 15
 access-class 23 in
 privilege level 15
```

```
 password 7 01070308550E12
 login
 transport input telnet ssh
line vty 16 1180
 password 7 120D001B1C0E18
 login
!
end

Todd#config t
Todd(config)#no service password-encryption
Todd(config)#^Z
Todd#
```

Password'ler şimdi, şifrelenmiş olacaktır. Sadece password'leri şifreleyin, `show run` çalıştırın ve komutu kapatın. Enable password ve line password'lerinin hepsinin şifrelendiğini görebilirsiniz.

Router'larınızda, description (açıklama) ayarlarını göstermeden önce password'leri şifrelemekten biraz daha bahsedelim. Söylediğim gibi password'lerinizi ayarlarsanız ve sonra `service password-encryption` komutunu açarsanız, şifreleme servisini kapatmadan önce, bir `show running-config` çalıştırmalısınız yoksa password'leriniz şifrelenmeyecektir. Şifreleme servisini kapatmak zorunda değilsiniz. Şayet router'larınız düşük işlemciyle çalışıyorsa, bunu yaparsınız. Password'lerinizi ayarlamadan önce servisi açarsanız, onların şifrelemiş olduklarına bakmak zorunda dahi değilsiniz.

## Description (Açıklama)

Bir interface'deki açıklamaları ayarlamak, yöneticilere fayda sağlar ve hostname'de olduğu gibi, sadece lokalde anlamı vardır. `description` komutu, faydalıdır, çünkü örneğin, devre numaralarını takip etmek için kullanabilirsiniz.

Bir örnek verelim:

```
Todd#config t
Todd(config)#int s0/0/0
Todd(config-if)#description Wan to SF circuit number 6fdda12345678
Todd(config-if)#int fa0/0
Todd(config-if)#description Sales VLAN
Todd(config-if)#^Z
Todd#
```

Bir interface'in açıklamasını, `show running-config` ya da `show interface` komutu ile görebilirsiniz:

```
Todd#sh run
[output cut]
!
interface FastEthernet0/0
 description Sales VLAN
 ip address 10.10.10.1 255.255.255.248
 duplex auto
```

```
 speed auto
!
interface Serial0/0/0
 description Wan to SF circuit number 6fdda 12345678
 no ip address
 shutdown
!
[output cut]
Todd#sh int f0/0
FastEthernet0/0 is up, line protocol is down
  Hardware is MV96340 Ethernet, address is 001a.2f55.c9e8 (bia
001a.2f55.c9e8)
  Description: Sales VLAN
 [output cut]

Todd#sh int s0/0/0
Serial0/0/0 is administratively down, line protocol is down
  Hardware is GT96K Serial
  Description: Wan to SF circuit number 6fdda12345678
```

### *description*: Faydalı Bir Komut

*San Francisco'daki Acme Corporation'da kıdemli bir network yöneticisi olan Bob, U.S ve Kanada boyunca çeşitli şubelere 50 WAN linke sahiptir. Ne zaman bir interface bozulsa, Bob, hem devre numarasını hem de WAN link sağlayıcısının telefon numarasını öğrenmeye çalışırken çok zaman harcamaktadır.*

`description` *interface komutu, Bob için çok faydalı olabilir, çünkü her router interface'inin tam olarak nereye bağlı olduğunu ayırt etmek için LAN linklerinde bu komutu kullanabilir. İlgili servis sağlayıcının telefon numarası ile birlikte, her WAN interface'ine devre numaralarını ekleyerek çok büyük kazanç sağlayabilir.*

*Birkaç saat harcayıp, bu bilgileri her router interface'ine ekleyerek, WAN linklerinde problem olduğunda (bunun olacağını biliyorsunuz!), Bob çok büyük zaman kazanabilir.*

## Do Komutunu Kullanmak

IOS versiyon 12.3 ile başlayarak Cisco son olarak, configuration moddan yapılandırma ve istatistiklere bakmanızı sağlayan bir komutu IOS'a ekledi.(size yukarda verdiğim örneklerde, tüm `show` komutları priviledge moddan çalışıyordu)

Aslında, 12.3 öncesi router ile global-config moddan yapılandırmaya göz atmaya çalıştığınızda, aşağıdaki hatayı alıyordunuz:

```
Router(config)#sh run
               ^
% Invalid input detected at '^' marker.
```

**12.4 IOS çalışan router'ımda aynı komutu girdiğimde aldığım çıktıyı mukayese edin:**

```
Enter configuration commands, one per line.  End with CNTL/Z.
Todd(config)#do show run
```

```
Building configuration...

Current configuration : 3276 bytes
!
[output cut]

Todd(config)#do sh int f0/0
FastEthernet0/0 is up, line protocol is down
  Hardware is MV96340 Ethernet, address is 001a.2f55.c9e8 (bia
    001a.2f55.c9e8)
  Description: Sales VLAN
[output cut]
```

Aslında herhangi bir komutu, herhangi bir konfigürasyon istemcisinden çalıştırabileceksiniz. Güzel, değil mi? Bizim password'leri şifreleme ile ilgili örneğimize dönecek olursak, do komutu kesinlikle eğlenceyi daha önce başlatır, bu nedenle gerçekten çok iyi bir şeydir dostlarım!

# Router Interface'leri

Interface yapılandırmaları, en önemli router yapılandırılmalarından biridir. Çünkü interface olmaksızın router, tamamıyla kullanışsız bir nesne olur. Artı, interface konfigürasyonları, diğer cihazlarla iletişimi sağlamak için tamamıyla doğru yapılmalıdır. Network katmanı adresleri, ortam araç tipi, bant genişliği ve diğer yönetici komutlarının tamamı bir interface'i yapılandırmak için kullanılmaktadır.

Farklı router'lar, üzerlerinde kullanılan interface'leri seçmek için farklı yöntemler kullanırlar. Örneğin, aşağıdaki komutlar, 0'dan 9'a sınıflandırılmış, 10 interface'li 2522 Cisco router'ı göstermektedir:

```
Router(config)#int serial ?
 <0-9> Serial interface number
```

Şimdi, yapılandırmak istediğimiz interface'i seçme zamanı geldi. Bunu yapınca, belirtilen interface için, interface configuration modda olacaksınız. Aşağıdaki komut, örneğin, seri port 5'i seçmek için kullanılabilir:

```
Router(config)#int serial 5
Router(config)-if)#
```

2522 router, bir adet Ethernet 10BaseT port'una sahiptir ve `interface ethernet 0` yazarak, bu interface'i yapılandırabilirsiniz:

```
Router(config)#int ethernet ?
 <0-0> Ethernet interface number
Router(config)#int ethernet 0
Router(config-if)#
```

Yukarda gösterdiğim gibi, 2500 router, sabit-konfigürasyonlu bir router'dır. Yani, bu modeli satın aldığınızda, bu fiziksel konfigürasyonla baş başa kalırsınız. Onları fazla kullanmamamın en büyük sebebi budur. Onları kesinlikle, test ve sınıf ortamı dışında kullanmayacağım.

Bir interface'i yapılandırmak için daima `interface type` ***number*** sırasını kullanırız, fakat 2600 ve 2800 serilerinde (aslında, bu konuda tüm ISR router'lar), router'da fiziksel bir slot vardır (slota

takılı modülde bir port numarası ile). Bu sebeple, modüler bir router'da, yapılandırma interface `type slot / port` şeklinde olacaktır:

```
Router(config)#int fastethernet ?
 <0-1> FastEthernet interface number
Router(config)#int fastethernet 0
% Incomplete command.
Router(config)#int fastethernet 0?
/
Router(config)#int fastethernet 0/?
 <0-1> FastEthernet interface number
```

Sadece `int fastethernet 0` yazamayacağınızı unutmayın. Komutun tamamını yazmalısınız: `type slot / port` ya da `fastethernet 0/0` (veya `int fa 0/0`).

ISR serileri için aslında aynıdır, sadece daha fazla seçeneğe sahipsiniz. Örnek olarak, yerleşik FastEthernet interface'leri, 2600 serisinde kullandıklarımızla aynı konfigürasyonla çalışır:

```
Todd(config)#int fastEthernet 0/?
  <0-1>  FastEthernet interface number
Todd(config)#int fastEthernet 0/0
Todd(config-if)#
```

Fakat diğer modüller farklıdır. Onlar iki, yerine üç numara kullanır. İlk 0, router'ın kendisidir, sonra slot'u ve en son port'u seçersiniz. Aşağıda, 2811'deki seri interface ile ilgili bir örnek vardır:

```
Todd(config)#interface serial ?
  <0-2>  Serial interface number
Todd(config)#interface serial 0/0/?
  <0-1>  Serial interface number
Todd(config)#interface serial 0/0/0
Todd(config-if)#
```

Biliyorum, bu biraz karışık görünebilir, fakat bana güvenin, gerçekte o kadar zor değildir! İlk olarak daima running-config çıktısına bakmanız gerektiğini hatırlamanız yardımcı olacaktır. Böylece, ilgilenmek zorunda olacağınız interface'in ne olduğunu bilirsiniz. Aşağıda 2811'imin çıktısı vardır:

```
Todd(config-if)#do show run
Building configuration...
[output cut]
!
interface FastEthernet0/0
 no ip address
 shutdown
 duplex auto
 speed auto
!
interface FastEthernet0/1
 no ip address
 shutdown
```

```
 duplex auto
 speed auto
!
interface Serial0/0/0
 no ip address
 shutdown
 no fair-queue
!
interface Serial0/0/1
 no ip address
 shutdown
!
interface Serial0/1/0
 no ip address
 shutdown
!
interface Serial0/2/0
 no ip address
 shutdown
 clock rate 2000000
!
 [output cut]
```

Running-config çıktısının tamamını göstermedim, fakat ihtiyacınız olan herşeyi görüntüledim. İki yerleşik FastEthernet interface'i, slot 0 'daki iki seri interface'i (0/0/0 ve 0/0/1), slot 1'deki seri interface'i (0/1/0) ve slot 2 deki seri interface'i (0/2/0) görebilirsiniz. Interface'leri bu şekilde görünce, router'a modüllerin nasıl yerleştirildiğini anlamanız çok kolay olacaktır.

Şayet bir 2500'de `interface e0`, 2600'da `interface fastethernet 0/0` veya 2800'de `interface serial 0/1/0` yazarsanız, tüm yaptığınızın, konfigürasyon için bir interface seçmek olduğunu bilin. Aslında bundan sonra, hepsi aynı şekilde yapılandırılır.

Sonraki bölümlerde router interface'lerinden bahsetmeye devam edeceğim. Bir interface'in nasıl aktif edileceğini ve bir router interface'ine nasıl IP adresi verileceğini göstereceğim.

## Bir Interface'i Aktif Hale Getirmek

Bir interface'i `shutdown` komutu ile kapatabilir ve `no shutdown` komutu ile aktif hale getirebilirsiniz.

Şayet bir interface kapatılırsa, `show interfaces` (kısaca, `sh int`) komutu kullandığınızda o,administratively down olarak görünür:

```
Todd#sh int f0/1
FastEthernet0/1 is administratively down, line protocol is down
[output cut]
```

Bir interface'in durumunu kontrol etmenin diğer bir yolu, `show running-config` komutunu kullanmaktır. Tüm interfaceler varsayılan olarak kapalıdır. Interface'i, no `shutdown` (kısaca no `shut`) komutunu kullanarak aktif hale getirebilirsiniz:

```
Todd#config t
Todd(config)#int f0/1
```

```
Todd(config-if)#no shutdown
Todd(config-if)#
*Feb 28 22:45:08.455: %LINK-3-UPDOWN: Interface FastEthernet0/1,
     changed state to up
Todd(config-if)#do show int f0/1
FastEthernet0/1 is up, line protocol is up
[output cut]
```

Bir Interface'de IP adresi Yapılandırmak

Router'larınızda IP adresi kullanmak zorunda olmadığınız halde, insanlar genelde kullanırlar! Bir interface'de IP adresi yapılandırmak için interface configuration moddan `ip address` komutunu kullanın:

```
Todd(config)#int f0/1
Todd(config-if)#ip address 172.16.10.2 255.255.255.0
```

NOT *ip address address mask komutu, interface'deki IP işlemini başlatır.*

Interface'i `no shutdown` komutu ile enable etmeyi sakın unutmayın. Interface'in administravely shut down olup olmadığını öğrenmek için show interface komutuna bakacağınızı hatırlayın. Show running-config, yine aynı bilgiyi verecektir.

Şayet bir interface'e ikinci bir subnet mask adresi vermek isterseniz, `secondary` parametresini kullanmak zorunda kalırsınız. Şayet başka bir IP adresi yazar, Enter tuşuna basarsınız, o, mevcut IP adresi ve maskın yerine geçecektir. Bu kesinlikle Cisco IOS'unun en mükemmel özelliğidir.

Şimdi tekrar deneyelim. İkinci bir IP adresi eklemek için sadece `secondary` parametresi kullanın:

```
Todd(config-if)#ip address 172.16.20.2 255.255.255.0 ?
  secondary  Make this IP address a secondary address
  <cr>
Todd(config-if)#ip address 172.16.20.2 255.255.255.0 secondary
Todd(config-if)#^Z
Todd(config-if)#do sh run
Building configuration...
[output cut]

interface FastEthernet0/1
 ip address 172.16.20.2 255.255.255.0 secondary
 ip address 172.16.10.2 255.255.255.0
 duplex auto
 speed auto
!
```

Etkili olmadığı için aslında bir interface'e birden fazla IP adresi verilmesini tavsiye etmem. Fakat ben yine de, belki bir gün kendinizi kötü network tasarımı yapan MIS müdürü ile çalışır ve sizden ağı yönetmenizi ister bulursanız diye gösterdim. Kim bilir? Belki birisi size bunu sorar ve bildiğiniz için oldukça zeki görünürsünüz.

## Pipe(|) Kullanmak

Yok, borudan bahsetmiyorum. Output modifier demek istiyorum. (Kariyerimde birçok router konfigürasyonu görmeme rağmen, bazen şaşırırım!) Pipe (|), tüm konfigürasyonları veya diğer uzun

çıktıları zorla tamamlamamızı ve isteklerimizi hızlı açıklamamızı sağlar. Aşağıda bir örnek bulabilirsiniz:

```
Todd#sh run | ?
  append    Append redirected output to URL (URLs supporting
append operation
            only)
  begin     Begin with the line that matches
  exclude   Exclude lines that match
  include   Include lines that match
  redirect  Redirect output to URL
  section   Filter a section of output
  tee       Copy output to URL

Todd#sh run | begin interface
interface FastEthernet0/0
 description Sales VLAN
 ip address 10.10.10.1 255.255.255.248
 duplex auto
 speed auto
!
interface FastEthernet0/1
 ip address 172.16.20.2 255.255.255.0 secondary
 ip address 172.16.10.2 255.255.255.0
 duplex auto
 speed auto
!
interface Serial0/0/0
 description Wan to SF circuit number 6fdda 12345678
 no ip address
!
```

Yani basitçe, pipe sembolü (output modifier), bir router'ın tüm konfigürasyonu içinde dolanıp durmaktansa, gitmek istediğiniz yere çok hızlı gitmenize yardım etmesi için, ihtiyacınız olan şeydir

Belirli bir route'un, büyük bir routing tablosunda olup olmadığını anlamayı istediğimde onu kullandım. İşte bir örnek:

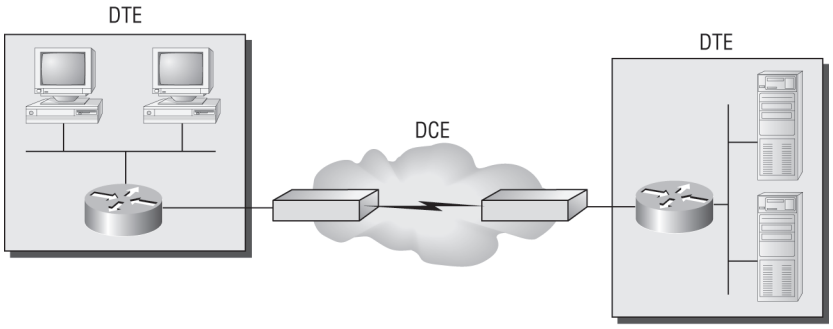
```
Todd#sh ip route | include 192.168.3.32
R       192.168.3.32 [120/2] via 10.10.10.8, 00:00:25,
FastEthernet0/0
Todd#
```

İlk olarak, routing tablonuzun 100 kayıttan fazlasına sahip olduğunu bilmelisiniz. Yoksa güvenilir pipe olmaksızın, muhtemelen bu çıktıya bakabilirim! Bu etkili araç, size zaman kazandırır ve bir konfigürasyondaki satırı kolayca bulmanızı sağlar (ya da yukarıdaki örnekteki gibi çok büyük bir routing tablosundan tek bir route'u bulmanızı sağlar).

Pipe komutuna biraz zaman harcayıp onu iyi kullanırsanız, router çıktılarını çabucak ayrıştırma konusunda gerçekten başarılı olabilirsiniz.

## Seri Interface Komutları

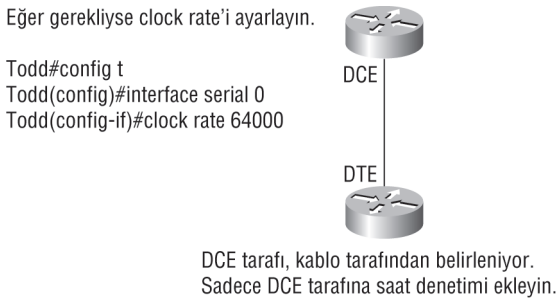
Bir seri interface'i yapılandırmaya geçmeden önce, bazı önemli bilgilere ihtiyacınız var. Şekil 4.4'te gösterildiği gibi, interface genellikle, router'a hat üzerinde saat denetimi sağlayan bir cihaz türü olan CSU/DSU'ya bağlı olacaktır.




Saat denetimi DCE network tarafından router'lar için sağlanır.
Lab ortamında her zaman DCE network bulunmaz.

***Şekil 4.4:*** *Tipik bir WAN bağlantısı.*

Burada görebileceğiniz gibi, bir CSU/DSU üzerinden DCE ağına bağlanmak için seri interface kullanılmaktadır. CSU/DSU, router interface'ine saat denetimi sağlamaktadır. Şayet back-to-back bir konfigürasyonunuz varsa (örneğin, Şekil 4.5'de gösterilen lab ortamında kullanıldığı şekilde), uçlardan biri (kablonun data communication equipment (DCE) ucu) saat denetimi sağlamalıdır!




DCE tarafı, kablo tarafından belirleniyor.
Sadece DCE tarafına saat denetimi ekleyin.

**Show controller'lar kablo bağlantı tipini gösterir.**

***Şekil 4.5:*** *Lab ortamındaki ağda saat denetimi sağlamak.*

Varsayılan olarak, Cisco router'ların hepsi, data terminal equipment (DTE) cihazlardır. Bunun anlamı, şayet onun bir DCE cihazı gibi davranmasını istiyorsanız, bir interface'ini saat denetimi sağlaması için yapılandırmanız gerektiğidir. Tekrar belirtmem gerekiyor ki, örneğin gerçek bir ağda bulunan T1 bağlantısına saat denetimi sağlayamazsınız, çünkü Şekil 4.4 'de gösterildiği gibi, seri interface'inize bağlı bir CSU/DSU vardır.

Bir DCE seri interface'ini, `clock rate` komutu ile yapılandırırsınız:

```
Todd#config t
Enter configuration commands, one per line.  End with CNTL/Z.
Todd(config)#int s0/0/0
Todd(config-if)#clock rate ?
        Speed (bits per second)
  1200
  2400
  4800
  9600
```

```
  14400
  19200
  28800
  32000
  38400
  48000
  56000
  57600
  64000
  72000
  115200
  125000
  128000
  148000
  192000
  250000
  256000
  384000
  500000
  512000
  768000
  800000
  1000000
  2000000
  4000000
  5300000
  8000000

  <300-8000000>    Choose clockrate from list above

Todd(config-if)#clock rate 1000000
```

`clock rate` komutu, saniyedeki bit sayısı (bits per second) olarak ayarlanır. DCE veya DTE olduğunu kontrol etmek için kablo ucuna bakmak dışında, router'ın seri interface'inin DCE kabloya sahip olup olmadığını, `show controllers int` komutu ile görebilirsiniz:

```
Todd#sh controllers s0/0/0
Interface Serial0/0/0
Hardware is GT96K
DTE V.35idb at 0x4342FCB0, driver data structure at 0x434373D4

Bir DCE bağlantısı gösteren çıktı örneği aşağıdadır:

Todd#sh controllers s0/2/0
Interface Serial0/2/0
Hardware is GT96K
DCE V.35, clock rate 1000000
```

Bilmeniz gereken diğer bir komut, `bandwidth` komutudur. Her Cisco router, varsayılan T1 (1.544Mbps) bant genişliği ile gelir. Bu, verinin bir link üzerinden nasıl transfer edileceği ile ilgili değildir. Bir seri linkin bant genişliği, EIGRP ve OSPF gibi routing protokolleri tarafından, uzak bir ağa en iyi yolu hesaplamak için kullanılmaktadır. Şayet RIP routing kullanıyorsanız, RIP'in bunu belirlemek için sadece hop sayısı kullanmasından dolayı, bir seri linke bant genişliği ayarlamak anlamsızdır. Bu bölümü tekrar okuduğunuzda routing protokolleri, metrikler nedir diye korkmayın. Bunları, bölüm 6, "IP Routing"de anlatacağım.

Aşağıda `bandwidth` komutu kullanılan bir örnek vardır:

```
Todd#config t
Todd(config)#int s0/0/0
Todd(config-if)#bandwidth ?
  <1-10000000>  Bandwidth in kilobits
  inherit       Specify that bandwidth is inherited
  receive       Specify receive-side bandwidth
Todd(config-if)#bandwidth 1000
```

`clock rate` komutunun tersine, `bandwidth` komutunun kilobit olarak ayarlandığına dikkat ettiniz mi?

NOT *clock rate komutu ile ilgili tüm bu konfigürasyon örneklerine baktıktan sonra, yeni ISR router'ların, DCE bağlantılarını otomatik olarak algıladıklarını ve clock rate'i 2000000 olarak ayarlandığını bilin. Bununla beraber, yeni router'lar onu otomatik ayarlasa da, clock rate komutunu bilmelisiniz.*

# Konfigürasyonlara Bakmak, Kaydetmek ve Silmek

Şayet setup modda çalışıyorsanız, yeni oluşturduğunuz konfigürasyonu kullanıp kullanmayacağınız sorulacaktır. Şayet evet derseniz bu, DRAM'de çalışan konfigürasyonu (running-config olarak bilinir), NVRAM'e kopyalayacak ve startup-config olarak isimlendirecektir. Daima setup mod yerine CLI veya SDM kullanacağınızı umuyorum.

Dosyayı, DRAM'den NVRAM'e manuel olarak, `copy running-config startup-config` komutunu kullanarak kaydedebilirsiniz (ayrıca kısaca `copy run star` olarak kullanabilirsiniz):

```
Todd#copy running-config startup-config
Destination filename [startup-config]? [press enter]
Building configuration...
[OK]
Todd#
Building configuration...
```

[ ]'de cevabı olan bir soru görürseniz, varsayılan cevabı seçmek için sadece Enter'a basın.

Ayrıca, komut hedef dosya ismi için istendiğinde varsayılan cevap, startup-config'di. Onu sormasının sebebi, konfigürasyonunu hemen hemen istediğiniz her yere kopyalayabilmenizdir. Bir bakalım:

```
Todd#copy running-config ?
  archive:        Copy to archive: file system
  flash:          Copy to flash: file system
  ftp:            Copy to ftp: file system
  http:           Copy to http: file system
  https:          Copy to https: file system
```

```
  ips-sdf         Update (merge with) IPS signature configuration
  null:           Copy to null: file system
  nvram:          Copy to nvram: file system
  rcp:            Copy to rcp: file system
  running-config  Update (merge with) current system configuration
  scp:            Copy to scp: file system
  startup-config  Copy to startup configuration
  syslog:         Copy to syslog: file system
  system:         Copy to system: file system
  tftp:           Copy to tftp: file system
  xmodem:         Copy to xmodem: file system
  ymodem:         Copy to ymodem: file system
```

Bölüm 5'te, dosyaları nereye ve nasıl kaydedeceğimize daha yakından bakacağız.

Dosyaları, privileged moddan, show running-config ya da `show startup-config` yazarak gözden geçirebilirsiniz. `Show running config` komutunun kısaltılmışı olan `sh run` komutu, bize mevcut konfigürasyon hakkında bilgi verir:

```
Todd#show running-config
Building configuration...

Current configuration : 3343 bytes
!
version 12.4
[output cut]
```

`Show startup-config` komutunun kısa yollarından biri olan `sh start` komutu, bize router tekrar başlatıldığında kullanılacak konfigürasyonu gösterir. Ayrıca, `startup-config` dosyasını saklamak için NVRAM'in ne kadarının kullanıldığını belirtir. İşte bir örnek:

```
Todd#show startup-config
Using 1978 out of 245752 bytes
!
version 12.4
[output cut]
```

## Konfigürasyonu Silmek ve Router'ı Reload Etmek

Startup-config dosyasını, `erase startup-config` komutunu kullanarak silebilirsiniz:

```
Todd#erase startup-config
Erasing the nvram filesystem will remove all configuration files!
    Continue? [confirm][enter]
[OK]
Erase of nvram: complete
Todd#
*Feb 28 23:51:21.179: %SYS-7-NV_BLOCK_INIT: Initialized the
geometry of nvram
Todd#sh startup-config
```

```
startup-config is not present
Todd#reload
Proceed with reload? [confirm]System configuration has been
modified.
    Save? [yes/no]: n
```

Şayet `erase startup-config` komutu kullandıktan sonra, router'ı reload ederseniz ya da elektriğini kapatıp açarsanız, NVRAM'de kayıtlı herhangi bir konfigürasyon olmadığından, setup moda girmek isteyip istemediğiniz sorulacaktır. Setup-moddan ne zaman çıkmak isterseniz, Ctrl+C tuş bileşimine basabilirsiniz (reload komutu sadece privileged modda kullanılabilir)

Bu noktada, router'ınızı yapılandırmak için setup modunu kullanamazsınız. Setup modunu istemiyorsanız, sadece no deyin, çünkü burası, Cash Line Interface'in (CLI) nasıl kullanılacağını bilmeyenlere yardım etmek içindir. Artık bu mod karşınıza çıkmayacaktır.

## Konfigürasyonunuzu Doğrulamak

Net olarak, show running config, konfigürasyonunuzu doğrulamak için, `show startup-config`, router bir dahaki sefere reload edildiğinde, konfigürasyonunuzu doğrulamak için en iyi yoldur.

Running-config'e bakınca, herşey iyi görünüyorsa, konfigürasyonunuzu Ping ve Telnet gibi araçlarla doğrulayabilirsiniz. Ping (Packet Internet Groper), ICMP echo request ve reply'lar kullanan bir programdır. (ICMP'den bölüm 2 "TCP/IP'ye Giriş"te bahsedildi.) Ping, uzaktaki host'a bir paket gönderir ve şayet host cevap verirse, onun aktif olduğunu bilirsiniz. Fakat onun hem aktif hem de iyi olup olmadığını bilemezsiz. Çünkü bir Microsoft sunucuyu pinglemeniz, onda oturum açabileceğiniz anlamına gelmez. Öyle olsa bile Ping, bir ağ topluluğunda hata tespiti için harika bir başlangıç noktasıdır.

Farklı protokollerle ping atabileceğinizi biliyor musunuz? Bunu yapabilirsiniz ve hem router user-modunda hem de priviledge modda, `ping ?` yazarak test edebilirsiniz:

```
Router#ping ?
  WORD       Ping destination address or hostname
  appletalk  Appletalk echo
  clns       CLNS echo
  decnet     DECnet echo
  ip         IP echo
  ipv6       IPv6 echo
  ipx        Novell/IPX echo
  srb        srb echo
  tag        Tag encapsulated IP echo
  <cr>
```

NOT *Cisco Discovery Protocol (CDP), bölüm 15 'te işlenecektir.*

Şayet bir komşunuzun Network katmanı adresini bulmak isterseniz, ya router'ın veya switch'in kendisine gitmeniz gerekir ya da ping'lemek için ihtiyacınız olan Network katmanı adresini bulmak için `show cdp entry *` protocol yazabilirsiniz.

Traceroute, bir ağ topluluğu boyunca bir paketin kullandığı yolu izlemek için IP TTL (time to live) zaman-aşımları ile ICMP kullanır. Ping'in tersine, sadece host'ları bulur ve cevaplar. Traceroute da birçok protokolle kullanılabilir.

```
Router#traceroute ?
  WORD       Trace route to destination address or hostname
```

```
  appletalk  AppleTalk Trace
  clns       ISO CLNS Trace
  ip         IP Trace
  ipv6       IPv6 Trace
  ipx        IPX Trace
  <cr>
```

Telnet, FTP ve HTTP, aslında en iyi araçlardır, çünkü onlar uzak bir host'la oturum oluşturmak için Network katmanında IP ve Transport katmanında TCP kullanır. Şayet bir cihaza, telnet, ftp ya da http yapabiliyorsanız, IP bağlantınız çalışıyor demektir.

```
Router#telnet ?
 WORD IP address or hostname of a remote system
 <cr>
```

Router komut satırından, sadece bir hostname ya da IP adresi yazın. Sizin telnet yapmayı istediğiniz düşünülecektir (gerçekte, telnet komutunu kullanmaya ihtiyacınız yoktur).

Aşağıdaki bölümlerde, interface istatistiklerini nasıl doğrulayacağınızı göstereceğim.

## show interface Komutu ile Doğrulamak

Konfigürasyonunuzu doğrulamanın diğer yolu, `show interface` komutunu yazmaktır. `show interface ?` yazarsanız, konfigürasyon için uygun tüm interface'ler ortaya çıkacaktır.

*show interface komutu, bir router'daki tüm interface'lerin ayarlanabilen parametrelerini ve istatistiklerini gösterir.*

NOT

Bu komut, router ve network sorunlarını doğrulamak ve hata tespiti için oldukça kullanışlıdır.

Aşağıdaki çıktı, yeni silinmiş ve reload edilmiş 2811 router'dan alınmıştır:

```
Router#sh int ?
  Async              Async interface
  BVI                Bridge-Group Virtual Interface
  CDMA-Ix            CDMA Ix interface
  CTunnel            CTunnel interface
  Dialer             Dialer interface
  FastEthernet       FastEthernet IEEE 802.3
  Loopback           Loopback interface
  MFR                Multilink Frame Relay bundle interface
  Multilink          Multilink-group interface
  Null               Null interface
  Port-channel       Ethernet Channel of interfaces
  Serial             Serial
  Tunnel             Tunnel interface
  Vif                PGM Multicast Host interface
  Virtual-PPP        Virtual PPP interface
  Virtual-Template   Virtual Template interface
  Virtual-TokenRing  Virtual TokenRing
  accounting         Show interface accounting
  counters           Show interface counters
  crb                Show interface routing/bridging info
  dampening          Show interface dampening info
```

```
  description        Show interface description
  etherchannel       Show interface etherchannel information
  irb                Show interface routing/bridging info
  mac-accounting     Show interface MAC accounting info
  mpls-exp           Show interface MPLS experimental accounting
info
  precedence         Show interface precedence accounting info
  pruning            Show interface trunk VTP pruning information
  rate-limit         Show interface rate-limit info
  stats              Show interface packets & octets, in & out,
by switching
                     path
  status             Show interface line status
  summary            Show interface summary
  switching          Show interface switching
  switchport         Show interface switchport information
  trunk              Show interface trunk information
  |                  Output modifiers
  <cr>
```

Sadece gerçek fiziksel interface'ler, FastEthernet, Serial ve Async 'dur, diğerleri doğrulamada kullanılan mantıksal interface'lerdir. Şimdiki komut, `show interface fastethernet 0/0` 'dır. Bu bize, hem donanım adresi, mantıksal adres ve enkapsülasyon yöntemi hem de collision'lardaki istatistikleri verir. Burada görebileceğiniz gibi:

```
Router#sh int f0/0
FastEthernet0/0 is up, line protocol is up
  Hardware is MV96340 Ethernet, address is 001a.2f55.c9e8 (bia
001a.2f55.c9e8)
  Internet address is 192.168.1.33/27
MTU 1500 bytes, BW 100000 Kbit, DLY 100 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  Keepalive set (10 sec)
  Auto-duplex, Auto Speed, 100BaseTX/FX
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input never, output 00:02:07, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output
drops: 0
  Queueing strategy: fifo
  Output queue: 0/40 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     0 packets input, 0 bytes
     Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
```

```
     0 watchdog
     0 input packets with dribble condition detected
     16 packets output, 960 bytes, 0 underruns
     0 output errors, 0 collisions, 0 interface resets
     0 babbles, 0 late collision, 0 deferred
     0 lost carrier, 0 no carrier
     0 output buffer failures, 0 output buffers swapped out
Router#
```

Muhtemelen tahmin ettiğiniz gibi, bu çıktıdaki, önemli istatistikleri konuşacağız. Fakat ilk olarak, size, FastEthernet 0/0 'ın hangi subnette olduğunu ve broadcast adresi ile geçerli host aralığının ne olduğunu sormak zorundayım.

Esasında bunları, çok hızlı bir şekilde yapabilmek zorundasınız. Yapamadıysanız, adres 192.168.1.33/27'dir. Dürüst olmalıyım ki, eğer bu noktada, /27'nin ne olduğunu bilmiyorsanız, sınavı geçmeniz mucize olacaktır. (/27, 255.255..255.224'tür). Dördüncü oktet, 32 blok boyutundadır. Subnetler, 0, 32, 64, …'dir. FastEthernet interface'i 32 subnet'indedir, broadcast adresi, 63 ve geçerli host aralığı, 33-62'dir.

Yukarıdaki interface çalışıyor ve iyi durumda olduğu görülüyor. `show interface` komutu, bir interface'ten hata alıp almadığınızı gösterecektir. Ayrıca, maximum transmission unit'leri (MTU), bandwidth (BW), reliability (255/255, mükemmel anlamındadır) ve load (1/255, yük yok demektir) görünecektir.

*Bunlardan herhangi biri ile sorun yaşıyorsanız, "Subnet'leme, VLSM'ler ve TCP/IP Hata Tespiti" başlıklı bölüm 3'e geri dönün. Tamamen anlayana kadar tekrar tekrar okuyun.*

NOT

Önceki çıktıyı kullanmaya devam edersek, interface'in bant genişliği nedir? FastEthernet olarak bilinen interface'in bant genişliği, 100000 Kbit'dir (Kbit, üç sıfır eklenmesi anlamına gelir). Bu 100 Mbps ya da FastEthernet demektir. Gigabit, 1000000 Kbps olmalıdır.

`show interface` komutunun verdiği en önemli istatistik, line çıktısı ve data-link protokol durumudur. Şayet çıktı, FastEthernet'in up, line protokolünün up olduğunu gösteriyorsa, interface up'tır ve çalışıyordur:

```
Router#sh int fa0/0
FastEthernet0/0 is up, line protocol is up
```

İlk parametre, Physical katmanı ve iletişim sinyali aldığında interface'in up olduğunu işaret eder. İkinci parametre, Data Link katmanını işaret eder ve bağlanan uçtan keepalive'lar arar (Keepalives, bağlantının kopmadığından emin olmak için cihazlar arasında kullanılmaktadır).

Aşağıdaki örnekte, seri interface'lerde, genelde problemin nerede olduğunu görebilirsiniz:

```
Router#sh int s0/0/0
Serial0/0 is up, line protocol is down
```

Şayet yukarıdaki gibi, line'ı up, fakat protokolü down görürseniz, bir saat denetimi veya framing problemi (muhtemelen enkapsülasyon uyumsuzluğu) yaşıyorsunuzdur. Onların eşleştiklerinden emin olmak için her iki uçta da keepalive'ları kontrol edin. Şayet gerekirse, clock rate'in ayarlandığından ve enkapsülasyon tiplerinin, her iki uçta aynı olduğundan emin olun. Yukarıdaki çıktı, Data Link katmanı problemi olarak kabul edilir.

Şayet hem line interface'i hem de protokolü down olarak görürseniz, bu bir kablo ya da interface problemidir. Aşağıdaki çıktı, Physical katman problemi olarak düşünülebilir:

```
Router#sh int s0/0/0
Serial0/0 is down, line protocol is down
```

Eğer bir uç, administratively shut down (sonraki örnekte görüldüğü gibi) ise, uzak uç, down down olarak görünecektir:

```
Router#sh int s0/0/0
Serial0/0 is administratively down, line protocol is down
```

Interface'i aktif hale getirmek için, interface configuration moddan no `shutdown` komutunu kullanın. `show interface serial 0/0/0` komutu, seri hattı ve varsayılan MTU'nun (maximum transmission unit) 1,500 byte olduğunu göstermektedir. Ayrıca, tüm Cisco seri linklerinin varsayılan bant genişliğini 1.544 Kbps olarak gösterir. Bu, EIGRP ve OSPF gibi routing protokolleri için hattın bant genişliğini belirlemek için kullanılmaktadır. Dikkat edilmesi gereken diğer önemli konfigürasyon, varsayılanda 10 saniye olan, keepalive'dır. Her router, komşusuna her 10 saniyede bir keepalive mesajı gönderir ve şayet her router aynı keepalive zamanı ile yapılandırılmamışsa, bu çalışmaz.

```
Router#sh int s0/0/0
Serial0/0 is up, line protocol is up
 Hardware is HD64570
 MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec,
   reliability 255/255, txload 1/255, rxload 1/255
 Encapsulation HDLC, loopback not set, keepalive set
  (10 sec)
 Last input never, output never, output hang never
 Last clearing of "show interface" counters never
 Queueing strategy: fifo
 Output queue 0/40, 0 drops; input queue 0/75, 0 drops
 5 minute input rate 0 bits/sec, 0 packets/sec
 5 minute output rate 0 bits/sec, 0 packets/sec
   0 packets input, 0 bytes, 0 no buffer
   Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
   0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored,
   0 abort
   0 packets output, 0 bytes, 0 underruns
   0 output errors, 0 collisions, 16 interface resets
   0 output buffer failures, 0 output buffers swapped out
   0 carrier transitions
   DCD=down DSR=down DTR=down RTS=down CTS=down
```

Interface'teki sayaçları, `clear counters` yazarak temizleyebilirsiniz:

```
Router#clear counters ?
  Async              Async interface
  BVI                Bridge-Group Virtual Interface
  CTunnel            CTunnel interface
  Dialer             Dialer interface
  FastEthernet       FastEthernet IEEE 802.3
  Group-Async        Async Group interface
```

```
  Line               Terminal line
  Loopback           Loopback interface
  MFR                Multilink Frame Relay bundle interface
  Multilink          Multilink-group interface
  Null               Null interface
  Serial             Serial
  Tunnel             Tunnel interface
  Vif                PGM Multicast Host interface
  Virtual-Template   Virtual Template interface
  Virtual-TokenRing  Virtual TokenRing
  <cr>

Router#clear counters s0/0/0
Clear “show interface” counters on this interface
  [confirm][enter]
Router#
00:17:35: %CLEAR-5-COUNTERS: Clear counter on interface
  Serial0/0/0 by console
Router#
```

### show ip interface Komutu ile Doğrulamak

`show ip interface` komutu, bir router interface’nin katman3 konfigürasyonları hakkında bilgileri sağlar:

```
Router#sh ip interface
FastEthernet0/0 is up, line protocol is up
  Internet address is 1.1.1.1/24
  Broadcast address is 255.255.255.255
  Address determined by setup command
  MTU is 1500 bytes
  Helper address is not set
  Directed broadcast forwarding is disabled
  Outgoing access list is not set
  Inbound  access list is not set
  Proxy ARP is enabled
  Security level is default
  Split horizon is enabled
[output cut]
```

Interface’in durumu, IP adresi ve maskı, interface’e bir access list uygulanıp uygulanmadığı bilgisi ve temel IP bilgileri, bu çıktıda yer almaktadır.

### show ip interface brief Komutunu Kullanmak

`show ip interface brief` komutu, muhtemelen, bir Cisco router’da kullanabileceğiniz en faydalı komutlardan biridir. Bu komut, mantıksal adres ve durumunu içererek, router interface’lerine hızlı bir göz atma sağlar:

```
Router#sh ip int brief
Interface          IP-Address      OK? Method Status  Protocol
FastEthernet0/0    unassigned      YES unset  up      up
FastEthernet0/1    unassigned      YES unset  up      up
Serial0/0/0        unassigned      YES unset  up      down
Serial0/0/1        unassigned      YES unset  administratively
down down
Serial0/1/0        unassigned      YES unset  administratively
down down
Serial0/2/0        unassigned      YES unset  administratively
down down
```

Administratively down'ın, interface altında no shutdown yazmanız gerektiği anlamına geldiğini hatırlayın. Serial0/0/0 'ın up/down olduğuna dikkat edin. Bunun anlamı, physical katmanın iyi olduğu ve bağlantı sinyalinin algılandığı, ancak uzak uçtan, keepalive'ların alınmadığıdır. Benim çalıştığım gibi bir lab ortamında, clock rate ayarlanmaz.

## show protocols Komutu ile Doğrulamak

show protocols komutu, hem her interface'in katman 1 ve katman 2 durumlarına hem de kullanılan IP adreslere hızlıca bakmak için gerçekten faydalıdır.

Aşağıda, gerçek bir ağda çalışan router'larıma bir bakalım:

```
Router#sh protocols
Global values:
  Internet Protocol routing is enabled
Ethernet0/0 is administratively down, line protocol is down
Serial0/0 is up, line protocol is up
  Internet address is 100.30.31.5/24
Serial0/1 is administratively down, line protocol is down
Serial0/2 is up, line protocol is up
  Internet address is 100.50.31.2/24
Loopback0 is up, line protocol is up
  Internet address is 100.20.31.1/24
```

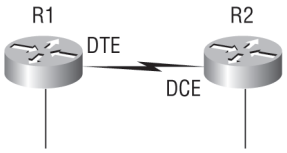
## show controllers Komutu Kullanmak

show controlers komutu, fiziksel interface'in kendisi ile ilgili bilgileri görüntüler. O, ayrıca seri port'a bağlı seri kablo tipini de gösterir. Genelde, bu sadece, bir DSU (data service unit) türüne bağlı DTE kablodur.

```
Router#sh controllers serial 0/0
HD unit 0, idb = 0x1229E4, driver structure at 0x127E70
buffer size 1524 HD unit 0, V.35 DTE cable
cpb = 0xE2, eda = 0x4140, cda = 0x4000

Router#sh controllers serial 0/1
HD unit 1, idb = 0x12C174, driver structure at 0x131600
buffer size 1524 HD unit 1, V.35 DCE cable
cpb = 0xE3, eda = 0x2940, cda = 0x2800
```

serial0/1, bir DCE kabloya bağlıyken, serial0/0 'ın DTE kabloya sahip olduğuna dikkat edin. Serial0/1, `clock rate` komutuyla saat denetimi sağlamak zorundadır. Serial0/0, saat denetimini, DSU'dan alacaktır.

Gelin bu komuta bir kez daha bakalım. Şekil4.5 'te, iki router arasındaki DTE/DCE kabloyu gördünüz mü? Bunu gerçek ağlarda görmeyeceğinizi bilin!



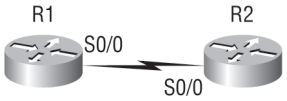

***Şekil 4.6:*** *show controllers komutu.*

Router R1, bir DTE bağlantısına sahiptir (tüm Cisco router'lar için varsayılandır). Router R1 ve R2 bağlantı kuramazlar. Aşağıda, `show controllers s0/0` çıktısına bir bakalım:

```
R1#sh controllers serial 0/0
HD unit 0, idb = 0x1229E4, driver structure at 0x127E70
buffer size 1524 HD unit 0, V.35 DCE cable
cpb = 0xE2, eda = 0x4140, cda = 0x4000
```

`show controllers s0/0` komutu, interface'in V.35 DCE kablo olduğunu göstermektedir. Yani, R1, router R2 'ye hattın saat denetimini sağlaması gerekir. Aslında, interface, R1 router'ın seri interface'indeki kabloda yanlış etikete sahiptir. Fakat R1 router'ın seri interface'inde saat denetimi eklerseniz, network çalışacaktır.

Gelin şekil 4.6 'da görünen, `show controllers` komutunu kullanarak çözebileceğiniz bir diğer soruna bakalım. Tekrar belirmeliyim ki, router R1 ve R2 haberleşemezler.




***Şekil 4.7:*** *show ip interface ile kullanılan show controllers komutu.*

Aşağıda R1'in `show controllers s0/0` ve `show ip interface s0/0` komutlarının çıktıları vardır:

```
R1#sh controllers s0/0
HD unit 0, idb = 0x1229E4, driver structure at 0x127E70
buffer size 1524 HD unit 0,
DTE V.35 clocks stopped
cpb = 0xE2, eda = 0x4140, cda = 0x4000

R1#sh ip interface s0/0
Serial0/0 is up, line protocol is down
  Internet address is 192.168.10.2/24
  Broadcast address is 255.255.255.255
```

Şayet, `show controllers` ve `show ip interfaces` komutlarını kullanırsanız, router R1 'in hattın saat denetimini alamadığını göreceksiniz. Bu network, gerçek hayatta kullanılmayan (lab ortamında olan) bir ağdır, bu nedenle hattın saat denetimini sağlaması için bir CSU/DSU bağlanmamıştır. Yani, kablonun DCE ucu (bu örnekte R2 router'ı), saat denetimi sağlayacaktır. `show ip interface`, interface'in up olduğunu, fakat protokolün down olduğunu belirtmektedir. Bu, uzak uçtan keepalive paketlerinin alınmadığı anlamına gelir. Bu örnekte, muhtemel sebep, bozuk kablo olması ya da saat denetimi olmamasıdır.

# Cisco Security Device Manager (SDM)

Son olarak Cisco SDM hakkında bahsedeceğimiz bölüme geldik. Bu araç, bir router'ı, HTTP veya HTTPS interface'lerinden yapılandırmamıza yardımcı olması için kullanılmaktadır. Doğruyu söylemek gerekirse, Cisco, geçmişte bunun gibi bazı araçlar geliştirdi, fakat bunun gibi iyi çalışmadı.

SDM gerçekten iyidir. SDM, Cisco 830 serisinden, 7301 serisine kadar Cisco router'larda kullanılabilmektedir. Artı, tüm yeni 850, 870, 1800, 2800 ve 3800 serisi router'larda kurulu gelmektedir.

Fakat bilinmesi gereken bir nokta vardır. SDM, aslında çok kullanışlı bir araç olduğu halde, sadece detaylı konfigürasyonlarda kullanılması iyidir (bu modülde kullandığımız örneklerdeki gibi basit yapılandırmalar için değil). Şöyle açıklayabilirim; gelişmiş bir access list, IPSec ile VPN ve router'ınızda intrusion koruması kurarken kullanırsınız. Konfigürasyonu ve IPSec'in ne olduğunu bilmek zorunda bile değilsiniz.

NOT *Cisco PIX ürünleri için, SDM değil de, Pix Device Manager (PDM) kullanın.*

Fakat burada hem iyi hem de kötü bir durum vardır. Tabii ki, artık detaylı konfigürasyonları çok daha kolay halledebiliriz, fakat aynı zamanda, bu o kadar güzel değil, çünkü artık herkes aynı şeyi yapabilir!

SDM kullanarak, nasıl oturum açılacağını göstereceğim. Kullanıcı adınızı, banner ve enable secret password'ünü ayarlayın, bir router'da DHCP pool'u ve bir interface'e IP adresi tanımlayın. Herşey yolunda giderse, bu bölümden sonra, SDM'in çok daha iyi olduğunu göreceksiniz. Çünkü SDM, IOS, NAT, kablosuz teknolojiler ve güvenlikle ilgili modülleri okuduğunuzda, basit IOS router konfigürasyonlarında başarısız olduğumuzda kullandığımız bir yöntem olmaktan ziyade, size çok daha fazlasını sağlayacağını kanıtlayacaktır. Onu, sahip olacağınız kolaylıkları göstermek ve karmaşıklığını açıklamak için her modülde biraz kullanacağım. Ayrıca SDM'in eksikliklerini de ortaya koyacağım.

Sadece SDM hakkında bir kitap yazabilirim, fakat bunu yapmama gerek yok, çünkü Cisco zaten bunu yaptı. SDM hakkında daha fazla detaya `www.cisco.com/go/sdm` adresinden ulaşabilirsiniz. Artı, yeni bir router, genellikle, fiziksel olarak router'a bağlanıp onu adım adım yapılandırmanızı anlatacak bir CD ile gelecektir. Fakat router'ınıza bağlanmak ve SDM kullanmak için bu CD'ye gerçekten ihtiyacınız yoktur.

Tüm ihtiyacınız, desteklenmiş bir ISR router (1800/2800 vs.) 'dır ve SDM'in son versiyonunu, kurulum açıklamaları ile birlikte, `www.cisco.com/pcgi-bin/tablebuild.pl/sdm` adresinden indirebilirsiniz.

NOT *Bu yazılımı indirmek için, bir Cisco Connection Online (CCO) oturumu başlatmanız gerekmektedir. www.cisco.com adresine gidin ve sağ-üst köşedeki Register butonuna tıklayın. Kısa formu doldurun, kullanmak istediğiniz kullanıcı adı ve şifrenizi ekleyin. Şimdi SDM ve SDM demosunu indirebilirsiniz.*

Bu adresten,Cisco SDM demo'sunun kullanılmasını da etkinleştirebilirsiniz.

NOT *Çalışmak için bir ISR router'a sahip değilseniz, SDM demosunu indirmenizi ve çalışmanızı tavsiye ederim. İlk olarak, bilgisayarınıza SDM'i kurun ve sonra www.cisco.com/pcgi-bin/tablebuild.pl/sdm-tool-demo adresindeki demoyu indirin. Kurulum direktiflerini okuyun veya kuruluma bakmak ve SDM konusunda hazır olmak için bu bölümün sonundaki Pratik Lab 4.6'ya bakın.*

Şunu da bilmeniz gerekir ki; SDM kullanarak, host'unuzun oturum açması için ilk olarak router'unuzun yapılandırıldığından emin olmalısınız. Bundan önceki bölümde, konfigürasyonumu silmiş ve router'u reload etmiştim, bu nedenle sıfırdan başlamak zorundayım. Fakat bunu yapmak gerçekten zor değil. Sadece router'ın bir LAN interface'ini seçip çapraz bir kablo kullanarak host'u direkt olarak router'a bağlayabilirsiniz.

```
Would you like to enter the initial configuration dialog? [yes/
no]: n
Press RETURN to get started!
Router>en
Router#config t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#int f0/0
Router(config-if)#ip address 1.1.1.1 255.255.255.0
```

```
Router(config-if)#no shut
Router(config-if)#do ping 1.1.1.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 1.1.1.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/4
ms
```

Peki, bunlar ne içindir? Yukarıdaki konfigürasyonda, FastEthernet interface'ine bir IP adresi verdim ve onu no shutdown komutu ile etkinleştirdim. Sonra, router komut istemcisinden, direkt bağlı olduğum host'a ping atarak bağlantımı kontrol ettim. (Bu, minimum bir konfigürasyon'dur ve SDM yoluyla bağlanmanıza izin verir.) Buradan, pop-up'ları etkinleştirerek browser 'ı açın, **http://1.1.1.1** yazın ve bağlanınca kolay adımları takip edin.

Şayet router'ı HTTPS kullanması için ayarlamak isterseniz durum değişir. Bu, bağlantıda, privileged moda erişmenizi sağlar (bu, router'ı, orijinal varsayılan konfigürasyonuna döndürüyoruz anlamına gelir). Biraz daha fazla komut eklemeye ihtiyacınız var.

İlk olarak, HTTP/HTTPS sunucusunu etkinleştirin (şayet gelişmiş IOS servisleri yoksa router'ınız HTTPS'i desteklemeyecektir):

```
Router(config)#ip http server
Router(config)#ip http secure-server
% Generating 1024 bit RSA keys, keys will be non-exportable...[OK]
Router(config)#ip http authentication local
```

İkinci olarak, privileged level 15 (bu en üst seviyedir) kullanarak, bir kullanıcı hesabı oluşturun:

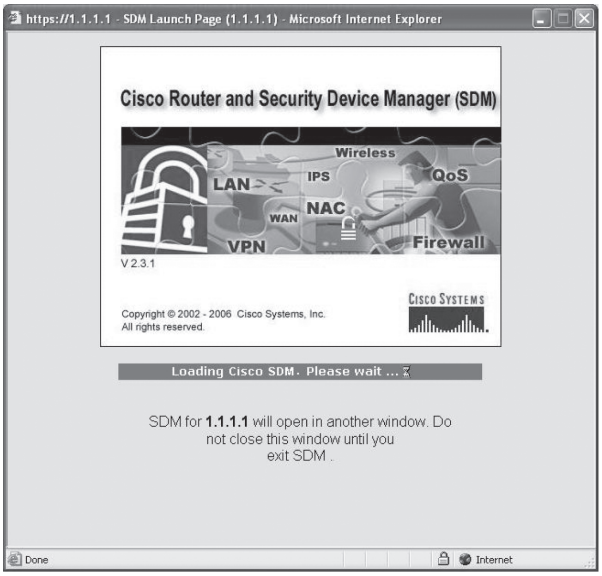
```
 Router(config)#username cisco privilege ?
  <0-15>  User privilege level

Router(config)#username cisco privilege 15 password ?
  0     Specifies an UNENCRYPTED password will follow
  7     Specifies a HIDDEN password will follow
  LINE  The UNENCRYPTED (cleartext) user password
Router(config)#username cisco privilege 15 password 0 cisco
```

Son olarak, privileged level erişimde yerel oturum açma kimlik denetimi sağlamak için konsol, SSH ve Telnet'i yapılandırın:

```
Router(config)#line console 0
Router(config-line)#login local
Router(config-line)#exit
Router(config)#line vty 0 ?
  <1-1180>  Last Line number
  <cr>
Router(config)#line vty 0 1180
Router(config-line)#privilege level 15
Router(config-line)#login local
Router(config-line)#transport input telnet
Router(config-line)#transport input telnet ssh
Router(config-line)#^Z
```

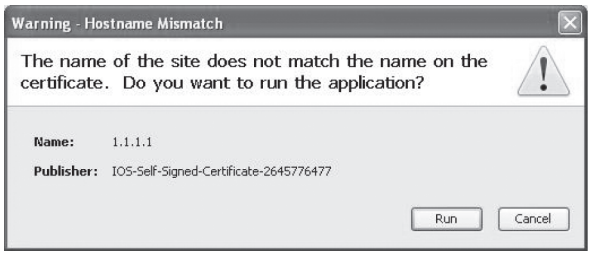
`https://1.1.1.1` yardımıyla bağlanır bağlanmaz, bir güvenlik alarmı mesajı aldık.


Sonra, oluşturulan kullanıcı adı/ şifre ile oturum açtım. SDM yüklenmeye başladı ve bana beklememi söyledi (başka pencerede yüklenmesi için biraz zamana ihtiyacı olduğu anlamına gelir). Bu pencereyi sakın kapatmayın.

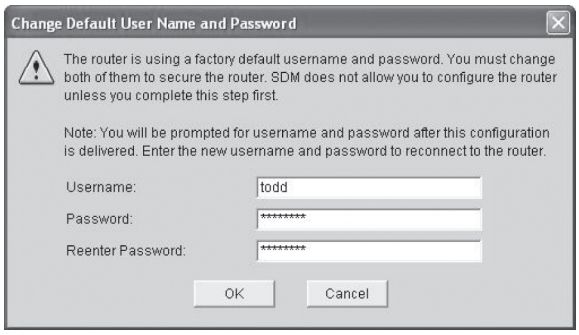



`ip http-secure-server` komutu ile oluşturduğum sertifika, router'a yüklendi. Always Trust Content from this Publisher seçeneğini tıklayıp, Yes'i seçtim.

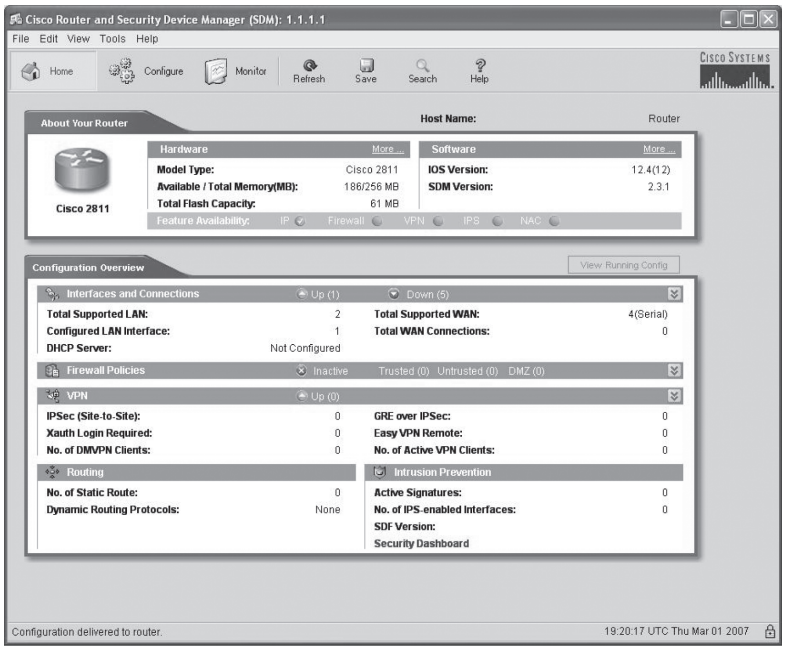

Sertifika herhangi bir site adıyla eşleşmeyeceğinden, çalışması için doğrulamak zorunda kaldım.



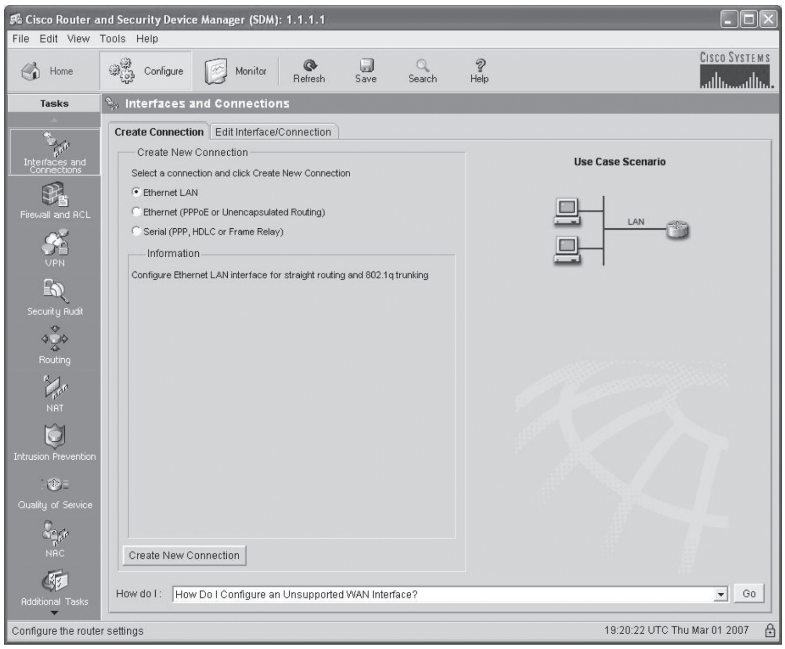

Sonra, tekrar oturum açtım ve router benim varsayılan kullanıcı adı ve şifremi kontrol ederken, SDM'nin yüklenmesini bekledim.



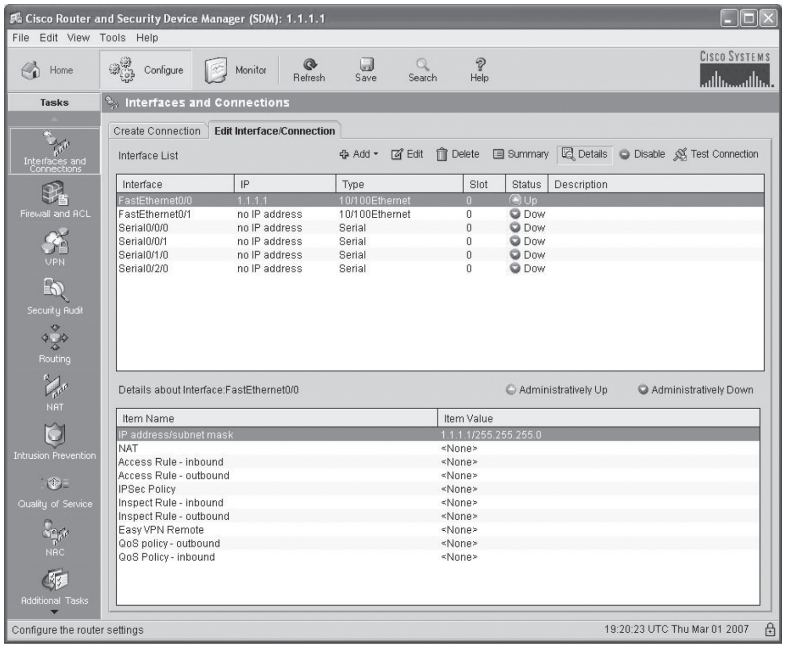

Sonunda SDM'ye bağlandım!



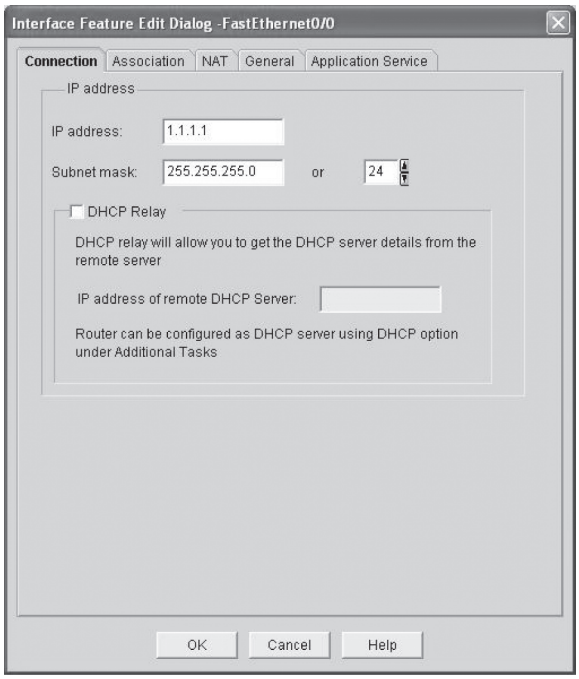

Sayfanın başındaki Configure butonuna tıklayarak, yapılandırmak istediğim interface tipini seçerek, adım adım interface konfigürasyonunu seçtim ve sonra sayfanın altındaki Create New Connection butonuna tıkladım. Bu, yapılandırmak için seçtiğiniz interface'e bağlı olarak, LAN veya WAN sihirbanızı açar.(Router interface'lerimizi, bölüm 6'da Interface Wizard ile yapılandıracağız).



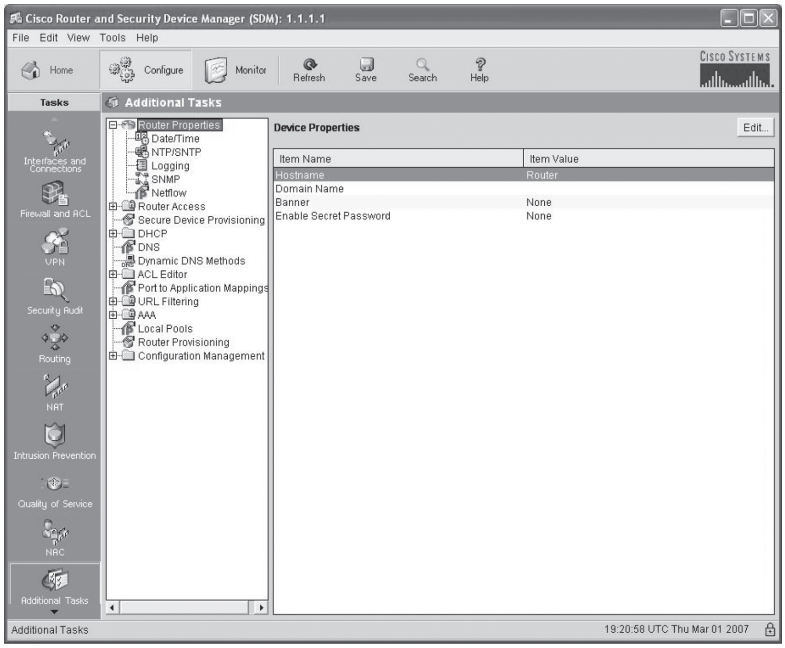

Edit Interface/Connection sekmesine tıklayarak, interface'inizin durumunu görebilirsiniz.



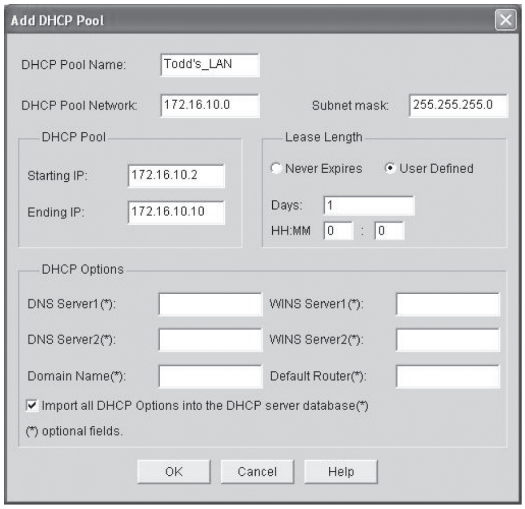

Hepsi bu değil, düzenlemek için bir interface üzerine çift-tıklayın (bunu sadece, LAN veya WAN sihirbazına gittikten ve interface'i yapılandırdıktan sonra yapabilirsiniz).




Sihirbazın sol altındaki bölüme bakın ve Additional Tasks butonuna tıklayın. Buradan, Router Properties simgesine tıklayın.




Burada, hostname, MOTD banner ve enable secret password'ü oluşturabilirsiniz. Son olarak, DHCP klasörünü tıkladım, sonrada DHCP pool simgesini seçtim. Add butonuna tıkladım ve router'ımda bir DHCP havuzu oluşturdum.




Şimdi, gelin router'daki konfigürasyona bir bakalım:

```
Todd#sh run
Building configuration...
[output cut]
hostname Todd
!
ip domain name lammle.com
[output cut]
ip dhcp excluded-address 172.16.10.1
ip dhcp excluded-address 172.16.10.11 172.16.10.254
!
ip dhcp pool Todd's_LAN
   import all
   network 172.16.10.0 255.255.255.0
!
crypto pki trustpoint TP-self-signed-2645776477
 enrollment selfsigned
 subject-name cn=IOS-Self-Signed-Certificate-2645776477
 revocation-check none
 rsakeypair TP-self-signed-2645776477
!
crypto pki certificate chain TP-self-signed-2645776477
 certificate self-signed 01
  3082023E 308201A7 A0030201 02020101 300D0609 2A864886 F70D0101
   04050030 31312F30 2D060355 04031326 494F532D 53656C66
2D536967
   6E65642D 43657274 69666963 6174652D 32363435 37373634
3737301E
   170D3037 30333031 3139313 33335A17 0D323030 31303130 30303030
   305A3031 312F302D 06035504 03132649 4F532D53 656C662D
5369676E
```

```
   65642D43 65727469 66696361 74652D32 36343537 37363437
3730819F
   300D0609 2A864886 F70D0101 01050003 818D0030 81890281
8100BB24
   [output cut]
  quit
username todd privilege 15 secret 5 $1$nvgs$QRNCWKJ7rfmtNNkD2xvG
q/
[output cut]
!
line con 0
 login local
line aux 0
line vty 0 4
 privilege level 15
 login local
 transport input telnet ssh
line vty 5 1180
 privilege level 15
 login local
 transport input telnet ssh
!
```

Burada görebileceklerimiz router'ın, hostname, DHCP pool ve sertifika oluşturduğudur. HTTPS kullandığımızdan, daha fazla ayar yaptık. Sadece HTTP kullanmak için kurulum yapmak hem daha kolay hem de zahmetsizdir. Fakat bizim yaptığımızın, SDM kullanarak bir ISR varsayılan konfigürasyonu olduğunu unutmayın. Bu kitap boyunca SDM kullanmaya devam edeceğiz, fakat ben kendiniz için SDM'i kurmanızı ve ona alışmanızı tavsiye ederim.

# Özet

Bu eğlenceli bir bölümdü, değil mi? Size bazı Cisco IOS bilgileri verdim ve gerçekten, Cisco router dünyasıyla ilgili birçok yeni bakış kazandığınızı umuyorum. Bu bölüm, Cisco Internetwork Operating System'i (IOS), IOS'u nasıl kullanacağınızı ve Cisco router'ları nasıl yapılandıracağınızı açıklayarak başladı. Router'ın nasıl aktif edildiğini ve setup modun ne olduğunu öğrendiniz. Bu arada, Cisco router'ları basit olarak yapılandırabileceğinizden, setup modu asla kullanmazsınız, değil mi?

Bir router'a, konsol ve LAN bağlantısıyla nasıl bağlanacağını tartıştıktan sonra, Cisco yardım özelliklerini ve komutlar ile komut parametrelerinin nasıl kullanılacağını işledim. İlave olarak, konfigürasyonlarınızı doğrulamak için bazı temel show komutlarını gösterdim.

Bir router'daki yönetimsel fonksiyonlar, ağınızı yönetmek ve doğru cihazı yapılandırdığınızı doğrulamak için size yardımcı olur. Router şifresi ayarlamak, router'ınızda çalıştırabileceğiniz en önemli konfigürasyonlardan biridir. Size, ayarlanması için beş şifre gösterdim. Ek olarak, router'ınızı yönetmenizi kolaylaştırmak için,hostname, interface description ve banner'ları kullandım.

Son olarak, Secure Device Manager kullanarak bağlanabilmeniz ve konfigürasyon için router'ınızı nasıl yapılandıracağınızı gösterdim. Tekrar ediyorum ki, temel router fonksiyonlarını yapılandırmak için CLI kullanmak tamamıyla daha kolaydır, fakat yakında göreceğiniz daha gelişmiş konfigürasyonlarda, SDM'in gerçekten nasıl yardımcı olabileceğini size göstereceğim.

Bu, Cisco IOS'a girişinizi sonlandırır. Ve her zamanki gibi, sonraki bölümlere geçmeden önce, bu modülde incelenen temel bilgilere sahip olmanız çok önemlidir.

# Sınav Gereklilikleri

**Bir router'ı power düğmesinden açtığınızda ne olduğunu (ve sırasını) anlamak:** Bir Cisco router'ı ilk olarak aktifleştirdiğinizde, bir power-on-self-test (POST) çalıştıracaktır ve şayet bunu geçerse, flash bellekten, Cisco IOS'a bakacak ve dosya varsa, yükleyecektir. Sonra, IOS yüklenmesine devam eder ve NVRAM'de, startup-config olarak bilinen, geçerli bir konfigürasyon bakar. Şayet NVRAM'de bir dosya yoksa router setup moda geçecektir.

**Setup modun ne sağladığını hatırlamak:** Setup mod, şayet bir router boot eder ve NVRAM'de startup-config yoksa otomatik olarak başlatılır. Siz ayrıca, privileged moddan setup yazarak `setup` modunu etkinleştirebilirsiniz. Setup, komut satırından bir Cisco router'ı nasıl yapılandıracağını bilmeyen biri için kolay bir formatta, minimum konfigürasyon sağlar.

**User mod ve privileged mod arasındaki farkları anlamak:** User mod, varsayılan olarak, az sayıda komutla, bir command-line interface (CLI) sağlar. User mod, konfigürasyonun, bakılıp, değiştirilmesine izin vermez. Privileged mod, bir kullanıcının, router'ın yapılandırmasına bakıp, değiştirmesine izin verir. `enable` komutunu yazarak ve şayet ayarlandıysa, enable password ve enable secret password girerek, privileged moda geçebilirsiniz.

Show version **komutunun ne sağladığını hatırlamak:** `show version` komutu, hem sistem donanımı hem de yazılım versiyonu, konfigürasyon dosyalarının isimleri ile kaynakları, `config-register` ayarları ve boot imajları için temel yapılandırma bilgilerini sağlayacaktır.

**Bir router'ın hostname'inin nasıl ayarlandığını hatırlamak:** Bir router'a hostname ayarlamak için komut sırası şöyledir:

```
enable
config t
hostname Todd
```

`enable password` **ve** `enable secret password` **arasındaki farkı hatırlamak:** Bu şifrelerin ikisi de privileged moda erişim sağlamak için kullanılır. Bununla beraber, enable secret password, daha yenidir ve varsayılan olarak daima şifrelidir. Ayrıca, şayet enable password oluşturup, sonrada enable secret password ayarlarsanız, sadece enable secret kullanılacaktır.

**Bir router'da enable secret 'ın nasıl ayarlandığını hatırlamak:** Enable secret ayarlamak için enable secret komutunu kullanın. `Enable secret password` ***password*** kullanmayın ya da şifrenizi, `password` ***password*** şeklinde ayarlayın. Aşağıda bir örnek vardır:

```
enable
config t
enable secret todd
```

**Bir router'da konsol şifresinin nasıl ayarlandığını hatırlamak:** Konsol şifresi oluşturmak için sıralama şöyledir:

```
enable
config t
line console 0
login
password todd
```

**Bir router'da Telnet şifresinin nasıl ayarlandığını hatırlamak:** Telnet şifresi oluşturmak için sıralama şöyledir:

```
enable
config t
line vty 0 4
password todd
login
```

**Bir seri link probleminin nasıl tespit edildiğini anlamak:** Şayet `show interface serial 0` yazdınız ve down, line protocol is down görürseniz, bu bir Fiziksel katman problemi olarak düşünülür. Şayet up, line protocol is down olarak görürseniz, o zaman, bu bir Data Link katmanı problemidir.

Show interfaces **komutu ile router'ınızı nasıl doğrulayacağınızı anlamak:** Şayet `show interfaces` yazarsanız, router'daki interface'ler için istatistiklere bakabilir, interface'in down olup olmadığını doğrulayabilir ve her interface'deki IP adreslerini görebilirsiniz.

# Yazılı Lab 4

Aşağıdaki sorular için komut veya komutları yazın:

1. Bir seri interface'e, başka router'a 64Kb saat denetimi sağlaması için hangi komut kullanılmaktadır?
2. Şayet bir router'a telnet yaparsanız ve `connection refused, password not set` cevabı alırsanız, bu mesajı almayı ve bir şifre istenmesini durdurmak için hedef router'da ne yaparsınız?
3. Şayet `show inter et 0` yazarsanız ve port'un administravely down olduğunu fark ederseniz, ne yaparsınız?
4. NVRAM'de tutulan konfigürasyonu silmek isterseniz, ne yazarsınız?
5. Konsol port'u için bir user-mode şifresi oluşturmayı isterseniz, ne yazarsınız?
6. Enable secret password'ünüzü cisco olarak ayarlamak isterseniz, ne yazarsınız?
7. Bir seri interface'in saat denetimi sağlamaya ihtiyacı olup olmadığını anlamak için, hangi komutu kullanırsınız?
8. Terminal history boyutunu görmek için hangi komutu kullanırsınız?
9. Router'ı yeniden başlatmak ve running-config'i mevcut startup-config 'le değiştirmek istiyorsunuz. Hangi komutu kullanacaksınız?
10. Router'ın adını Chicago olarak nasıl ayarlarsınız?

(Yazılı lab4'ün cevapları, bu bölüm için gözden geçirme sorularına cevaptan sonra bulunabilir.)

# Pratik Lab'lar

Bu bölümde, bir Cisco router'da, bu modülde öğrendiklerinizi anlamanıza yardımcı olacak komutları çalıştıracaksınız.

En az bir router'a ihtiyacınız olacak (iki tane olması daha iyi, üç adet olması mükemmel olur). Bu bölümdeki pratik lab'lar, gerçek Cisco router'ları kullanmayı içermektedir. Şayet RouterSim.com ya da Sybex 'ten yazılım kullanıyorsanız, lütfen, bu programlarda bulunan pratik lab'ları kullanın. Bu lab'larda hangi tip router serisi kullandığınızın bir önemi yoktur.(2500, 2600, 800 veya 2800).

Bu bölüm aşağıdaki altı lab'ı içermektedir:

Lab 4.1: Bir Router'a bağlanmak

Lab 4.2: Help ve Editing Özelliklerini Kullanmak.

Lab 4.3: Bir Router Konfigürasyonunu kaydetmek.

Lab 4.4: Şifrelerinizi Ayarlamak.

Lab 4.5: Hostname, Description, IP Adresi ve Saat denetimi ayarlamak.

Lab 4.6: Bilgisayarınıza SDM Yüklemek.

## Pratik Lab 4.1: Bir Routera Bağlanmak

1. Router'ınıza bağlanmak için Enter tuşuna basın. Bu sizi, user moda götürecektir.
2. `Router>` satırında, bir soru işareti (?) yazın.
3. Ekranın altındaki –`more -`'a dikkat edin.
4. Komutları satır satır görmek için Enter tuşuna basın. Bir defada komutları tüm ekranda görmek için spacebar'a basın. İstediğiniz zaman q yazarak çıkabilirsiniz.
5. `enable` ya da `en` yazın ve Enter tuşuna basın. Bu, router konfigürasyonlarına bakıp değiştirebileceğiniz privileged moda geçirecektir.
6. `Router#` istemcisinde, bir soru işareti (?) yazın. Privileged modda ne kadar seçeneğiniz olduğuna dikkat edin.
7. Çıkmak için q yazın.
8. `config` yazın ve Enter tuşuna basın.
9. Terminalinizi kullanarak router'ınızı yapılandırmak için Enter tuşuna basın.
10. `Router(config)#` istemcisinde, soru işareti (?) yazın, sonra q yazıp çıkın veya komutları görmek için spacebar'a dokunun.
11. `interface e0` veya `int 0` (veya `int fa0/0`) yazın ve Enter 'a basın. Bu, interface Ethernet 0 'ı yapılandırmanızı sağlar.
12. Router(config-if)# satırında, soru işareti (?) yazın.
13. `int s0` (`int s0/0`) veya `interface s0` (`interface serial 0` komutu ile aynı) yazın ve Enter tuşuna basın. Bu, interface serial0'ı yapılandırmanızı sağlar. Interface'den interface'e kolayca geçebileceğinize dikkat edin.
14. `encapsulation?` yazın.
15. `exit` yazın. Bunun sizi bir önceki seviyeye nasıl getirdiğine dikkat edin.
16. Ctrl+Z tuş bileşimine basın. Bunun sizi configuration mod dışına atıp privileged moda nasıl getireceğine dikkat edin.
17. `disable` yazın. Bu sizi user moda götürecektir.
18. `exit` yazın. Bu, router oturumunuzu kapatacaktır.

## Pratik Lab 4.2: Help ve Editing Özelliklerini Kullanmak

1. Router'a bağlanın ve `en` ya da `enable` yazarak privileged moda gidin.
2. Bir soru işareti (?) yazın.
3. `cl?` yazın ve Enter tuşuna basın. cl ile başlayan tüm komutları görebileceğinize dikkat edin.
4. `clock ?` yazın ve Enter tuşuna basın.

**NOT** *3. ve 4. adımlar arasındaki farka dikkat edin. 3. adımda, harfleri yazıp arada boşluk bırakmadan soru işareti koyuyorsunuz. Bu size, cl ile başlayan tüm komutları verecektir. 4. adımda, bir komut yazıp boşluk bırakıp soru işareti yazılıyor. Bunu yaparak, sonraki mümkün parametreyi görürsünüz.*

5. Clock ? yazarak ve help ekranlarını izleyerek ve router'ın zaman ve tarihini ayarlayarak, router'ın saatini ayarlayın.
6. clock ? yazın.
7. clock set ? yazın.
8. clock set 10:30:30 ? yazın.
9. clock set 10:30:30 14 March? yazın.
10. clock set 10:30:30 14 March 2002? Yazın.
11. Enter 'a basın.
12. Zaman ve tarihe bakmak için show clock yazın.
13. Privileged moddan, show access-list 10 yazın. Enter'a basmayın.
14. Ctrl+A tuş bileşimine basın. Bu, sizi satırın başına götürecektir.
15. Ctrl+E yazın. Bu sizi, tekrar satırın sonuna götürecektir.
16. Ctrl+A sonra da Ctrl+F tuşlarına basın. Bu sizi, bir karakter ileri götürür.
17. Ctrl+B tuş bileşimine basın. Bir karakter geriye gidersiniz.
18. Enter daha sonra da Ctrl+P tuş bileşimine basın. Bu, son komutu tekrarlayacaktır.
19. Klavyenizdeki üst ok tuşuna basın. Bu da son komutu tekrarlayacaktır.
20. `sh history` yazın. Bu, size girilen son 10 komutu verecektir.
21. `terminal history size ?` yazın. Bu, history giriş boyutunu değiştirir. ?, kabul edilen satırların sayısıdır.
22. Terminal istatistikleri ve history boyutu için `show terminal` yazın.
23. `terminal no editing` yazın. Bu, gelişmiş düzenlemeyi kapatır. `terminal editing` yazana kadar, düzenleme anahtarlarının kısa yollarının etkisi olmadığını görmek için 14'ten 18'e kadar adımları tekrarlayın.
24. `terminal editing` yazın ve gelişmiş düzenlemeler için Enter tuşuna basın.
25. `sh run` yazın ve Tab tuşuna basın. Bu, sizin için komutun tamamlanmasını sağlayacaktır.
26. `sh start` yazın ve Tab tuşuna basın. Bu, sizin için komutun tamamlanmasını sağlayacaktır.

## Pratik Lab 4.3: Bir Router Konfigürasyonunu Kaydetmek

1. Router'a bağlanın ve `en` ya da `enable` yazıp Enter tuşuna basarak privileged moda gidin.
2. NVRAM'de tutulan konfigürasyonu görmek için `sh start` yazıp Tab tuşuna ve sonra da Enter tuşuna basın ya da `show startup-config` yazıp Enter'a basın. Bununla beraber, konfigürasyon kaydedilmediyse, bir hata mesajı alacaksınız.
3. Konfigürasyonu, startup-config olarak bilinen NVRAM'e kaydetmek için aşağıdakilerden birini yapabilirsiniz:
   - `copy run start` yazın ve Enter 'a basın
   - `copy running` yazıp Tab tuşuna basın, start yazıp Tab'a ve sonra Enter'a basın.
   - `copy running-config startup-config` yazın ve Enter'a basın.
4. `sh star` yazıp Tab tuşuna ve sonra da Enter'a basın.
5. `sh run` yazıp Tab tuşuna ve sonra da Enter'a basın.
6. `erase star` yazıp Tab 'e ve sonra da Enter'a basın.

7. `sh star` yazıp Tab tuşuna ve sonrada Enter'a basın. Bir hata mesajı almalısınız.
8. `reload` yazın ve Enter'a basın. Enter'a basarak reload işlemini onaylayın. Router'ın reload olmasını bekleyin.
9. Setup moda girmemek için no deyin ya da sadece Ctrl+C tuş bileşimine basın.

## Pratik Lab 4.4: Şifrelerinizi Ayarlamak

1. Router'a bağlanın ve `en` ya da `enable` yazarak privileged moda girin.
2. `config t` yazın ve Enter 'a basın.
3. `enable ?` yazın.
4. `enable secret password` (üçüncü kelime sizin kendi şifreniz olmalıdır) yazarak enable secret şifrenizi ayarlayın ve Enter'a basın. Secret parametresinden sonra, password parametresini kullanmayın (bu sizin şifrenizi password olarak ayarlayacaktır). Örnek olarak, `enable secret todd` yazabilirsiniz.
5. Şimdi, her şekilde router'a bağlanıp, oturum açtığınızda ne olduğunu görelim. Ctrl+Z tuş bileşimine basarak oturumu kapatalım ve `exit` yazıp Enter'a basalım. Privileged moda girin. Priviliged moda kabul edilmeden önce, sizden şifre istenecektir. Şayet secret şifrenizi başarılı bir şekilde girerseniz, devam edebileceksiniz.
6. Secret şifrenizi silin. Privileged moda gidin, config t yazın ve Enter 'a basın. `no enable secret` yazın ve Enter 'a basın. Oturumu kapatıp, sonra tekrar oturum açın. Şimdi sizden şifre istenmemesi gerekir.
7. Privileged moda girmek için kullanılan bir şifre de, enable password olarak bilinir. Eski, daha güvensiz ve enable secret kullanıldığında kullanılmayan bir şifredir. Onu nasıl oluşturduğumuz ile ilgili örnek aşağıdadır:

```
config t
enable password todd1
```

8. Enable secret ve enable password'ün farklı olduklarına dikkat edin. Onlar aynı olamazlar.
9. Konsol ve auxiliary şifrelerini ayarlamada doğru seviyede olmak için `config t` ve sonra `line ?` yazın.
10. Line komutları için parametrelerin `auxiliary`, `vty` ve `console` olduğuna dikkat edin. Üçünü de ayarlayacaksınız.
11. Telnet ya da VTY şifresi oluşturmak için `line vty 0 4` yazın ve Enter 'a basın. 0 4, Telnet ile bağlanmak için kullanılan uygun sanal hatların 5 'inin aralığıdır. Şayet bir enterprise IOS 'a sahipseniz, bunların sayısı değişebilir. Router'ınızdaki uygun line sayısını anlamak için soru işaretini kullanın.

NOT *Telnet kullandığınızda, user-mode password ekranını iptal etmek için no login yazın*

12. Şimdiki komut, kimlik denetimini açıp kapatmak için kullanılmaktadır. `login` yazın ve router'a telnet yaptığınızda, bir user-mod şifresi istemcisi için Enter'a basın. Şayet şifre oluşturulmadıysa, router'a telnet yapamayacaksınız.
13. VTY şifreniz için ayarlamaya ihtiyacınız olan diğer bir komut, password 'dür. Şifre oluşturmak için `password` ***password*** yazın (`password`, sizin şifrenizdir.).
14. Aşağıda, VTY şifresinin nasıl ayarlandığıyla ilgili örnek bulabilirsiniz:

```
config t
line vty 0 4
login
password todd
```

15. İlk olarak, `line auxiliary 0` ya da `line aux 0` yazarak auxiliary şifrenizi ayarlayın.
16. `login` yazın.
17. `password` ***password*** yazın.
18. İlk olarak `line console 0` ya da `line con 0` yazarak, konsol şifrenizi ayarlayın.
19. login yazın.
20. `password` ***password*** yazın. Bu son iki komut için örnek aşağıdadır:

```
config t
line con 0
login
password todd1
line aux 0
login
password todd
```

21. `Exec-timeout 0 0` komutunu console 0 hattına ekleyebiliriz. Bu, zaman aşımı ile sizin oturumunuzu kapatarak konsolu bitirecektir. Komut şimdi şöyle olacaktır:

```
config t
line con 0
login
password todd2
exec-timeout 0 0
```

22. `logging synchronous` komutunu kullanarak konsol mesajlarının, yazdığınız komutun üzerine yazmaması için konsol satırını ayarlayın.

```
config t
line con 0
logging synchronous
```

## Pratik Lab 4.5: Hostname, Açıklamalar, IP Adresleri ve Clock Rate Ayarlamak.

1. Router'a bağlanın ve en ya da `enable` yazarak privileged moda girin.
2. hostname komutunu kullanarak, router'ınızda makine adını ayarlayın. Bunun tek kelime olduğuna dikkat edin. İşte bir örnek:

```
Router#config t
Router(config)#hostname RouterA
RouterA(config)#
```

   Enter 'a basar basmaz router'ın makine adının değiştiğine dikkat edin.

3. `banner` komutunu kullanarak, network yöneticilerinin göreceği bir banner oluşturun.
4. `config t` ve sonra `banner ?` yazın.
5. Dört farklı banner yazabileceğinize dikkat edin. Bu lab için, biz sadece, login ve message of the day (MOTD) banner'ları ile ilgileneceğiz.

```
config t
banner motd #
This is an motd banner
#
```

yazarak, MOTD banner ‘ınızı oluşturun. Bu, router’a bir konsol, auxiliary ya da Telnet bağlantısı yapıldığında görünecektir.

7. Yukarıdaki örnekte, ayırıcı bir karakter olarak # işareti kullanıldı. Bu, router’a, mesajın bittiğini söyler. Ayırıcı karakteri mesajın kendisinde kullanamazsınız.
8. MOTD banner ‘ını,

```
config t
no banner motd
```

yazarak silebilirsiniz.

9. login banner’ı

```
config t
banner login #
This is a login banner
#
```

yazarak oluşturabilirsiniz.

10. Login banner, MOTD ‘den hemen sonra, user-mode şifre satırından önce görüntülenecektir. User-mod şifrelerinizi, konsol, auxiliary ve VTY line şifreleri ayarlayarak oluşturduğunuzu hatırlayın.
11. login banner’ı

```
config t
no banner login
```

yazarak silebilirsiniz.

12. `ip address` komutu ile bir interface’e IP adresi ekleyebilirsiniz. İlk olarak, interface configuration moda girmeniz gerekmektedir. Bunu nasıl yaptığımızla ilgili örnek aşağıdadır:

```
config t
int e0 (you can use int Ethernet 0 too)
ip address 1.1.1.1 255.255.0.0
no shutdown
```

Bir interface’de IP adresi (1.1.1.1) ve subnet mask (255.255.0.0) ayarlandığına dikkat edin. no shutdown (ya da kısaca no shut), interface’i etkinleştirmek için kullanılmaktadır. Tüm interface’ler, varsayılan olarak kapalıdır.

13. `description` komutunu kullanarak, bir interface’e tanımlama ekleyebilirsiniz. Bu, bağlantılar hakkında bilgi eklemek için kullanışlıdır. Bunları sadece yöneticiler görebilir, kullanıcılar göremez. Burada bir örnek bulabilirsiniz:

```
config t
int s0
ip address 1.1.1.2 255.255.0.0
no shut
description Wan link to Miami
```

14. Bir DCE WAN linki simüle ettiğimizde, hem bir seri linkin bant genişliğini hem de saat hızı (clock rate) ekleyebiliriz. Örneğin:

```
config t
int s0
bandwidth 64
clock rate 64000
```

## Pratik Lab 4.6: Bilgisayarınıza SDM Kurulumu

Bu lab, bilgisayarınıza SDM programını indirip kurmanızı ve sonra buna demo programı eklemenizi sağlayacaktır. Bu lab'ın içerdiği linkler, bu yazının yazıldığı tarih itibarıyla geçerlidir, fakat onlar istenildiğinde değiştirilebilir.

1. Aşağıdaki Cisco'nun lokasyonundan, güncel SDM programını indirip yükleyebilirsiniz: `www.cisco.com/pcgi-bin/tablebuild.pl/sdm`.
2. Bilgisayarınıza SDM programını kurduktan sonra, SDM demo'sunu `www.cisco.com/pcgi-bin/tablebuild.pl/sdm-tool-demo` adresinden indirin.
3. Seçtiğiniz bir klasöre, demo programını unzip edin.
4. `dataFile.zip` dosyasını, `C:\` 'ye kopyalayın.
5. Browser'ınızda, pop-up 'ları bloklamayı kapattığınızdan emin olun.
6. Bir PC'ye SDM kurulduğunda Internet Explorer, SDM'ye erişmeye çalıştığınızda, HTML kaynak kodlarını gösterebilir. Bu problemi düzeltmek için Tools > Internet Options > Advanced seçin. Sonra, Security bölümünü kaydırın, Allow Active Content to Run in Files on My Computer'ı işaretleyin ve Apply butonuna tıklayın. Sonra SDM'yi tekrar açın.
7. SDM demo versiyonuna erişmek için masaüstünüzdeki Cisco SDM simgesine tıklayın. 127.0.0.1 loopback adresini girin ve SDM demo uygulamasını başlatmak için Launch butonuna tıklayın. Şayet isterseniz, HTTPS kullanmayı seçebilirsiniz. Bunu yaparsanız, ayrı pencerede açılan herhangi bir güvenli sertifika uyarı mesajını kabul edin.
8. SDM demosu size router'da debugging (hata ayıklama) çalıştığını ve performans düşüklüğüne sebep olabileceğinden kapatmanızı söyleyecektir. Fakat simülasyonda bunu açık bırakın, çünkü herhangi bir router performans düşmesi olmaz.
9. Tüm Tab'ları gözden geçirin: Configure Interfaces, Routing Protocols, Create a DHCP Pool vs. ve bulduğunuz herşeyi. SDM 'i öğrenmek için biraz zaman harcayın.
10. Demoda tüm özelliklerin desteklenmediğini bilin. Fakat hiç SDM olmamasında daha iyidir ve demo aslında oldukça iyi çalışmaktadır.
11. Sertifikalar kullanarak uygulama yapmak isterseniz, certificate authority (ca) ve router sertifikaları göstermek için ca.cer ve router.cer sertifikalarını kullanın. Bu sertifikaları, `SDM_demo_tool.zip` 'den alabilirsiniz.

# Gözden Geçirme Soruları

NOT *Aşağıdaki sorular, bu bölümün materyallerini anladığınızı test etmek için tasarlanmıştır. İlave sorulara nasıl sahip olacağınızla ilgili daha fazla bilgi için bu kitabın Giriş bölümüne bakın.*

1. `show running-config` yazın ve şu çıktıyı alın:

   ```
   i.      [output cut]
   ii.     Line console 0
   iii.    Exec-timeout 1 44
   iv.     Password 7O98COBQR
   v.      Login
   vi.     [output cut]
   ```

   Exec-timeout komutundaki iki sayı ne anlama gelmektedir?

   A. Şayet 44 saniye içinde hiçbir komut girilmezse, konsol bağlantısı kapanacaktır.

   B. 1 saat 44 dakika içinde hiçbir router aktivitesi algılanmazsa, konsol bağlantısı kapanacaktır.

   C. Şayet 1 dakika 44 saniye içinde hiçbir komut girilmezse, konsol bağlantısı kapanacaktır.

   D. Şayet bir Telnet bağlantısıyla router'a bağlandıysanız, 1 dakika 44 saniye içinde bir giriş tespit edilmelidir, yoksa bağlantı kapanacaktır.

2. Router'ınızda, bir LAN'da kullanılan broadcast adresini bulmanız gerekmektedir. Broadcast adresini bulmak için, user moddan router'a hangi komutu yazarsınız?

   A. `show running-config`

   B. `show startup-config`

   C. `show interfaces`

   D. `show protocols`

3. Router'ı yeniden başlatmayı ve mevcut running-config 'i mevcut startup-config 'in yerine yerleştirmek istiyorsunuz. Hangi komutu kullanırsınız?

   A. `replace run start`

   B. `copy run start`

   C. `copy start run`

   D. `reload`

4. Serial 0'a, bir DTE 'mi yoksa bir DCE'mi bağlı olduğunu hangi komut gösterir?

   A. `sh int s0`

   B. `sh int serial 0`

   C. `show controllers s 0`

   D. `show serial 0 controllers`

5. Hangi tuşa basmak setup modunu sonlandıracaktır?

   A. Ctrl+Z

   B. Ctrl+^

   C. Ctrl+C

   D. Ctrl+Shift+^

6. Konsol şifresi oluşturdunuz, fakat konfigürasyonu görüntülediğinizde, şifre görünmemektedir; görebildiğiniz sadece şudur:

```
[output cut]
Line console 0
Exec-timeout 1 44
Password 7O98COBQR
Login
[output cut]
```

   Şifrenin bu şekilde saklanmasına ne sebep olmuştur?

   A. `encrypt password`

   B. `service password-encryption`

   C. `service-password-encryption`

   D. `exec-timeout 1 44`

7. Aşağıdaki komutlardan hangisi, bir router'daki varsayılan VTY port'larının hepsini yapılandıracaktır?

   A. `Router#line vty 0 4`

   B. `Router(config)#line vty 0 4`

   C. `Router(config-if)#line console 0`

   D. `Router(config)#line vty all`

8. Aşağıdaki komutlardan hangisi, secret password'ü, Cisco olarak ayarlar?

   A. `enable secret password Cisco`

   B. `enable secret cisco`

   C. `enable secret Cisco`

   D. `enable password Cisco`

9. Router'a bağlandığında, yöneticilerin bir mesaj görmesini isterseniz, hangi komutu kullanırsınız?

   A. `message banner motd`

   B. `banner message motd`

   C. `banner motd`

   D. `message motd`

10. Bir Cisco router, varsayılan olarak eşzamanlı kaç Telnet oturumunu destekler?

    A. 1

    B. 2

    C. 3

    D. 4

    E. 5

    F. 6

11. RAM'de tutulan konfigürasyonu, NVRAM'e kaydetmek için hangi komutu kullanırsınız?

    A. `Router(config)#copy current to starting`

    B. `Router#copy starting to running`

    C. `Router(config)#copy running-config startup-config`

    D. `Router#copy run startup`

12. Router Corp'tan, SFRouter'a telnet yapmayı deniyorsunuz ve aşağıdaki mesajı alıyorsunuz:

```
Corp#telnet SFRouter
Trying SFRouter (10.0.0.1)…Open

Password required, but none set
[Connection to SFRouter closed by foreign host]
Corp#
```

    Aşağıdaki sıralamanın hangisi bu problemi doğru olarak çözecektir?

    A. `Corp(config)#line console 0`

    B. `SFRemote(config)#line console 0`

    C. `Corp(config)#line vty 0 4`

    D. `SFRemote(config)#line vty 0 4`

13. Hangi komut router'daki NVRAM 'in içeriğini silecektir?

    A. `delete NVRAM`

    B. `delete startup-config`

    C. `erase NVRAM`

    D. `erase start`

14. `show interface serial 0` yazdığınızda ve aşağıdaki mesajı alıyorsanız, interface ile ilgili sorun nedir?

    `Serial0 is administratively down, line protocol is down`

    A. keepalives süreleri farklıdır.

    B. Administrator, kapalı bir interface' e sahiptir.

    C. Administrator, interface'den ping atıyordur.

    D. Kablo bağlı değildir.

15. Aşağıdaki komutlardan hangisi, bir router'daki tüm interface ayarlanabilen parametreleri ve istatistikleri görüntüler?

    A. `show running-config`

    B. `show startup-config`

    C. `show interfaces`

    D. `show versions`

16. Şayet NVRAM'in içindekileri siler ve router'ı reboot ederseniz, hangi moda düşersiniz?

    A. Privileged mode

    B. Global mode

    C. Setup mode

    D. NVRAM loaded mode

17. Router'a aşağıdaki komutu yazınca, altındaki çıktıyı alıyorsunuz:

```
Router#show serial 0/0
           ^
% Invalid input detected at '^' marker.
```

Neden bu hata mesajı görüntülenir?

A. Privleged moda olmanız gerekir.

B. Serial ile 0/0 arasında boşluk olmaması gerekir.

C. Router, serial0/0 interface'ine sahip değildir.

D. Komutun parçası eksiktir.

18. `Router#sh ru` yazdınız ve bir % ambiguous command hatası alıyorsunuz. Bu mesajı neden alırsınız?

A. Komut, ilave seçenek ya da parametreler gerektiriyor.

B. ru ile başlayan birden fazla `show` komutu vardır.

C. ru ile başlayan bir `show` komutu yoktur.

D. Komut, yanlış router modundan çalıştırılmaktadır.

19. Aşağıdaki komutlardan hangileri, mevcut IP adreslemesi ve bir interface'deki katman 1 ile katman 2 durumunu görüntüler?

A. `show version`

B. `show protocols`

C. `show interfaces`

D. `show controllers`

E. `show ip interface`

F. `show running-config`

20. Şayet `show interface serial 1` yazınca, aşağıdaki mesajı alıyorsanız, problemin, OSI modelinin hangi katmanında olduğunu düşünürsünüz?

Serial1 is down, line protocol is down

A. Physical katman.

B. Data Link katmanı.

C. Network katmanı.

D. Hiçbirisi, bu bir router problemidir.

# Gözden Geçirme Sorularının Cevapları

1. C exec-timeout komutu, dakika ve saniye olarak ayarlanır.
2. C show ip protocols komutu, aslında size her interface'in broadcast adresini gösterir (çok kötü bir seçim olacaktır, bu sebeple muhtemel cevaplardan değildir). En iyi cevap, show interfaces komutudur. Bu, her interface'in IP adresini ve subnet maskını sağlayacaktır. Modül3 'te öğrendiğiniz engin subnet'leme bilgisiyle mask'ı tespit edebilirsiniz.
3. D C seçeneğini seçebilirsiniz (bu kötü bir cevap değildir). Konfigürasyonun yerine konmadığını hatırlayın, onun yerine, ilave edilir. Running-config'i, startup-config ile tamamen değiştirmek için, router'ı reload etmeniz gerekir.
4. C show controllers serial 0 komutu, interface'e, DTE ya da DCE kablosu bağlı olduğunu gösterecektir. Şayet bir DCE bağlantısıysa, clock rate komutu ile saat denetimi eklemeniz gerekmektedir.
5. C Setup moddan, Ctrl+C tuş bileşimine basarak istediğiniz zaman çıkabilirsiniz.
6. B Global configuration moddan, service password-encryption komutu, password'leri şifreleyecektir.
7. B Global configuration moddan, 5 varsayılan VTY hattını ayarlamak için line vty 0 4 komutunu kullanın.
8. C enable secret şifresi, büyük/küçük harf duyarlıdır. Bu nedenle ikinci şık yanlıştır. Enable secret şifresini oluşturmak için, global configuration moddan, enable secret password komutunu kullanın.
9. C Normalde kullanılan banner, message of the day'dir (MOTD) ve banner motd, global configuration modu banner motd komutu kullanarak oluşturulur.
10. E Enterprise edition bir IOS yoksa Cisco router'lar, varsayılan olarak beş eşzamanlı Telnet oturumuna sahiptir.
11. D Router tekrar başlatıldığında kullanılması için running-config'i NVRAM 'e kopyalamak için, copy running-config startup-config (kısaca copy run star) komutunu kullanın.
12. D Router'ınıza bir VTY'e (Telnet) izin vermek için VTY şifresi ayarlamalısınız. Yanlış router'da şifre ayarlandığında, C şıkkı yanlıştır. Cevaplarda, şifre oluşturmadan, login komutunu kullandığına dikkat edin. Cisco'nun, login komutundan önce şifre oluşturmanıza izin verebileceğini hatırlayın.
13. D erase startup-config komutu, NVRAM'in içindekileri siler ve router tekrar başlatılırsa sizi setup moda götürür.
14. B Şayet bir interface kapalıysa, show interface komutu, interface'i administratively olarak gösterir.(kablo bağlı olmaması ihtimali var, fakat bu mesaja bakarak bunu söyleyemezsiniz).
15. C show interfaces komutu ile ayarlanabilen parametreleri görebilir, router'daki interfaceler için istatistikler alabilir, interface'in kapalı olup olmadığını doğrulayabilir ve her interface'in IP adresini görebilirsiniz.
16. C Şayet, startup-config'i silip, router'ı reload ederseniz, router otomatik olarak setup moda girecektir. Ayrıca, privileged modda, istediğiniz zaman setup komutu yazarak setup moda geçebilirsiniz.
17. D User moddan interface istatistiklerini görebilirsiniz, fakat komut, show interface serial 0/0'dır.
18. B % ambiguous command hatasının anlamı, ru ile başlayan birden fazla muhtemel komut olmasıdır. Doğru komutu bulmak için soru işaretini (?) kullanın.

19. B, C, E show protocols, show interfaces ve show ip interface komutları, size, router'ınızın interfacelerinin IP adreslerini ve katman 1 ile katman2 durumunu gösterecektir.
20. A Şayet, hem seri interface hem de protokolü down olarak görürseniz, Fiziksel katman problemi vardır. Şayet, serial1 is up, line protocol is down görürseniz, uzak uçtan, (Data Link) keepalive'lar almıyorsunuzdur.

# Yazılı Lab 4 Cevapları

1. clock rate 64000
2. config t, line vty 0 4, no login
3. config t, int e0, no shut
4. erase startup-config
5. config t, line console 0, login, password todd
6. config t, enable secret cisco
7. show controllers int
8. show terminal
9. Router#reload
10. config t, hostname Chicago

# 5 Bir Cisco Ağ Topluluğunu Yönetmek

# 5 Bir Cisco Ağ Topluluğunu Yönetmek

- Bir Cisco Router'ın İç Bileşenleri
- Router Boot Sıralaması
- Configuration Register'ı Yönetmek
- Cisco IOS'u Yedeklemek ve Geri Yüklemek
- Cisco Konfigürasyonunu Yedeklemek ve Geri Yüklemek
- Cisco Discovery Protocol (CDP) Kullanmak
- Telnet Kullanmak
- Hostname'leri Çözümlemek
- Network Bağlanırlığını Kontrol Etmek ve Hata Tespiti Yapmak
- Özet
- Sınav Gereklilikleri
- Yazılı Lab 5
- Pratik Lab'lar
- Gözden Geçirme Soruları
- Gözden Geçirme Sorularının Cevapları
- Yazılı Lab 5 Cevapları

# Bir Cisco Ağ Topluluğunu Yönetmek

Bölüm 5'te bir ağ topluluğundaki Cisco router'ların nasıl yönetileceğini göstereceğim. Internetwork Operating System (IOS) ve bir Cisco cihazda farklı lokasyonlarda bulunan konfigürasyon dosyalarının nerede bulundukları ve nasıl çalıştıklarını anlamak gerçekten çok önemlidir.

Bir router'ın temel bileşenlerini, router boot sıralaması ve configuration register'ı öğreneceksiniz. (Şifre kurtarmak için configuration register'ın nasıl kullanıldığı da anlatılacaktır.) Bundan sonra, Cisco IOS File System (IFS) ve Cisco SDM kullanıldığında, bir TFTP host ile `copy` komutu kullanarak router'ların nasıl yönetileceğini öğreneceksiniz.

Bölümü (Cisco Discovery Protocol) CDP'yi inceleyerek bitireceğiz ve hostname'lerin çözümlenmesi ile bazı önemli Cisco IOS hata tespiti tekniklerini öğreneceksiniz.

*Bu bölümle ilgili son güncellemeler için www.lammle.com ve/veya www.sybex.com adreslerine bakınız.*

NOT

## Bir Cisco Router'ın İç Bileşenleri

Bir Cisco ağ topluluğunu yapılandırmak ve hata tespiti için, Cisco router'ın temel bileşenlerini bilmeniz ve her birinin işlevlerini anlamanız gerekir. Tablo 5.1, Cisco router'ın temel bileşenlerini açıklamaktadır.

**Tablo 5.1:** Cisco Router Bileşenleri

| Bileşen | Açıklama |
|---|---|
| Bootstrap | ROM'un mikro kodunda tutulur. Bootstrap, router'ın başlatılması esnasında, aktif duruma getirilmesi için kullanılmaktadır. Router'ı boot edecek ve IOS'u yükleyecektir. |
| POST (power-on self-test) | ROM'un mikro kodunda tutulur, router donanımının temel fonksiyonlarının kontrol edilmesi ve hangi interface'lerin olduğunun tespit edilmesi için kullanılmaktadır. |
| ROM monitor | ROM'un mikro kodunda tutulur. ROM monitor, üretim, test etme ve hata tespiti için kullanılmaktadır. |
| Mini-IOS | Cisco tarafından, RXBOOT veya bootloader olarak tanımlanır. Mini-IOS, bir interface'i aktifleştirmek ve flash belleğe bir Cisco IOS yüklemek için kullanılabilir. Mini-IOS ayrıca diğer bazı onarım operasyonları da çalıştırılabilir. |
| RAM (random access memory) | Paket arabellekleri, ARP önbelleği, routing tablosu ve router'ın çalışmasını sağlayan veri yapısını ve yazılımları tutmak için kullanılır. Running-config, RAM'de tutulur ve çoğu router, boot sırasında, IOS'u flash'tan RAM'e açar. |
| ROM (read-only memory) | Router'ı başlatmak ve devamlılığını sağlamak için kullanılır. Hem POST ve bootstrap programını hem de mini-IOS'u tutar. |
| Flash memory | Varsayılan olarak Cisco IOS'u saklar. Router reload edildiğinde, flash bellek silinmez. Intel tarafından oluşturulan bir EEPROM'dur (electronically erasable programmable read-only memory). |
| NVRAM (nonvolatile RAM) | Router ve switch konfigürasyonunu tutmak için kullanılır. Router veya switch reload edildiğinde, NVRAM silinmez. IOS'u tutmaz. Configuration register, NVRAM'de bulunur. |
| Configuration register | Router'ın nasıl boot edildiğini kontrol etmek için kullanılır. Bu değer, `show version` komutu çıktısının son satırında bulunabilir ve varsayılan olarak, router'a IOS'u flash bellekten, konfigürasyonu da NVRAM'den yükleyeceğini söyleyen, 0x2102 olarak ayarlanmıştır. |

# Router Boot Sıralaması

Bir router boot edildiğinde, donanımı test etmek ve gerekli yazılımı yüklemek için boot sıralaması denilen adımları çalıştırır. Boot sıralaması, aşağıdaki adımlardan oluşur:

1. Router bir POST çalıştırır. POST, cihazın tüm bileşenlerinin kullanıma hazır ve mevcut olduğunu doğrulamak için donanımı test eder. Örnek olarak, POST, router'daki farklı interface'leri kontrol eder. POST, ROM'da (read-only memory) saklanır ve ROM'dan çalışır.
2. Sonra bootstrap, Cisco IOS yazılımını arar ve onu yükler. Programları çalıştırmak için kullanılan bootstrap, ROM'da bulunan bir programdır. Bootstrap programı, her IOS programının nerede bulunduğunu bulmak ve sonra dosyayı yüklemekten sorumludur. Varsayılan olarak IOS yazılımı, tüm Cisco router'larda flash bellekten yüklenir.

NOT *Bir router'dan IOS yüklenmesinin varsayılan sırası, Flash, TFTP sunucusu ve ROM'dur.*

3. IOS yazılımı, NVRAM'de tutulan geçerli bir konfigürasyon dosyası arar. Bu dosya, startup-config olarak tanımlanır ve sadece bir yönetici, NVRAM'e kopyalarsa vardır. (Zaten bildiğiniz gibi yeni ISR router'lar, önceden yüklü ufak bir startup-config'e sahiptir.)
4. Şayet bir startup-config dosyası NVRAM'de mevcutsa, router bu dosyayı kopyalayacak, RAM'e yerleştirecek ve onu running-config olarak adlandıracaktır. Router bu dosyayı, router'ın çalışması için kullanacaktır. Router, şimdi çalışabilir durumdadır. Şayet NVRAM'de bir startup-config dosyası bulunmazsa router tüm interfacelerinden, üzerinde konfigürasyon dosyası bulunan bir TFTP host'u bulmak için broadcast gönderecektir. Şayet bu başarısız olursa (genelde başarısız olacaktır, birçok insanın bu prosesin işlediğinden haberi bile olmaz) router, setup mod konfigürasyon işlemi başlatacaktır.

# Configuration Register'ı Yönetmek

Tüm Cisco router'lar NVRAM'e yazılı, 16-bit bir yazılım register'ına sahiptir. Varsayılan olarak configuration register, Cisco IOS'un flash bellekten ve startup-config dosyasının NVRAM'den bulunup yüklenmesi için ayarlanmaktadır. Aşağıdaki bölümlerde, configuration register ayarlarını ve bu ayarların router'ınızda şifre kurtarmayı nasıl sağlayacağını göstereceğim.

## Configuration Register Bit'lerini Anlamak

16 bit (2byte) configuration register, soldan sağa doğru, 15'ten 0'a okunur. Cisco router'lardaki varsayılan configuration register ayarı, 0x2102'dir. Bunun anlamı, Tablo 5.2'de görüldüğü gibi 13, 8 ve 1 bit'lerinin aktif olmasıdır. Her 4 bit grubunun (nibble olarak belirtilir) 8, 4, 2, 1 değerleriyle binary olarak okunduğuna dikkat edin.

**Tablo 5.2:** Configuration Register Bit Numaraları

| **Configuration Register** | **2** | | | | | | | **1** | | | | **0** | | | **2** | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit numarası | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Binary | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |

Tablo 5.3, yazılımsal konfigürasyon bit'lerinin anlamlarını listelemektedir. 6. bit'in, NVRAM içeriklerini göz ardı etmek için kullanılabileceğine dikkat edin. Bu bit'ler, şifre kurtarmak için kullanılır (bu bölümde, "Şifre Kurtarmak" başlığı altında göreceğiz).

NOT *Configuration register adresine, 0x ekleyin. 0x'in anlamı, bunu izleyen sayıların hexadecimal olduğudur.*

NOT *Hexadecimal'd, tasarımın 0-9 ve A-F şeklinde olduğunu hatırlayın (A = 10, B = 11, C = 12, D = 13, E = 14 ve F = 15). Yani, configuration register'ı 210F olarak ayarlamak, gerçekte 210(15) veya binary olarak 1111'dir.*

**Tablo 5.3:** Yazılımsal Konfigürasyon Anlamları

| Bit | Hex | Açıklama |
|---|---|---|
| 0–3 | 0x0000–0x000F | Boot alanı (Tablo 5.4'e bakın). |
| 6 | 0x0040 | NVRAM içeriklerini yok sayın. |
| 7 | 0x0080 | OEM bit'i etkin. |
| 8 | 0x101 | Break devre dışı. |
| 10 | 0x0400 | Tamamı sıfırlarla, broadcast adresi. |
| 5, 11–12 | 0x0800–0x1000 | Konsol hat hızı. |
| 13 | 0x2000 | Network'ten boot gerçekleşmezse, varsayılan ROM yazılımından boot etme. |
| 14 | 0x4000 | Net numarası olmayan IP broadcast'i. |
| 15 | 0x8000 | Sistem tanı mesajlarını etkin kılma ve NVRAM içeriğini yok sayma. |

Configuration register'da 0-3 bit'lerden oluşan boot alanı, router boot sıralamasını kontrol eder. Tablo 5.4, boot alanı bit'lerini açıklamaktadır.

**Tablo 5.4:** Boot Alanı (Configuration Register Bit'leri 00-03)

| Boot alanı | Anlamı | Kullanımı |
|---|---|---|
| 00 | ROM monitor mode | ROM monitor moda boot etmek için configuration register'ı 2100'e ayarlayın. Router'ı, manuel olarak, boot etmelisiniz. Router `rommon>` istemcisini gösterecektir. |
| 01 | ROM'dan boot imajı | ROM'da tutulan bir IOS imajını boot etmek için configuration register değerini 2101 olarak ayarlayın. Router, `Router(boot)>` istemcisini gösterecektir. |
| 02–F | Varsayılan bir boot dosya adı belirtir. | 2102'den 210F'e kadar herhangi bir değer router'a NVRAM'de belirtilen boot komutlarını kullanmasını söyler. |

## Mevcut Configuration Register Değerini Kontrol Etmek

`show version` (kısaca, `sh version` veya `show ver`) komutunu kullanarak, mevcut configuration register değerini görebilirsiniz:

```
Router#sh version
Cisco IOS Software, 2800 Software (C2800NM-ADVSECURITYK9-M), Version
    12.4(12), RELEASE SOFTWARE (fc1)
[output cut]
Configuration register is 0x2102
```

Bu komut çıktısından alınan son satırdaki bilgi, configuration register değeridir. Bu örnekte, değer, varsayılan olan 0x2102'dir. 0x2102 configuration register değeri, router'a, boot sıralaması için NVRAM'e bakmasını söyler.

`show version` komutunun ayrıca IOS versiyonunu da verdiğine dikkat edin. Yukarıdaki örnekte IOS versiyonu, 12.4(12) olarak görünmektedir.

NOT *show version komutu, bir router'daki sistem donanım konfigürasyon bilgisini, yazılım versiyonunu ve boot imaj isimlerini görüntüleyecektir.*

## Configuration Register'ı Değiştirmek

Router'ın nasıl boot edeceğini ve çalışacağını belirlemek için configuration register değerini değiştirebilirsiniz. Configuration register değerini değiştirmek istemenin ana sebepleri şunlar olabilir:

- Sistemi ROM monitor moduna geçirmeye zorlamak için.
- Boot kaynağını ve varsayılan boot dosya adını seçmek için.
- Break fonksiyonunu etkin kılmak veya kapatmak için.
- Broadcast adreslerini kontrol etmek için.
- Konsol terminal aktarım hızını ayarlamak için.
- ROM'dan işletim sistemi yazılımı yüklemek için.
- Bir Trivial File Transfer Protocol (TFTP) sunucusundan boot etmeyi mümkün kılmak için.

NOT *Configuration register'ı değiştirmeden önce, kullanımdaki configuration register değerini bildiğinizden emin olun. Bu bilgiyi almak için show version komutunu kullanın.*

Configuration register'ı `config-register` komutunu kullanarak değiştirebilirsiniz. İşte bir örnek; aşağıdaki komutlar, router'a ROM'dan sınırlı IOS'u boot etmesini söyler ve sonra da, kullanılan configuration register değerini gösterir:

```
Router(config)#config-register 0x2101
Router(config)#^Z
Router#sh ver
[output cut]
Configuration register is 0x2102 (will be 0x2101 at next reload)
```

`show verion` komutunun, mevcut olan configuration değerini ve ayrıca router boot ettiğinde bu değerin ne olacağını gösterdiğine dikkat edin. Configuration register'da yapılan herhangi bir değişiklik, router, reload edilene kadar etkili olmayacaktır. 0x2101, router tekrar boot edildiği zaman, ROM'daki IOS'u yükleyecektir. Onun, 0x2101 olarak listelendiğini görebilirsiniz. Aslında, ikisi de aynı şeydir ve her iki şekilde de yazılabilmektedir.

Configuration register'ı 0x2101 olarak ayarlayıp reload edildiğinde, router çıktısı şöyledir:

```
Router(boot)#sh ver
Cisco IOS Software, 2800 Software (C2800NM-ADVSECURITYK9-M),
Version
    12.4(12), RELEASE SOFTWARE (fc1)
[output cut]

ROM: System Bootstrap, Version 12.4(13r)T, RELEASE SOFTWARE (fc1)

Router uptime is 3 minutes
System returned to ROM by power-on
System image file is "flash:c2800nm-advsecurityk9-mz.124-12.bin"
[output cut]

Configuration register is 0x2101
```

Bu noktada `show flash` yazarsanız, flash bellekteki çalışmaya hazır IOS'u görürsünüz. Fakat router'ımıza ROM'dan yüklemesini söyledik.

```
Router(boot)#sh flash
-#- –length– ––-date/time––– path
1    21710744 Jan 2 2007 22:41:14 +00:00 c2800nm-advsecurityk9-
mz.124-12.bin
2    1823 Dec 5 2006 14:46:26 +00:00 sdmconfig-2811.cfg
3    4734464 Dec 5 2006 14:47:12 +00:00 sdm.tar
4    833024 Dec 5 2006 14:47:38 +00:00 es.tar
5    1052160 Dec 5 2006 14:48:10 +00:00 common.tar
6    1038 Dec 5 2006 14:48:32 +00:00 home.shtml
7    102400 Dec 5 2006 14:48:54 +00:00 home.tar
8    491213 Dec 5 2006 14:49:22 +00:00 128MB.sdf
9    1684577 Dec 5 2006 14:50:04 +00:00 securedesktop-ios-
     3.1.1.27-k9.pkg
10   398305 Dec 5 2006 14:50:34 +00:00 sslclient-win-1.1.0.154.pkg

32989184 bytes available (31027200 bytes used)
```

Flash bellekte komple IOS'umuz olduğu halde, configuration register'ı değiştirerek, router yazılımının varsayılan yüklemesini değiştirdik. Şayet, configuration register'ı tekrar varsayılan değerine ayarlamak isterseniz, sadece şunu yazın:

```
Router(boot)#config t
Router(boot)(config)#config-register 0x2102
Router(boot)(config)#^Z
Router(boot)#reload
```

Şimdiki bölümde, şifre kurtarma işlemi yapabilmeniz için router'ın, ROM monitor moda nasıl düşürüldüğünü göstereceğim.

## Şifre Kurtarmak

Şayet router'ınız, şifreyi unuttuğunuz için kitlendiyse, tekrar çalışır duruma gelmesine yardımcı olması için configuration register değerini değiştirebilirsiniz. Daha önce söylediğim gibi, configuration register'daki 6.bit, router'a, router konfigürasyonunu yüklemek için NVRAM'in içindekileri kullanıp kullanmayacağını söyler.

Varsayılan configuration register değeri, 0x2102'dir (6.bit'in kullanılmadığı anlamına gelir). Varsayılan ayar ile router, NVRAM'de (startup-config) tutulan bir router konfigürasyonunu bulur ve yükler. Bir şifreyi kurtarmak için, 6.bit kullanmanız gerekir. Bunu yapmak router'a NVRAM'in içindekileri göz ardı etmesini söyler. 6.bit'i kullandığınızda configuration register değeri, 0x2142'dir.

Şifre kurtarmanın başlıca adımları şunlardır:

1. Router'ı boot edin ve break tuşuna basarak boot sıralamasını atlayın. Bu, router'ı ROM monitor moduna düşürecektir.
2. Configuration register değerini, 6. bit'i kullanacak şekilde, 0x2142 olarak değiştiririn.
3. Router'ı reload edin.
4. Privileged moda girin.
5. Startup-config'i, running-config'e kopyalayın.
6. Şifreyi değiştirin.

7. Configuration register'ı, varsayılan değere getirin.
8. Router konfigürasyonunu kaydedin.
9. Router'ı reload edin (opsiyonel).

Bu adımları sonraki bölümlerde detaylı bir şekilde işleyeceğim. Aynı zamanda ISR, 2600 ve hatta 2500 serisi router'lara erişimi tekrar sağlamak için kullanılan komutları da göstereceğim. (Hala 2500 serisi router'ları kullanıyor olabilirsiniz ve ihtiyacınız olduğunda bu bilgileri bilmiyor olabilirsiniz!)

Söylediğim gibi, router boot edilirken Ctrl+Break tuş kombinasyonuna basarak ROM monitor moda girebilirsiniz. Fakat IOS, bozuk ya da kayıpsa bir TFTP host'una erişmek için network bağlantısı yoksa veya ROM'daki mini-IOS yüklenmezse (router için varsayılan son umudun yok olduğu anlamına gelir), router varsayılan olarak ROM monitor moduna düşecektir.

### Router Boot Sıralamasını Atlamak

İlk adımınız, router'ı boot etmek ve break işlemidir. Bu genelde, router ilk tekrar boot edilirken, HyperTerminal (kişisel olarak ben SecureCRT'yi tercih ederim) kullanıldığında, Ctrl+Break tuş kombinasyonuna basarak yapılmaktadır.

Ctrl+Break yaptıktan sonra, bir 2600 serisi router için şunları görmelisiniz (ISR serisi router'larla nerdeyse aynı çıktıdır):

```
System Bootstrap, Version 11.3(2)XA4, RELEASE SOFTWARE (fc1)
Copyright (c) 1999 by cisco Systems, Inc.
TAC:Home:SW:IOS:Specials for info
PC = 0xfff0a530, Vector = 0x500, SP = 0x680127b0
C2600 platform with 32768 Kbytes of main memory
PC = 0xfff0a530, Vector = 0x500, SP = 0x80004374
monitor: command "boot" aborted due to user interrupt
rommon 1>
```

Monitor satırına dikkat edin: `command "boot" aborted due to user interrupt`. Bu noktada, ROM monitor modu olarak belirtilen, `rommon 1>` istemcisinde olacaksınız.

## Configuration Register'ı Değiştirmek

NOT *Configuration register değerini 0x2142 olarak değiştirirseniz, startup-config'in devre dışı bırakılacağını ve router'ın startup moduna geçeceğini hatırlayın.*

Daha önce açıkladığım gibi, `config-register` komutunu kullanarak, configuration register'ı değiştirebilirsiniz. 6.bit'i kullanmak için configuration register değerini, 0x2142 olarak kullanın.

### Cisco ISR/2600 Serisi Komutları

Bir Cisco ISR/2600 serisi router'daki bit değerini değiştirmek için `rommon 1>` deki komutu girin:

```
rommon 1 >confreg 0x2142
You must reset or power cycle for new config to take effect
rommon 2 >reset
```

### Cisco 2500 Serisi Komutlar

Bir 2500 serisi router'daki configuration register'ı değiştirmek için, router'da "break sıralaması" oluşturduktan sonra o yazın. Bu, configuration register seçenek ayarlarının olduğu bir menü getirir. Configuration register'ı değiştirmek için, yeni register değeri ile devam eden `o/r` komutunu girin. Aşağıda, bir 2501 router'da 6.bit'in kullanılmasıyla ilgili bir örnek bulabilirsiniz:

```
System Bootstrap, Version 11.0(10c), SOFTWARE
Copyright (c) 1986-1996 by cisco Systems
2500 processor with 14336 Kbytes of main memory
Abort at 0x1098FEC (PC)
>o
Configuration register = 0x2102 at last boot
Bit#    Configuration register option settings:
15      Diagnostic mode disabled
14      IP broadcasts do not have network numbers
13      Boot default ROM software if network boot fails
12-11   Console speed is 9600 baud
10      IP broadcasts with ones
08      Break disabled
07      OEM disabled
06      Ignore configuration disabled
03-00   Boot file is cisco2-2500 (or'boot system' command)
>o/r 0x2142
```

Router çıktısındaki son kayıtın 03-00 olduğuna dikkat edin. Varsayılan olarak, router, flash bellekte bulunan ilk dosyayı kullanacaktır. Bu nedenle, eğer farklı dosya adından boot etmek isterseniz, configuration register'ı değiştirebilir veya `boot system flash: ios_name` komutunu kullanabilirsiniz (`boot system` komutuna birazdan değineceğim).

## Router'ı Reload Etmek ve Privileged Moda Girmek

Bu noktada router'ı şu şekilde sıfırlamalısınız:

- ISR/2600 serisi router'dan, `I` (initialize (başlatmak) için) veya `reset` yazın.
- 2500 serisi router'dan `I` yazın.

Router reload olacaktır ve setup modu kullanmak isteyip istemediğiniz sorulacaktır (çünkü kullanılacak startup-config yoktur). Setup-moda girmemek için no deyin ve user moda gitmek için Enter tuşuna basın. Sonrada, privileged moda girmek için `enable` yazın.

## Konfigürasyona Bakmak ve Değiştirmek

Şimdi, bir router'da user mod ve privileged mod şifreleri girmeniz gereken noktayı geçtiniz. Startup-config'i running-config'e kopyalayın:

```
copy startup-config running-config
```

Veya kısa yol kullanın

```
copy start run
```

Konfigürasyon şimdi, random access memory'de (RAM) çalışıyor ve siz privileged moddasınız (yani, konfigürasyona göz atıp, onu değiştirebilirsiniz). Fakat şifrelendiğinden, password için **enable secret** ayarını göremezsiniz. Şifreyi değiştirmek için şunu yapın:

```
config t
enable secret todd
```

### Configuration Register'ı Sıfırlamak ve Router'ı Yeniden Başlatmak

Şifre değişimini tamamladıktan sonra, configuration register'ı, `config-register` komutu ile tekrar varsayılan değerine ayarlayın:

```
config t
config-register 0x2102
```

NOT *Şayet konfigürasyonunuzu kaydedip router'ı yeniden başlattığınızda setup moduna düşerseniz, configuration register ayarı muhtemelen yanlıştır.*

Son olarak, `copy running-config startup config` ile yeni konfigürasyonu kaydedin ve router'ı yeniden başlatın.

## Boot Sistem Komutları

Flash bozuksa, router'ınızı başka IOS'tan boot etmek için yapılandırabileceğinizi biliyor musunuz? Evet, bunu yapabilirsiniz. Aslında router'ınızı her seferinde bir TFTP host'undan boot etmeyi isteyebilirsiniz. Çünkü bu yöntemle her router'ı teker teker güncellemek zorunda kalmazsınız. Bu, TFTP host'undaki bir dosyada değişiklik yaparak güncelleme sağladığından, kullanmak için en akıllıca yöntemdir

Kullanmayı düşünebileceğiniz bazı boot komutları vardır. Bunlar router'ınızı Cisco IOS'la boot etmenizi yönetmek için yardımcı olur. Bu arada, router'ın konfigürasyonundan değil de, router'ın IOS'undan bahsettiğimizi unutmayın!

```
Router>en
Router#config t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#boot ?
  bootstrap  Bootstrap image file
  config     Configuration file
  host       Router-specific config file
  network    Network-wide config file
  system     System image file
```

`boot` komutu, tamamıyla bir seçenek zenginliği sunar, fakat ilk olarak size Cisco'nun önerdiği standart ayarları göstereceğim. Gelin başlayalım; `boot system` komutu router'a, flash bellekten boot etmesi için hangi dosyayı kullanacağını söylemenizi sağlar. Varsayılan olarak, router'ın ilk başta flash bellekte bulunan dosyayı boot ettiğini hatırlayın. Bunu şu şekilde değiştirebilirsiniz:

```
Router(config)#boot system ?
  WORD   TFTP filename or URL
  flash  Boot from flash memory
  ftp    Boot from a server via ftp
  mop    Boot from a Decnet MOP server
  rcp    Boot from a server via rcp
  rom    Boot from rom
  tftp   Boot from a tftp server
Router(config)#boot system flash c2800nm-advsecurityk9-mz.124-12.bin
```

Yukarıdaki komut, router'ı kendinde kayıtlı IOS'tan boot etmesi için yapılandırır. Bu, flash'a yeni bir IOS yükleyip, onu test etmek istediğinizde ve hatta varsayılan olarak yüklenen IOS'u tamamıyla değiştirmek istediğinizde faydalı bir komuttur.

Şimdiki komut son çare olarak düşünülmektedir, fakat söylediğim gibi, router'larınızı bir TFTP host'undan boot etmek için onu kalıcı yöntem olarak kullanabilirsiniz. Kişisel olarak, bunu (single point of failure) yapmanızı kesinlikle tavsiye etmem, ben sadece size olabilirliğini gösteriyorum:

```
Router(config)#boot system tftp ?
  WORD  System image filename
Router(config)#boot system tftp c2800nm-advsecurityk9-mz.124-
12.bin ?
  Hostname or A.B.C.D  Address from which to download the file
  <cr>
Router(config)#boot system tftp c2800nm-advsecurityk9-mz.124-
12.bin 1.1.1.2
Router(config)#
```

Şayet flashtaki IOS yüklenmez ve TFTP host'u IOS sağlamazsa, önerilen son çare olarak ROM'dan mini-IOS şu şekilde yüklenir:

```
Router(config)#boot system rom
Router(config)#do show run | include boot system
boot system flash c2800nm-advsecurityk9-mz.124-12.bin
boot system tftp c2800nm-advsecurityk9-mz.124-12.bin 1.1.1.2
boot system rom
Router(config)#
```

Özet olarak şimdi router'ımızda yapılandırılmış, Cisco'nun önerdiği IOS yedekleme yöntemlerine sahibiz: Flash, TFTP host, ROM.

# Cisco IOS'u Yedeklemek ve Geri Yüklemek

Bir Cisco IOS'u güncellemeden veya geri yüklemeden önce, yeni imajın bozulması durumunda mevcut dosyayı yedek olarak bir TFTP host'una kopyalayın.

Bunun için herhangi bir TFTP host'unu kullanabilirsiniz. Varsayılan olarak, bir router'daki flash bellek, Cisco IOS'u saklamak için kullanılır. Aşağıdaki bölümlerde, flash belleğin boyutunun nasıl kontrol edileceğini, Cisco IOS'un flash bellekten bir TFTP host'una ve IOS'un bir TFTP host'undan, flash belleğe nasıl kopyalanacağını açıklayacağım.

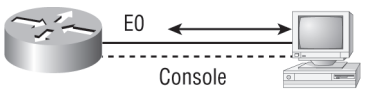
*İlk olarak, bir TFTP host'u ile onların nasıl yönetileceğini öğrendikten sonra, IOS dosyalarınızı yönetmek için Cisco IFS ve SDM'i nasıl kullanacağınızı göreceksiniz.* **NOT**

Bir IOS imajını, iç ağınızdaki bir network sunucusuna yedeklemeden önce şu üç şeyi yapmalısınız:

- Network sunucusuna erişebildiğinizden emin olun.
- Network sunucusunun, kod imajı için yeterli alana sahip olduğundan emin olun.
- Dosya isimlendirmesini ve yol gereksinimlerini kontrol edin.

Şayet, Şekil 5.1'de gösterildiği gibi, router'ın Ethernet interface'ine direkt bağlı laptop veya workstation'ın Ethernet portuna sahipseniz, router'a veya router'dan imaj kopyalamaya girişmeden önce şunları doğrulamalısınız:

- TFTP sunucu yazılımı, yönetici (admin) makinesinde çalışıyor olmalıdır.
- Router ve workstation arasındaki Ethernet bağlantısı, çapraz kablo ile yapılmalıdır.
- Workstation, router'ın interface'i ile aynı subnette olmalıdır.
- Şayet router flash'ından kopyalıyorsanız, `copy flash tftp` komutunun, workstation'ın IP adresini desteklemesi gerekir.
- Şayet flash'a kopyalıyorsanız, kopyalanacak dosyayı kaydetmek için flash bellekte yeterli alanın olup olmadığını kontrol etmelisiniz.




***Şekil 5.1:*** *IOS'u bir workstation'dan bir router'a kopyalamak.*

## Flash Belleği Kontrol Etmek

Router'ınızdaki Cisco IOS'u yeni bir IOS dosyası ile upgrade etmeden önce, yeni imajı saklamak için flash belleğinizde yeterli alan olup olmadığını kontrol etmek iyi bir fikirdir. Flash belleğin boyutunu ve flash bellekte tutulan dosya veya dosyaları, `show flash` (kısaca `sh flash`) komutunu kullanarak, kontrol edersiniz:

```
Router#sh flash
-#- –length– ––-date/time––– path
1     21710744 Jan 2 2007 22:41:14 +00:00 c2800nm-advsecurityk9-
      mz.124-12.bin
[output cut]
32989184 bytes available (31027200 bytes used)
```

NOT *show flash komutu, hem kullanılan IOS imajının kapladığı alanın boyutunu gösterir hem de size, mevcut ve yeni imajları saklamak için uygun alan olup olmadığını söyler. Eski ve yüklemeyi istediğiniz yeni imaj için yeterli yeriniz olup olmadığını bilmelisiniz. Eğer yer yoksa eski imaj silinecektir!*

Yukarıdaki ISR router, 64MB RAM'e sahiptir ve yaklaşık olarak bunun yarısı kullanılmaktadır. Flash boyutunu, ISR router'larda `show version` komutu kullanarak, kontrol etmek gerçekten kolaydır:

```
Router#show version
[output cut]
Cisco 2811 (revision 49.46) with 249856K/12288K bytes of memory.
Processor board ID FTX1049A1AB
2 FastEthernet interfaces
4 Serial(sync/async) interfaces
1 Virtual Private Network (VPN) Module
DRAM configuration is 64 bits wide with parity enabled.
239K bytes of non-volatile configuration memory.
62720K bytes of ATA CompactFlash (Read/Write)
```

Son satırda, flash bellek boyutunu görebilirsiniz. Ortalama olarak, 64MB flash boyutuna sahibiz.

Bu örnekteki dosya adının `c2800nm-advsecurityk9-mz.124-12.bin` olduğuna dikkat edin. `show flash` ve `show version` komutlarının çıktıları arasındaki başlıca farklılık, `show flash` komutunun flash'taki tüm dosyaları görüntülemesi ve `show version` komutunun, router'ın router'ı çalıştırmak için kullandığı dosyanın gerçek ismini göstermesidir.

## Cisco IOS'u Yedeklemek

Cisco IOS'u bir TFTP sunucusuna yedeklemek için `copy flash tftp` komutunu kullanırsınız. Bu, sadece kaynak dosya adı ve TFTP sunucusunun IP adresini gerektiren, anlaşması kolay bir komuttur.

Bu yedekleme işleminde başarının anahtarı, TFTP'ye iyi, sağlam bir bağlantınızın olduğundan emin olmanızdır. Router konsol istemcisinden TFTP cihazını pingleyerek bunu şu şekilde test edin:

```
Router#ping 1.1.1.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 1.1.1.2, timeout
  is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max
  = 4/4/8 ms
```

*Packet Internet Groper (Ping) aracı, network bağlanırlığını kontrol etmek için kullanılır ve ben onu bu bölümde çeşitli örneklerde kullanıyorum. "Network Bağlanırlığını Kontrol Etmek ve Hata Tespiti" başlıklı ilerleyen bölümde bunun hakkında detaylı olarak bahsedeceğim.*

NOT

IP'nin çalıştığından emin olmak için TFTP sunucusunu ping'ledikten sonra, aşağıda görüldüğü gibi, IOS'u TFTP sunucusuna kopyalamak için `copy flash tftp` komutunu kullanabilirsiniz:

```
Router#copy flash tftp
Source filename []?c2800nm-advsecurityk9-mz.124-12.bin
Address or name of remote host []?1.1.1.2
Destination filename [c2800nm-advsecurityk9-mz.124-
12.bin]?[enter]
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!
21710744 bytes copied in 60.724 secs (357532 bytes/sec)
Router#
```

IOS dosya adını, ya `show flash` ya da `show version` komutundan kopyalayın ve sonra onu kaynak dosya adı istenen yere yapıştırın.

Yukarıdaki örnekte flash belleğin içindekiler TFTP sunucusuna başarıyla kopyalanmıştır. Uzak host'un adresi, TFTP host'unun IP adresidir ve kaynak dosya adı, flash bellekteki dosyadır.

*copy flash tftp komutu, herhangi bir dosyanın lokasyonunu istemeyecek ve bu dosyayı nereye yerleştireceğinizi sormayacaktır. Yani, TFTP sunucusu belirli ve geçerli bir dizine sahip olmalıdır, aksi halde çalışmayacaktır.*

NOT

## Cisco Router IOS'unu Tekrar Yüklemek ya da Yükseltmek

IOS'u upgrade etmek istediğinizde veya bozulan orijinal bir dosyanın yerine koymak için Cisco IOS'u flash belleğe geri yüklemeniz gerekirse, ne yaparsınız? Dosyayı, `copy tftp flash` komutunu kullanarak, bir TFTP sunucusundan flash belleğe indirebilirsiniz. Bu komut için TFTP host'unun IP adresi ve indirmek istediğiniz dosyanın adı gerekir.

Fakat başlamadan önce, flash belleğe yerleştirmeyi istediğiniz dosyanın host'unuzdaki geçerli TFTP dizininde olduğundan emin olun. Komutu çalıştırdığınızda TFTP dosyanın nerede olduğunu sormayacaktır, bu nedenle kullanmak istediğiniz dosya TFTP host'unun geçerli dizininde değilse, bu çalışmayacaktır.

```
Router#copy tftp flash
Address or name of remote host []?1.1.1.2
Source filename []?c2800nm-advsecurityk9-mz.124-12.bin
Destination filename [c2800nm-advsecurityk9-mz.124-12.bin]?[enter]
%Warning:There is a file already existing with this name
Do you want to over write? [confirm][enter]
Accessing tftp://1.1.1.2/c2800nm-advsecurityk9-mz.124-12.bin...
Loading c2800nm-advsecurityk9-mz.124-12.bin from 1.1.1.2 (via
   FastEthernet0/0): !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
[OK - 21710744 bytes]

21710744 bytes copied in 82.880 secs (261954 bytes/sec)
Router#
```

Yukarıdaki örnekte aynı dosyayı flash belleğe kopyaladım, bu nedenle, bana onu üstüne yazıp yazmak istemediğimi sordu. Flash bellekteki dosyalarla çalıştığımızı unutmayın. Dosyamı üzerine yazarak bozsaydım, router'ımı yeniden boot edene kadar bilemeyecektim. Bu komuta dikkat edin! Şayet dosya bozulursa, ROM monitor moddan bir IOS yeniden yükleme işlemi gerçekleştirmeniz gerekecektir.

NOT *Bir Cisco router, flash bellekte çalışan bir router sistem imajı için TFTP sunucu host'u olabilir. Global konfigürasyon komutu tftp-server: ios_filename'dir.*

Şayet yeni bir dosya yüklüyorsanız ve yeni ile mevcut kopyaların her ikisini de saklamak için flash bellekte yeterli alanınız yoksa router, yeni dosyayı flash belleğe yazmadan önce, flash belleğin içeriğini silip silmeyeceğinizi soracaktır.

## Cisco IOS File System (IFS) Kullanmak

Cisco, Cisco IFS olarak belirtilen bir Windows DOS komut istemcisindeymişsiniz gibi dosya ve dizinlerle çalışmanızı sağlayan bir dosya sistemi oluşturdu. Kullandığınız komutlar şunlardır; `dir`, `copy`, `more`, `delete`, `erase` veya `format`, `cd` ile `pwd`, `mkdir` ve `rmdir`.

IFS ile çalışmak size tüm dosyalara göz atmak ve onları sınıflamak kabiliyeti sağlar (bunlar uzak sunucularda olsa dahi). Uzak sunucularınızdan birindeki imajın, onu kopyalamadan önce geçerli olup olmadığını anlamayı kesinlikle istersiniz, değil mi? Aynı zamanda onun boyutunun ne kadar olduğunu da bilmeniz gerekir (burada boyut önemlidir). Ayrıca, uzak sunucunun konfigürasyonuna bakmak ve router'ınızdaki dosyayı yüklemeden önce, onun iyi durumda olduğundan emin olmak gerçekten iyi bir fikirdir.

IFS'in dosya sistemi kullanıcı ara yüzünü evrensel yapması iyi bir özelliktir (platform spesifik değildir). Tüm router'larınızdaki tüm komutlarınız için aynı sözdizimini kullanabilirsiniz (platform önemli değildir).

İnanılmayacak kadar iyi mi görünüyor? Evet, öyle sayılır. Çünkü hiçbir dosya sisteminde ve platformda, tüm komutlar için destek bulunmadığını anlayacaksınız. Fakat farklı dosya sistemleri çalıştırdıkları uygulamalarda farklılık gösterdiğinden, bu çok büyük sorun değildir. Belirli dosya sistemleri ile ilgili olmayan komutlar, çok desteklenmeyenlerdir. Herhangi bir dosya sistemi veya platformun, yönetmeniz gereken tüm komutları tamamıyla desteklediğinden emin olun.

Diğer bir güzel IFS özelliği, komutların çoğu için tüm bu zorunlu komut istemcilerini azaltmasıdır. Şayet bir komut satırına girmek isterseniz tüm yapmanız gereken, komut satırına düzgün şekilde tüm gerekli bilgiyi yazmaktır. Artık istemcilerin etrafında zıplamak yok! Bu nedenle, eğer bir dosyayı FTP sunucusuna kopyalamak isterseniz, tüm yapmanız gereken ilk olarak istenen kaynak

dosyasının router'ınızda nerede olduğunu göstermek, hedef dosyanın FTP sunucusunda tam yerini belirlemek, bu sunucuya bağlanmak istediğinizde kullanacağınız kullanıcı adı ve şifresini tanımlamak ve bunların hepsini tek bir satıra düzgün bir şekilde yazmaktır. Ve değişikliklere direnenleriniz, hala gereken tüm bilgiler için router istemcisine sahipsiniz ve daha önce yaptığınızdan daha hassas, minimize edilmiş komutlar girmenin tadını çıkarabilirsiniz.

Fakat tüm buna rağmen, router'ınız hala sizden bilgi isteyebilir. (komut satırınızda her şeyi doğru yapsanız dahi). Mesele, `file prompt` komutuna nasıl sahip olduğumuz ve hangi komutu kullanmaya çalıştığınızdır. Fakat kaygılanmayın, şayet bu olursa, varsayılan değer, komuttaki doğru yere girilecektir ve tüm yapmanız gereken, doğru değerleri kontrol etmek için Enter'a basmaktır.

IFS ayrıca, çeşitli dizinler aramaya ve istediğiniz bir dizindeki dosyaların envanterini çıkarmanıza izin verir. Artı, flash bellekte ve bir kartta altdizinler oluşturabilirsiniz, fakat bunu daha güncel platformlardan birinde çalışıyorsanız yapabilirsiniz.

Yeni dosya sistemi arayüzü, bir dosyanın olduğu yeri tespit etmek için URL'ler kullanır. Web'te kesin yerlerini belirlemek gibi URL'ler şimdi, dosyanın Cisco router'ınızda, hatta uzak bir dosya sunucusunda nerede olduğunu gösterir. Dosya veya dizinin nerede olduğunu tespit etmek için sadece komutlarınıza URL'leri doğru yazın. Bir dosyayı bir yerden başka bir yere kopyalamak gerçekten kolaydır. Sadece, `copy` ***`source-url destination-url`*** komutunu girersiniz. IFS URL'leri, yine de kullandıklarınızdan biraz farklıdır ve dosyanın kesin olarak nerede olduğuna bağlı olarak değişen bir formatı vardır.

Biz Cisco IFS komutlarını, daha önce IOS bölümünde `copy` komutunu kullandığımızla hemen hemen aynı yöntemle kullanacağız:

- IOS'u yedeklemek için.
- IOS'u upgrade etmek için.
- Text dosyalarına bakmak için.

Bütün bu yazılanlarla, gelin, IOS'u yönetmek için uygun, genel IFS komutlarına bir göz atalım. En kısa zamanda konfigürasyon dosyalarından bahsedeceğim, fakat şimdi, yeni Cisco IOS'larını yönetmek için kullanılan temel komutlarla başlayacağım.

**`dir:`** Windows'la aynı olan bu komut, bir dizindeki dosyalara göz atmanıza izin verir. `dir` yazıp Enter'a basın, varsayılan olarak `flash:/` dizininin içindekilerin çıktısını alırsınız.

**`copy:`** Bu oldukça popüler bir komuttur. Sıkça, bir IOS'u upgrade etmek, yeniden yüklemek veya yedeklemek için kullanılır. Fakat söylediğim gibi, onu kullandığınız zaman ne kopyaladığınız, onun nereden geldiği ve nereye yerleşeceği gibi detaylara odaklanmanız gerçekten önemlidir.

**`more:`** Unix ile aynı olan bu komut, size bir text dosyası verecektir ve ona bir kartta bakmanıza izin verecektir. Siz onu, konfigürasyon dosyanızı veya yedek konfigürasyon dosyanızı kontrol etmek için kullanabilirsiniz. Gerçek konfigürasyona geçtiğimizde onu daha detaylı işleyeceğiz.

**`show file:`** Bu komut, belirli bir dosya veya dosya sisteminde kolaylık sağlar, insanlar onu çok kullanmadığından, pek bilinen bir komut değildir.

**`delete:`** Tahmin edebileceğiniz gibi, bir şeyleri siler, Fakat bazı router türlerinde, tam düşündüğünüz gibi değildir. Dosyayı sakladığı halde, kullandığı alan her zaman boş değildir. Alanı gerçekten geri almak için `squeeze` komutunu da kullanmak zorundasınız.

**`erase/format:`** Bunu dikkatli kullanın. Dosya kopyalarken, dosya sistemini silmek isteyip istemediğinizi sorduğunda "hayır" deyin. Kullandığınız bellek türü, flash sürücüsünü reddedip edemeyeceğinizi belirler.

**`cd/pwd:`** Unix ve Dos ile aynıdır. `cd`, dizin değiştirmek için kullandığınız bir komuttur. `pwd`'yi, çalışılan dizini göstermesi için kullanın.

**mkdir/rmdir:** Bu komutu, belirli router ve switch'lerde dizinleri oluşturmak ve silmek için kullanın (`mkdir` komutunu dizin oluşturmak, `rmdir` komutunu silmek için) `cd` ve `pwd` komutlarını, bu dizinleri değiştirmek için kullanın.

## Bir IOS'u Upgrade Etmek İçin Cisco IFS Kullanmak

Gelin, bu Cisco IFS komutlarının bazılarına benim ISR router'ımda (1841 serisi), R1 hostname'i ile bir bakalım.

Varsayılan dizinimizi doğrulamak için `pwd` komutu ile başlayalım ve sonra varsayılan dizinin (`flash:/`) içindekileri kontrol etmek için `dir` komutunu kullanalım:

```
R1#pwd
flash:
R1#dir
Directory of flash:/
  1  -rw-    13937472  Dec 20 2006 19:58:18 +00:00  c1841-ipbase-
     mz.124-1c.bin
  2  -rw-        1821  Dec 20 2006 20:11:24 +00:00  sdmconfig-
     18xx.cfg
  3  -rw-     4734464  Dec 20 2006 20:12:00 +00:00  sdm.tar
  4  -rw-      833024  Dec 20 2006 20:12:24 +00:00  es.tar
  5  -rw-     1052160  Dec 20 2006 20:12:50 +00:00  common.tar
  6  -rw-        1038  Dec 20 2006 20:13:10 +00:00  home.shtml
  7  -rw-      102400  Dec 20 2006 20:13:30 +00:00  home.tar
  8  -rw-      491213  Dec 20 2006 20:13:56 +00:00  128MB.sdf
  9  -rw-     1684577  Dec 20 2006 20:14:34 +00:00  securedesktop-
 ios-3.1.1.27-k9.pkg
 10  -rw-      398305  Dec 20 2006 20:15:04 +00:00  sslclient-win-
 1.1.0.154.pkg

32071680 bytes total (8818688 bytes free)
```

Burada, (`c1841-ipbase-mz.124-1c.bin`) temel IOS'a sahip olduğumuzu görebiliriz. Görüldüğü üzere 1841 router'ımızı upgrade etmeye ihtiyacımız var. Şimdi, Cisco'nun IOS tipine, nasıl dosya adı verdiğini öğrenmek üzeresiniz. İlk olarak, flash'taki dosya boyutunu `show file` (`show flash` komutu da çalışacaktır) komutunu kullanarak kontrol edelim:

```
R1#show file info flash:c1841-ipbase-mz.124-1c.bin
flash:c1841-ipbase-mz.124-1c.bin:
  type is image (elf) []
  file size is 13937472 bytes, run size is 14103140 bytes
  Runnable image, entry point 0x8000F000, run from ram
```

Bu boyutu ile 21MB'tan daha fazla olan yeni IOS dosyamızı (`c1841-advipservicesk9-mz.`) eklemeden önce, mevcut IOS'un silinmesi gerekir. `delete` komutunu kullanacağız, fakat hatırlayın, biz flash bellekteki herhangi bir dosya üzerinde oynayabiliriz, tekrar boot edene kadar ciddi bir sorun olmayacaktır. Şayet hata yaptıysak, bu olur. Daha önce bu noktaya işaret ettiğim gibi, burada kesinlikle dikkatli olmamız gerekir!

```
R1#delete flash:c1841-ipbase-mz.124-1c.bin
Delete filename [c1841-ipbase-mz.124-1c.bin]?[enter]
Delete flash:c1841-ipbase-mz.124-1c.bin? [confirm][enter]
R1#sh flash
-#- –length– ––-date/time––– path
1         1821 Dec 20 2006 20:11:24 +00:00 sdmconfig-18xx.cfg
2      4734464 Dec 20 2006 20:12:00 +00:00 sdm.tar
3       833024 Dec 20 2006 20:12:24 +00:00 es.tar
4      1052160 Dec 20 2006 20:12:50 +00:00 common.tar
5         1038 Dec 20 2006 20:13:10 +00:00 home.shtml
6       102400 Dec 20 2006 20:13:30 +00:00 home.tar
7       491213 Dec 20 2006 20:13:56 +00:00 128MB.sdf
8      1684577 Dec 20 2006 20:14:34 +00:00 securedesktop-ios-
       3.1.1.27-k9.pkg
9       398305 Dec 20 2006 20:15:04 +00:00 sslclient-win-
       1.1.0.154.pkg
22757376 bytes available (9314304 bytes used)
R1#sh file info flash:c1841-ipbase-mz.124-1c.bin
%Error opening flash:c1841-ipbase-mz.124-1c.bin (File not found)
R1#
```

Yukarıdaki komutlarla mevcut dosyayı sildim ve sonra `show flash` ve `show file` komutlarını kullanarak silme işleminin doğruluğunu kontrol ettim. Şimdi, `copy` komutu ile yeni bir dosya ekleyelim, fakat bunun daha önce gösterdiğim ilk yöntemden daha güvenli olmamasından dolayı, dikkatli olmalıyım:

```
R1#copy tftp://1.1.1.2//c1841-advipservicesk9-mz.124-12.bin/
flash:/
    c1841-advipservicesk9-mz.124-12.bin
Source filename [/c1841-advipservicesk9-mz.124-12.bin/]?[enter]
Destination filename [c1841-advipservicesk9-mz.124-12.bin]?[enter]
Loading /c1841-advipservicesk9-mz.124-12.bin/ from 1.1.1.2 (via
    FastEthernet0/0): !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
[output cut]
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
[OK - 22103052 bytes]
22103052 bytes copied in 72.008 secs (306953 bytes/sec)
R1#sh flash
-#- –length– ––-date/time––– path
1         1821 Dec 20 2006 20:11:24 +00:00 sdmconfig-18xx.cfg
2      4734464 Dec 20 2006 20:12:00 +00:00 sdm.tar
3       833024 Dec 20 2006 20:12:24 +00:00 es.tar
4      1052160 Dec 20 2006 20:12:50 +00:00 common.tar
5         1038 Dec 20 2006 20:13:10 +00:00 home.shtml
6       102400 Dec 20 2006 20:13:30 +00:00 home.tar
7       491213 Dec 20 2006 20:13:56 +00:00 128MB.sdf
```

```
8      1684577 Dec 20 2006 20:14:34 +00:00 securedesktop-ios-
       3.1.1.27-k9.pkg
9       398305 Dec 20 2006 20:15:04 +00:00 sslclient-win-
       1.1.0.154.pkg
10    22103052 Mar 10 2007 19:40:50 +00:00 c1841-advipservicesk9-
       mz.124-12.bin
651264 bytes available (31420416 bytes used)
R1#
```

Dosya bilgisini, `show file` komutu ile kontrol edebiliriz:

```
R1#sh file information flash:c1841-advipservicesk9-mz.124-12.bin
flash:c1841-advipservicesk9-mz.124-12.bin:
  type is image (elf) []
  file size is 22103052 bytes, run size is 22268736 bytes
  Runnable image, entry point 0x8000F000, run from ram
```

IOS'un, router boot edildiğinde, RAM'e açıldığını hatırlayın. Bu nedenle, yeni IOS, siz router'ı yeniden başlatana kadar çalışmayacaktır. Şimdi gelin Cisco SDM'in router'ın IOS'unu nasıl upgrade ettiğine bir bakalım.

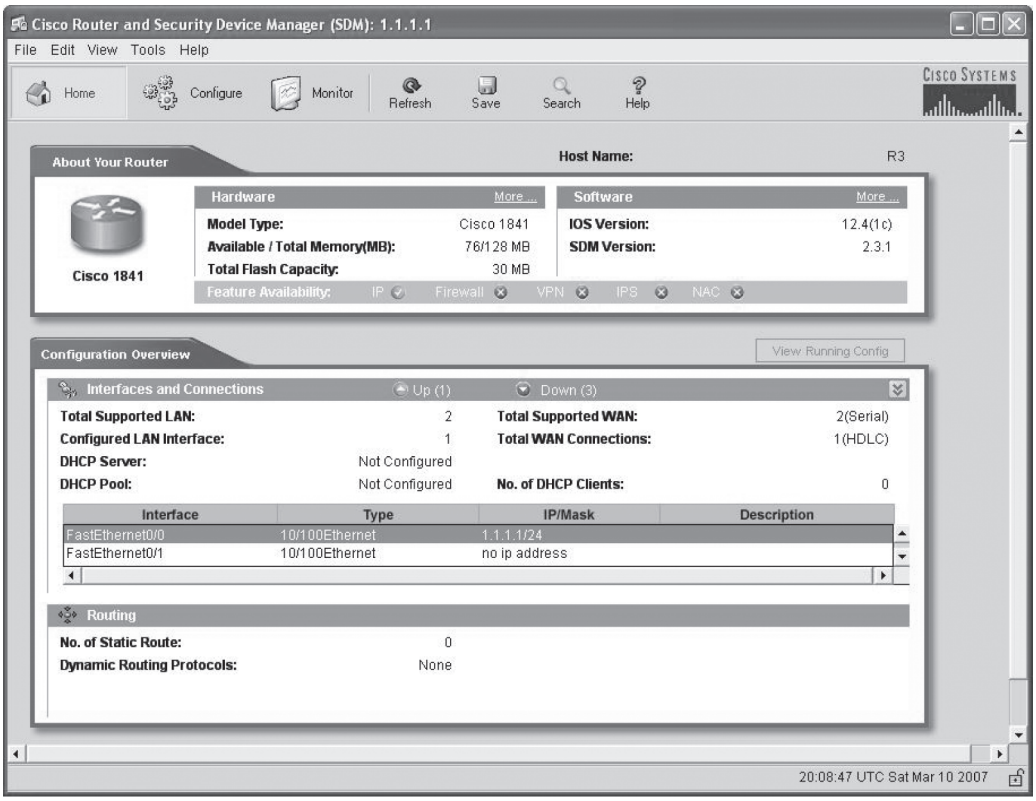
NOT *Ben gerçekten, bir router üzerinde bu komutları, iyi olduğunuzu hissettiğinizde kullanmanızı tavsiye ederim. Çünkü söylediğim gibi ilk kullandığınızda kesinlikle sorun çıkaracaktır.*

## Flash Belleği Yönetmek İçin SDM Kullanmak

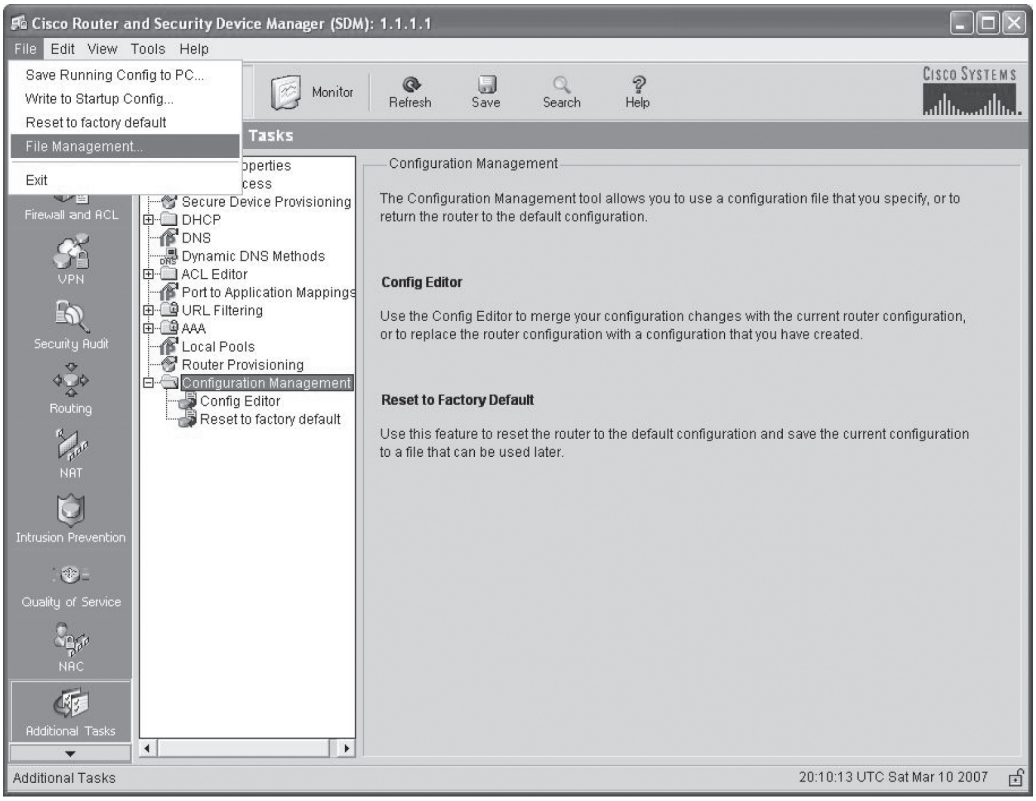
Bu bölümü "Router'ınızdaki IOS'u SDM kullanarak, Upgrade etmek/Yeniden Yüklemek/Yedeklemek" olarak tanımlayacaktım, fakat SDM, flash bellekteki (hem de NVRAM deki) tüm dosyaların yönetimini sağlar, fakat IOS'un yönetimini sağlamaz. SDM, flash bellek yönetimi için kolay bir yöntem olabilir fakat buna değmez. Onu bu yöntemle yapmayı güvenli bulmazsınız. Onun yerine, dosyalarınızı yönetmek için bir yöntemdir. Gelin kontrol edelim.

NOT *Bu bölümde sık sık daha güvenli yöntemlerden bahsettim. Açıkçası, flash bellekte çalıştığımda yeteri kadar dikkatli olmayarak çok ciddi problemlere sebep oldum. Flash bellekle oyalanırken, yeteri kadar dikkatli olmanızı söyleyemem!*

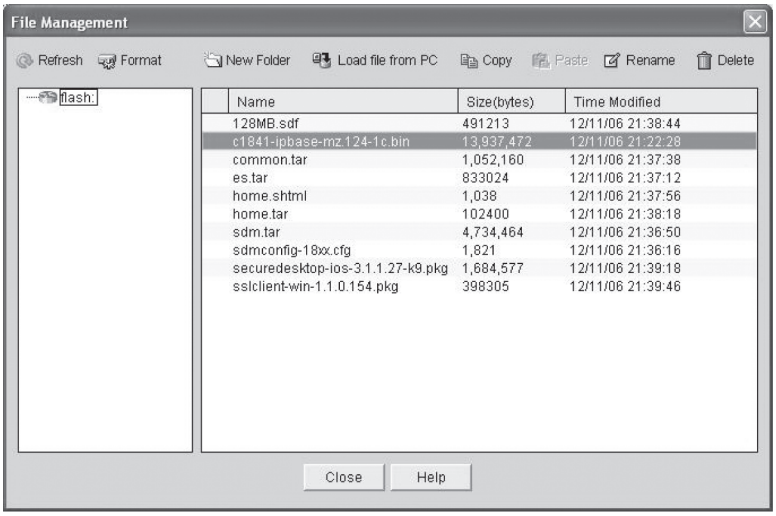
Diğer 1841 router'ımı (R3 ismini verdim) bağlayacağım ve SDM kullanarak IOS'u upgrade edeceğim. Gelin bağlayalım ve flash'ta ne olduğunu görelim. İlk ekrana baktığımızda, IP'nin tek kullanılabilir özellik olduğunu, Firewall, VPN, IPS ve NAC'ın olmadığını görebiliriz. Gelin bunu düzeltelim!



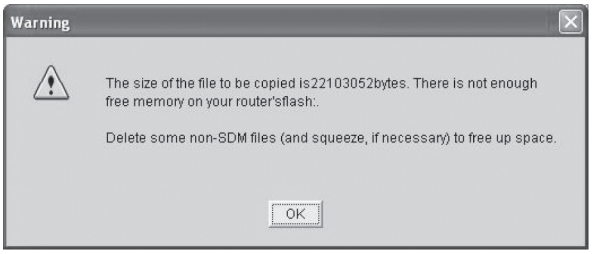

İkinci ekran, flash için File Management'ın nasıl açıldığını gösterir. Choose File – File Management



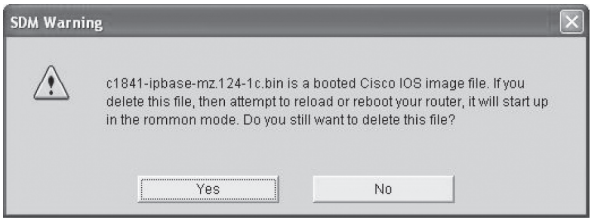
Bu noktada, ekran, flash'taki tüm dosyaları gösterir ve biz, "IP tabanlı" IOS'umuz olduğunu görebiliriz.



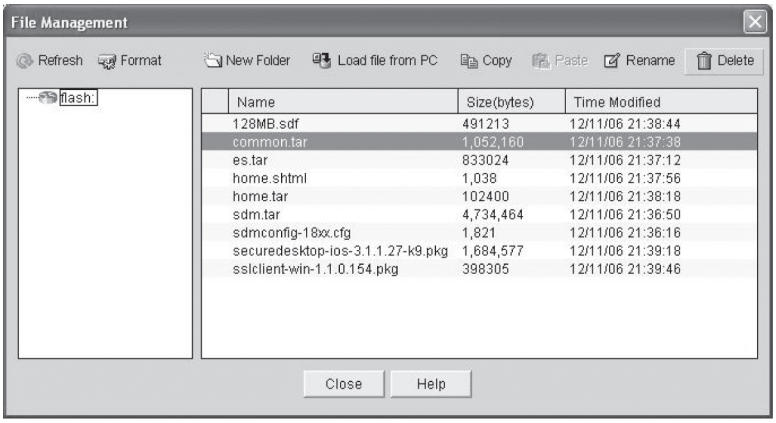

Yeni dosyayı eklemek için ekranın üzerindeki Load file from PC butonuna tıklayın. Yeni IOS'u yüklemeye çalıştığımda aşağıdaki mesajı aldım.



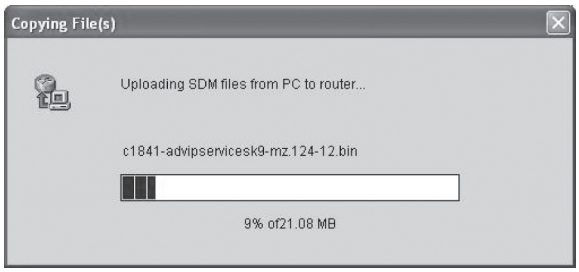

OK butonuna tıkladım, mevcut dosyayı silmeye çalıştığımda şu mesajı aldım.



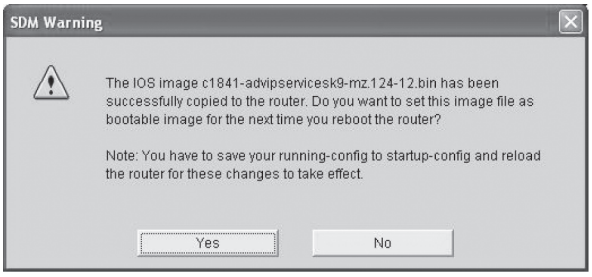

Yes'i seçtim ve sonra dosyanın silindiğini doğrulamak için File Management ekranına baktım.



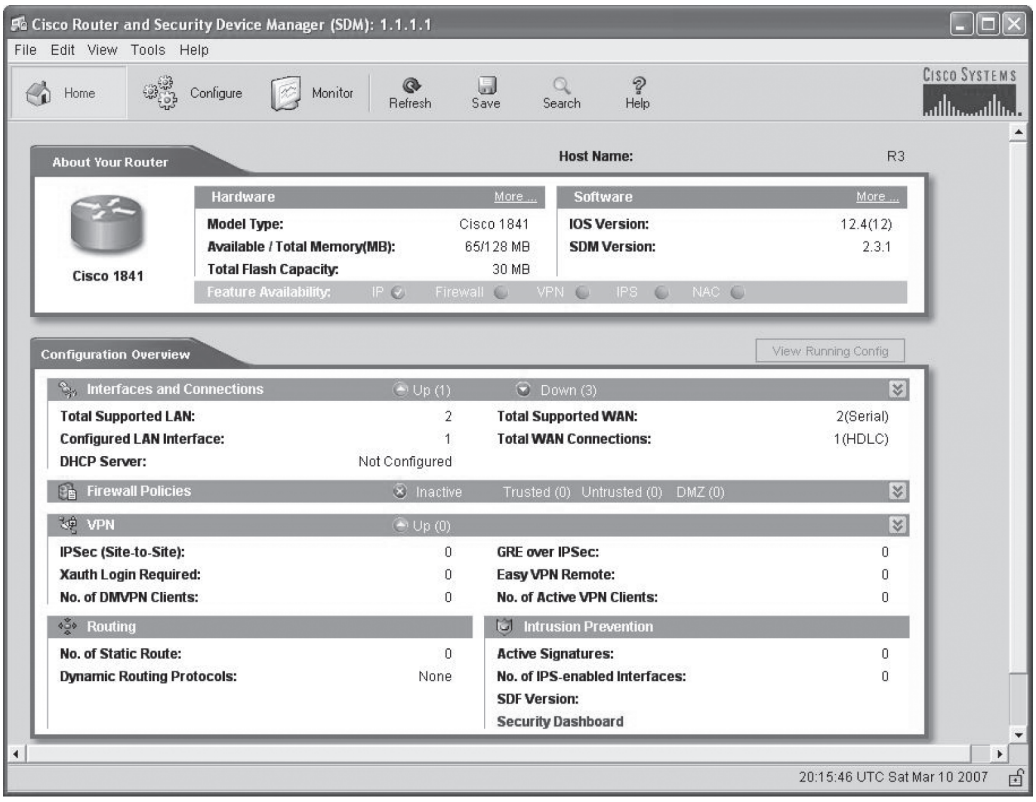
Sonra tekrar Load File from PC'yi seçtim ve dosya, flash belleğe gönderilmeye başladı.




Sonunda, tamamlandı!




Tekrar boot edildikten sonra, bu yeni IOS ile IP, Firewall, VPN, IPS ve NAC'ın kullanılır olduğunu görebiliriz!




NOT

*ISR router'ların mükemmel özelliklerinden birisi, router'ın ön veya arka yüzünden erişilebilen fiziksel flash kartlar kullanmasıdır. Bu kartları çıkarabilir ve PC'nizdeki uygun bir slot'a takabilirsiniz. Kart, bir drive olarak görünecektir. Bundan sonra dosyaları ekleyebilir, değiştirebilir ve silebilirsiniz. Flash kartınızı geri router'ınıza takın ve router'ı çalıştırın. İşte size anında upgrade! Güzel değil mi?*

# Cisco Konfigürasyonunu Yedeklemek ve Geri Yüklemek

Router konfigürasyonunda yaptığınız herhangi bir değişiklik, running-config dosyasında tutulacaktır. Şayet, running-config'de bir değişiklik yaptıktan sonra `copy run start` komutunu girmezseniz, router tekrar boot edilir veya elektrik giderse, bu değişiklik gidecektir. Bu yüzden siz muhtemelen router veya switch'inizin tamamıyla gitmesi durumunda, konfigürasyon bilginizin başka bir yedeğini almak istersiniz. Cihazınız iyi durumda olsa dahi, kaynak olarak kullanmak ve dokümantasyon nedeniyle bir yedeğe sahip olmak iyidir.

Şimdiki bölümde, bir router'ın konfigürasyonunun TFTP sunucusuna nasıl kopyalandığını ve bunun nasıl geri yüklendiğini açıklayacağım.

## Cisco Router Konfigürasyonunun Yedeğinin Alınması

Router konfigürasyonunu, bir router'dan TFTP sunucuya kopyalamak için `copy running-config tftp` veya `copy startup-config tftp` komutların kullanabilirsiniz. Komutlardan biri halen DRAM'de çalışan router konfigürasyonunu, diğeri de NVRAM'de saklanan konfigürasyonu yedekler.

### Çalışan Konfigürasyonu Kontrol Etmek

DRAM'deki konfigürasyonu kontrol etmek için `show running-config` (kısaca `sh run`) komutunu kullanın:

```
Router#show running-config
Building configuration...

Current configuration : 776 bytes
!
version 12.4
```

Çalışan konfigürasyon bilgisi, router'da 12.4 IOS versiyonunun çalıştığını gösterir.

### Saklanan Konfigürasyonunun Doğrulanması

Sonra, NVRAM'de tutulan konfigürasyonu kontrol etmelisiniz. Bunu görmek için `show startup-config` (kısaca `sh start`) komutunu kullanın:

```
Router#show startup-config
Using 776 out of 245752 bytes
!
version 12.4
```

İkinci satır, yedek konfigürasyonunuzun ne kadar yer tuttuğunu gösterir. Burada, NVRAM'in 239KB olduğunu görebiliriz (ISR router kullandığınızda `show version` komutu ile belleğe bakmak oldukça kolaydır) ve onun sadece 776 byte'ı kullanılmaktadır.

Şayet dosyaların aynı olduğundan ve running-config dosyasının kullanmak istediğiniz dosya olduğundan emin değilseniz, `copy running-config startup-config` komutunu kullanın. Bu, her iki dosyanın tamamen aynı olduğundan emin olmanıza yardımcı olacaktır. Bundan şimdiki bölümde bahsedeceğim.

### Mevcut Konfigürasyonu, NVRAM'e Kopyalamak

Running-config'i NVRAM'e yedek olarak kopyalayarak, aşağıdaki çıktıda görüldüğü gibi, running-config'inizin, router tekrar başlatıldığında daima tekrar yükleneceğinden emin olursunuz. Yeni IOS 12.0 versiyonunda, kullanmak istediğiniz dosya adı istenecektir:

```
Router#copy running-config startup-config
Destination filename [startup-config]?[enter]
Building configuration...
[OK]
Router#
```

Filename istemcisinin görünmesinin sebebi, `copy` komutunu kullandığınızda kullanabileceğiniz çok sayıda seçeneğin olmasıdır:

```
Router#copy running-config ?
  archive:        Copy to archive: file system
  flash:          Copy to flash: file system
  ftp:            Copy to ftp: file system
  http:           Copy to http: file system
  https:          Copy to https: file system
  ips-sdf         Update (merge with) IPS signature configuration
  null:           Copy to null: file system
  nvram:          Copy to nvram: file system
  rcp:            Copy to rcp: file system
  running-config  Update (merge with) current system configuration
  scp:            Copy to scp: file system
  startup-config  Copy to startup configuration
  syslog:         Copy to syslog: file system
  system:         Copy to system: file system
  tftp:           Copy to tftp: file system
  xmodem:         Copy to xmodem: file system
  ymodem:         Copy to ymodem: file system
```

`copy` komutunu tekrar, şimdiki bölümde kullanacağız.

### Konfigürasyonu, Bir TFTP Sunucusuna Kopyalamak

Dosyayı NVRAM'e kopyalayınca, `copy running-config tftp` (kısaca `copy run tftp`) komutunu kullanarak, TFTP sunucusunda ikinci bir yedekleme yapabilirsiniz:

```
Router#copy running-config tftp
Address or name of remote host []?1.1.1.2
Destination filename [router-confg]?todd-confg
!!
776 bytes copied in 0.800 secs (970 bytes/sec)
Router#
```

Yukarıdaki örnekte, router'a bir hostname vermediğim için `todd-confg` adını verdim. Şayet ayarlanmış bir hostname'iniz varsa komut, dosya ismi olarak, otomatikman, hostname artı `-confg` uzantısını kullanacaktır.

## Cisco Router Konfigürasyonunun Geri Yüklenmesi

Şayet router'ın running-config dosyasını değiştirdiyseniz ve konfigürasyonu startup-config dosyasındaki versiyona geri yüklemek istiyorsanız, bunu yapmanın en kolay yolu `copy startup-`

`config running-config` (kısaca `copy run star`) komutunu kullanmaktır. Aynı zamanda, bir konfigürasyonu tekrar yüklemek için eski `config mem` komutunu kullanabilirsiniz. Tabi ki bu sadece, herhangi bir değişiklik yapmadan önce NVRAM'e running-config'i kopyaladığınızda çalışacaktır!

Şayet, ikinci bir yedekleme olarak router konfigürasyonunu TFTP sunucusuna kopyaladıysanız, aşağıda görüldüğü gibi, konfigürasyonu `copy tftp running-config` (kısaca `copy tftp run`) komutunu veya `copy tftp startup-config` (kısaca `copy tftp star`) komutunu kullanarak geri yükleyebilirsiniz (bu fonksiyonu sağlayan eski komut `config net`'dir):

```
Router#copy tftp running-config
Address or name of remote host []?1.1.1.2
Source filename []?todd-confg
Destination filename[running-config]?[enter]
Accessing tftp://1.1.1.2/todd-confg...
Loading todd-confg from 1.1.1.2 (via FastEthernet0/0): !
[OK - 776 bytes]
776 bytes copied in 9.212 secs (84 bytes/sec)
Router#
*Mar  7 17:53:34.071: %SYS-5-CONFIG_I: Configured from
    tftp://1.1.1.2/todd-confg by console
Router#
```

Konfigürasyon dosyası bir ASCII text dosyasıdır, yani bir TFTP sunucusunda saklanan konfigürasyonu tekrar router'a kopyalamadan önce dosyada bir text editor ile değişiklik yapabilirsiniz. Son olarak, komutun `tftp://1.1.1.2/todd-config` şeklinde bir URL'e döndüğüne dikkat edin. Bu, Cisco IOS File System'dir (IFS) ve bunu birazdan, konfigürasyonumuzu yedekleyip tekrar yüklerken kullanacağız.

**NOT** *Bir konfigürasyonu, TFTP sunucusundan, router'ın RAM'ine kopyaladığınızda veya birleştirdiğinizde, interface'ler varsayılan olarak kapalıdır ve manuel olarak her interface'i no shutdown komutu ile etkinleştirebilirsiniz.*

## Konfigürasyonu Silmek

Bir Cisco router'daki startup-config'i silmek için `erase startup-config` komutunu kullanın:

```
Router#erase startup-config
Erasing the nvram filesystem will remove all configuration files!
    Continue? [confirm][enter]
[OK]
Erase of nvram: complete
*Mar  7 17:56:20.407: %SYS-7-NV_BLOCK_INIT: Initialized the
geometry of nvram
Router#reload
System configuration has been modified. Save? [yes/no]:n
Proceed with reload? [confirm][enter]
 *Mar  7 17:56:31.059: %SYS-5-RELOAD: Reload requested by console.
    Reload Reason: Reload Command.
```

Bu komut, router'daki NVRAM'in içindekileri siler. Şayet privileged modda, reload yazar ve değişiklikleri kaydetmesini kabul etmezseniz, router yeniden başlayacak ve setup moda geçecektir.

## Router'ınızın Konfigürasyonunu Yönetmek İçin Cisco IOS File System (Cisco IFS) Kullanmak

Eski, orijinal `copy` komutunu kullanmak hala faydalıdır ve ben tavsiye ederim. Yine de Cisco IFS konusunu bilmeniz gerekir. Yapacağımız ilk şey, NVRAM ve RAM'in içindekileri görmek için `show file` komutunu kullanmaktır:

```
R3#show file information nvram:startup-config
nvram:startup-config:
  type is config
R3#cd nvram:
R3#pwd
nvram:/
R3#dir
Directory of nvram:/

  190  -rw-         830            <no date>  startup-config
  191  ——           5              <no date>  private-config
  192  -rw-         830            <no date>  underlying-config
    1  -rw-           0            <no date>  ifIndex-table
196600 bytes total (194689 bytes free)
```

Aslında, NVRAM'in içindekileri gösterecek başka komut yoktur. Bununla birlikte, onları görmenin nasıl faydalı olacağından emin değilim. Gelin, RAM'in içindekilere bir bakalım:

```
R3#cd system:
R3#pwd
system:/
R3#dir ?
  /all             List all files
  /recursive       List files recursively
  all-filesystems  List files on all filesystems
  archive:         Directory or file name
  cns:             Directory or file name
  flash:           Directory or file name
  null:            Directory or file name
  nvram:           Directory or file name
  system:          Directory or file name
  xmodem:          Directory or file name
  ymodem:          Directory or file name
  <cr>
R3#dir
Directory of system:/

    3  dr-x           0             <no date>  lib
   33  dr-x           0             <no date>  memory
    1  -rw-         750             <no date>  running-config
    2  dr-x           0                   <no date>  vfiles
```

Çok heyecanlı sayılmaz. Cisco IFS ile bir dosyayı, TFTP host'undan RAM'e kopyalamak için `copy` komutunu kullanalım. İlk olarak, yıllar boyunca kullandığımız ve aynı çıktıyı almayı başardığımız eski komut olan `config net`'i kullanalım:

```
R3#config net
Host or network configuration file [host]?[enter]
This command has been replaced by the command:
        'copy <url> system:/running-config'
Address or name of remote host [255.255.255.255]?
```

Komut bize, komutun yeni URL komutu ile değiştirildiğini söylese de, eski komut hala çalışacaktır. Gelin onu Cisco IFS ile deneyelim:

```
R3#copy tftp://1.1.1.2/todd-confg system://running-config
Destination filename [running-config]?[enter]
Accessing tftp://1.1.1.2/todd-confg...Loading todd-confg from
1.1.1.2
    (via FastEthernet0/0): !
[OK - 776 bytes]
[OK]
776 bytes copied in 13.816 secs (56 bytes/sec)
R3#
*Mar 10 22:12:59.819: %SYS-5-CONFIG_I:
Configured from tftp://1.1.1.2/todd-confg by console
```

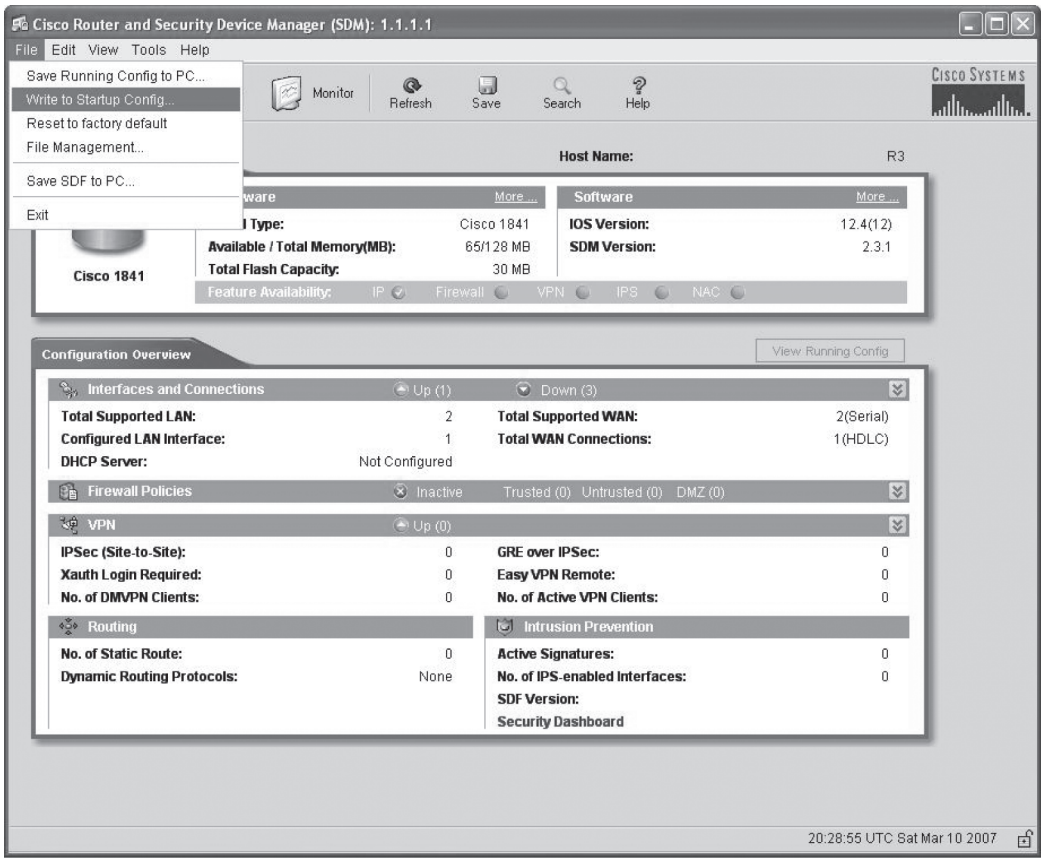
Bunun, `copy tftp run` komutundan daha kolay olduğunu söyleyebileceğimizi tahmin ediyorum. Cisco böyle düşünüyor, ben kimim ki bunu tartışayım? Gelin, router'ımıza HTTP veya HTTPS üzerinden bağlanarak ve konfigürasyon dosyalarımızı yönetmek için SDM kullanarak, bunu daha kolay yapıp yapamayacağımızı görelim.

## SDM Kullanarak, Router'ın Konfigürasyonunu Yedeklemek/Geri Yüklemek ve Düzenlemek

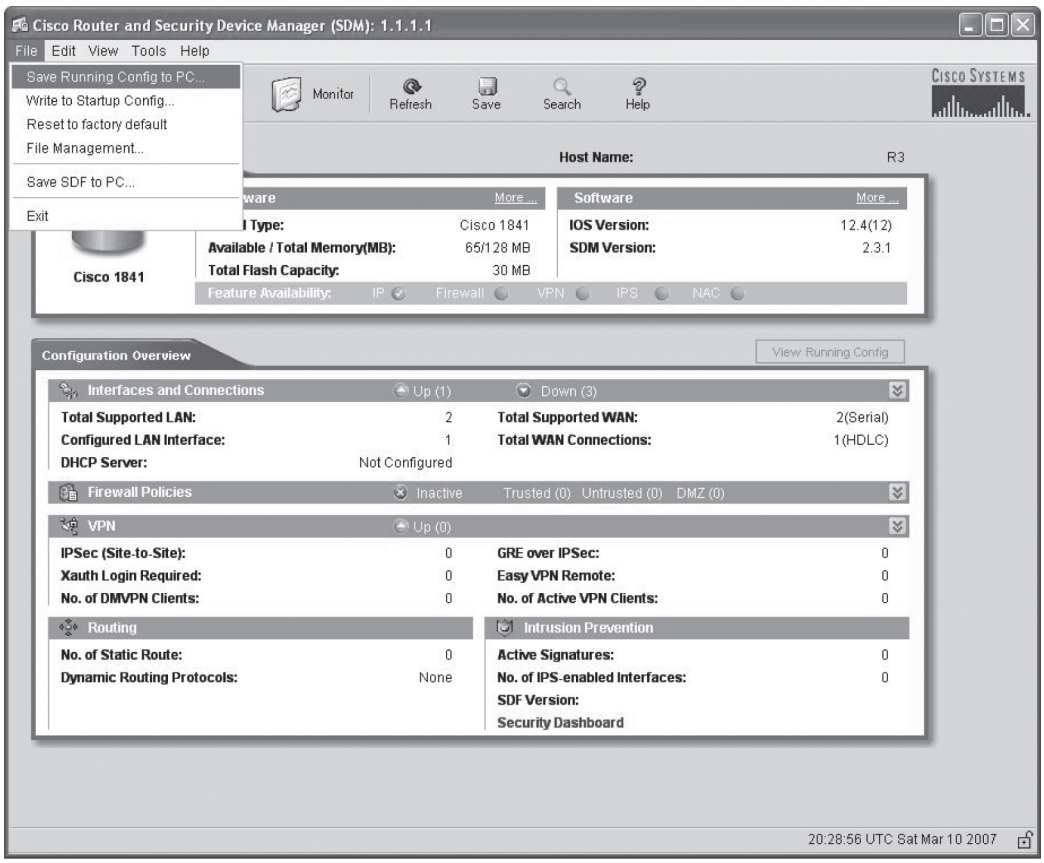
Dürüst olmak gerekirse, aslında SDM'in bir router'da konfigürasyonları nasıl ele aldığıyla ilgili olarak özel bir şey yoktur. Şayet bir router'a telnet yaptıysanız, `show run` çalıştırın ve çıktısını PC'nizdeki bir text dosyasına kopyalayın, SDM ve onun yönetim araçlarının ne yapabileceğini test ettiniz. Fakat bu hala, Cisco IFS ile yapmaktan daha az karışık bir yöntemdir.

Niçin? SDM kullanmayı, bu modülde daha önce gördüğümüz `copy` komutundan daha kolay yapan, TFTP host'una ihtiyaç olmamasıdır. SDM kullanarak, bir router'a http veya https yapabilir ve bir TFTP host'u yapılandırmak zorunda kalmak yerine, tüm dosyaları PC'nizde lokalde tutabilirsiniz. Bölüm 14'te açıkladığım gibi; SDM, güvenlik, IPS, QoS ve NAT gibi gelişmiş konfigürasyonlar için kullanılan en iyi yöntemdir. Şimdiye kadar fark ettiğiniz gibi, ben command-line interface (CLI) kullanan biriyim. Nasıl söylenir, eski alışkanlıklar zor değişir!

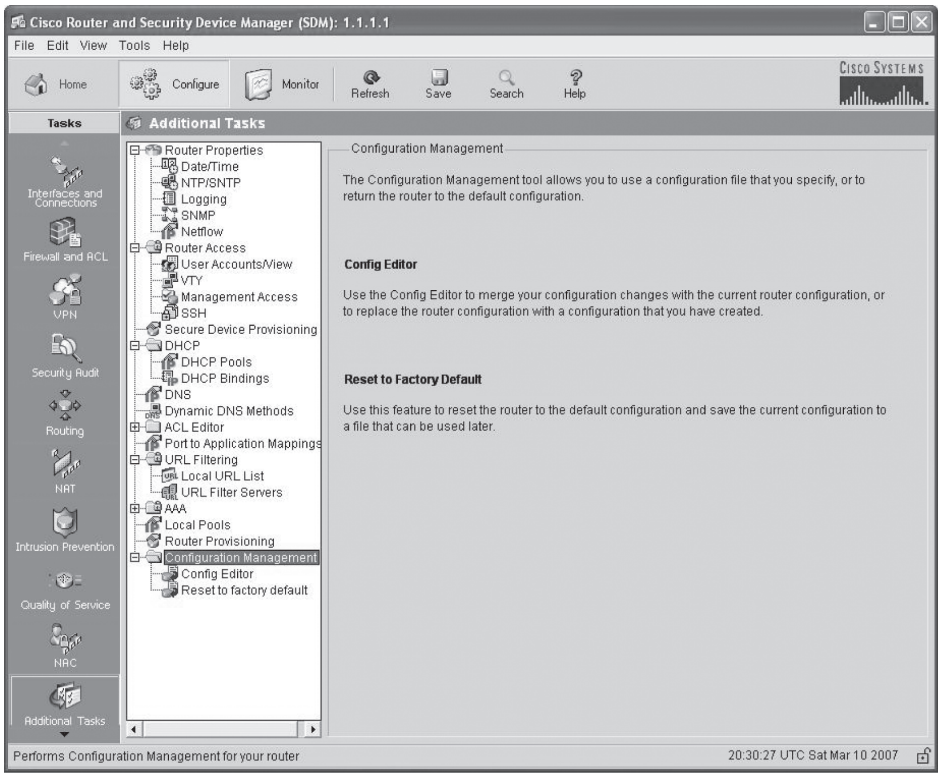
SDM'in, host'unuzdan, konfigürasyonunuzu nasıl yedekleyip geri yükleyebileceğine hızlı bir göz atalım. Ana menüden, File – Write to Startup Config to back up your configuration to NVRAM 'ı seçin.



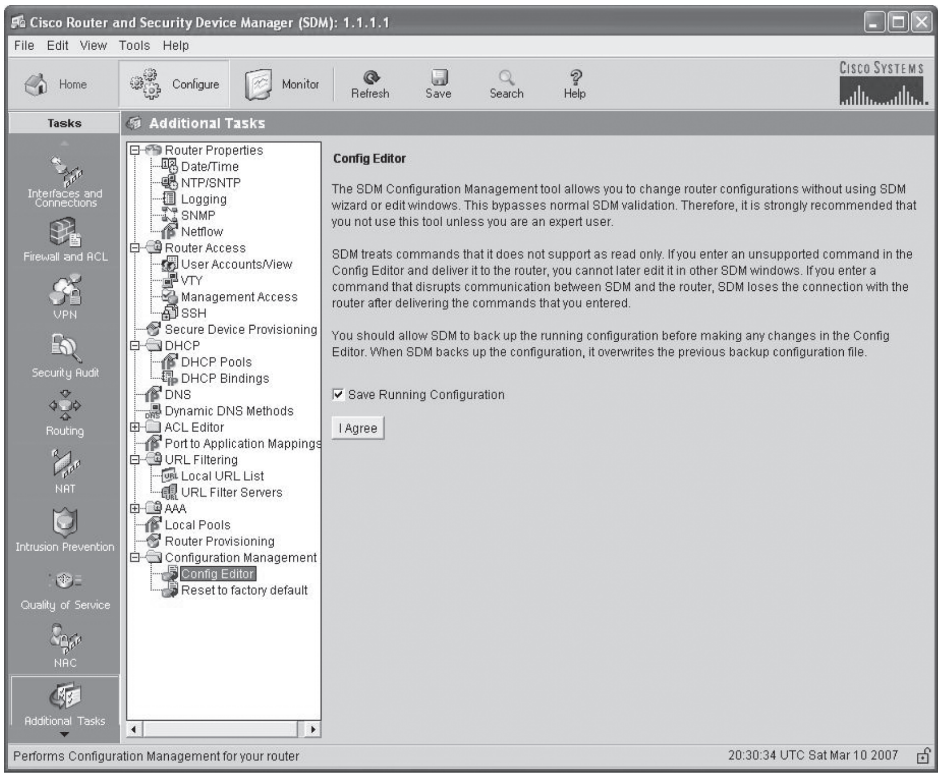

Sonra, File – Save Running Config to PC'ı seçin.



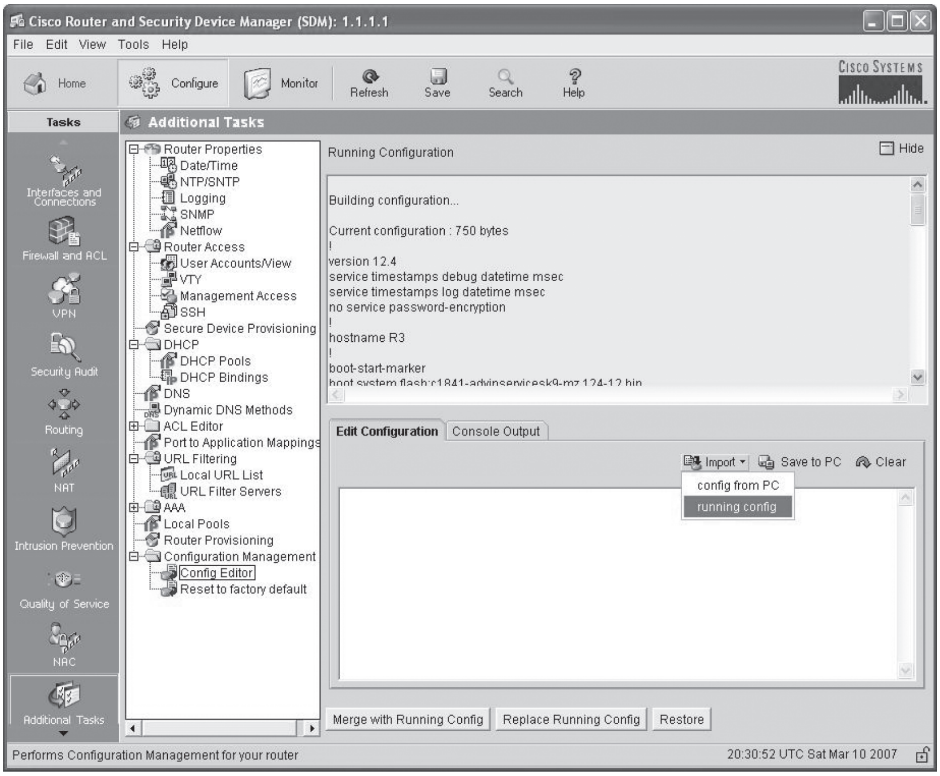
Dosyalarınızı yönetmek için son bir seçenek, Additional Tasks altında Configuration Management ekranlarını kullanmaktır.



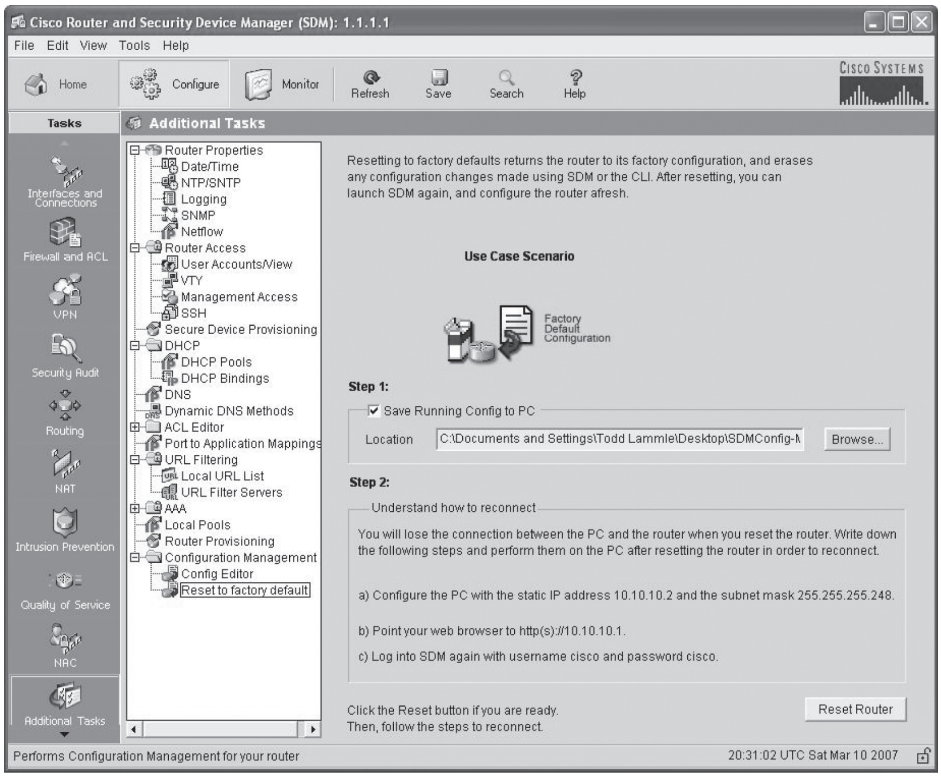

ConfigEditor, running-config'i değiştirmenizi sağlar, fakat bunu yapmanıza izin vermeden önce, router'ınızın konfigürasyonunu bozabileceğinizi göze almanız gerekmektedir!



En iyisi Save Running Config düğmesine basmaktır. Sonra, dosyayı RAM'de veya PC'nizde import etmeyi seçebilirsiniz.




Son olarak, Configuration Management'tan, Reset to Factory Default'ı seçebilirsiniz. Bu, router'da tekrar HTTPS management'ını yerleştirecektir.



Görebileceğiniz gibi, berbat etmenin birçok yöntemi vardır, flash, NVRAM ve hatta RAM'deki dosyaları kopyalamanın demek istiyorum! Bölüm 14'te gösterdiğim SDM demosu ile pratik yapın veya arkadaşınızın router'ını borç almaya çalışın. (Bu komutları kendi router'ınız üzerinde test etmek istemezsiniz, değil mi?)

# Cisco Discovery Protocol (CDP) Kullanmak

Cisco Discovery Protocol (CDP); yöneticilerin, uzak veya lokal olarak bağlı cihazlar hakkında bilgi toplamasına yardımcı olması için Cisco tarafından tasarlanan, tescilli bir protokoldür. CDP kullanarak, komşu cihazların donanım ve protokol bilgilerini toplayabilirsiniz. Bunlar, hata tespiti ve network ile ilgili doküman oluşturmak için çok kullanışlıdır.

Şimdiki bölümde, ağınızın çalıştığını kontrol etmek için kullanılan CDP timer ve CDP komutlarından bahsedeceğim.

## CDP Timers ve Holdtime Bilgilerine Ulaşmak

`sh cdp` komutu (kısaca `sh cdp`), Cisco cihazlarda yapılandırılabilen iki CDP global parametresi hakkında bilgi verir:

- CDP timer, CDP paketlerinin, tüm aktif interface'lere hangi sıklıkta aktarılacağını belirler.
- CDP holdtime, cihazın, komşulardan aldığı paketleri ne kadar tutacağını belirler.

Cisco router ve switch'ler aynı parametreleri kullanır.

NOT *Bu bölümde ve bölümün kalanında, 2811 router'ım, Corp ismini alacak ve R1, R2 ile R3 (R1'e iki bağlantı var) isimli router'lara dört seri bağlantı ve ap hostname'i ile 1242 access point'e bir adet FastEthernet bağlantısına sahip olacaktır.*

Corp router'ındaki çıktı şöyledir:

```
Corp#sh cdp
Global CDP information:
        Sending CDP packets every 60 seconds
        Sending a holdtime value of 180 seconds
        Sending CDPv2 advertisements is  enabled
```

Bir router'daki CDP holdtime ve timer'ı yapılandırmak için, `cdp holdtime` ve `cdp timer global` komutlarını kullanın:

```
Corp(config)#cdp ?
  advertise-v2      CDP sends version-2 advertisements
  holdtime          Specify the holdtime (in sec) to be sent in
                    packets
  log               Log messages generated by CDP
  run               Enable CDP
  source-interface  Insert the interface's IP in all CDP packets
  timer             Specify rate (in sec) at which CDP packets
                    are sent  run
Corp(config)#cdp holdtime ?
  <10-255>  Length  of time  (in sec) that receiver must keep
this packet
Corp(config)#cdp timer ?
  <5-254>  Rate at which CDP packets are sent (in  sec)
```

CDP'yi, router'ın global configuration modundan `no cdp run` yazarak tamamıyla kapatabilirsiniz. Bir interface için CDP'yi kapatmak veya açmak için `no cdp enable` ve `cdp enable` komutlarını kullanın. Biraz sabırlı olun, bunları birazdan kullanacağız.

## Neighbor Bilgilerini Toplamak

`show cdp neighbor` komutu (kısaca `sh cdp nei`), direkt bağlı cihazlar hakkındaki bilgileri iletir. CDP paketlerinin, bir Cisco switch'ten geçemeyeceğini ve bu yüzden sadece direkt bağlı olanları gördüklerini anlamanız önemlidir. Yani, eğer router'ınıza bir switch bağlıysa, bu switch'e bağlı herhangi bir cihazı göremezsiniz.

Aşağıda, ISR router'ımda kullanılan, `show cdp neighbor` komutunun çıktısını görebilirsiniz:

```
Corp#sh cdp neighbors
Capability Codes: R - Router, T - Trans Bridge, B - Source Route
Bridge
                  S - Switch, H - Host, I - IGMP, r - Repeater
Device ID  Local Intrfce   Holdtme  Capability  Platform   Port ID
ap         Fas 0/1         165         T I      AIR-AP124  Fas 0
R2         Ser 0/1/0       140        R S I     2801       Ser 0/2/0
R3         Ser 0/0/1       157        R S I     1841       Ser 0/0/1
R1         Ser 0/2/0       154        R S I     1841       Ser 0/0/1
R1         Ser 0/0/0       154        R S I     1841       Ser 0/0/0
Corp#
```

Corp ISR router'ına bir konsol kablosuyla direkt olarak bağlıyız ve router, dört cihaza direkt olarak bağlıdır. R1 router'ına iki bağlantımız vardır. Cihaz ID'si, bağlı cihazın yapılandırılmış hostname'ini gösterir. Lokal interface, bizim interface'imizdir ve port ID'si, uzak cihazın direkt bağlı interface'idir. Tüm gördüğünüz, direkt olarak bağlı cihazlardır.

Tablo 5.5, her cihaz için `sh cdp neighbor` ile görüntülenen bilgiyi özetlemektedir.

**Tablo 5.5:** `show cdp neighbor` Komutu

| Alan | Açıklama |
|---|---|
| `Device ID` | Direkt bağlı cihazın hostname'i. |
| `Local Interface` | CDP paketini aldığınız port veya interface. |
| `Holdtime` | Alınacak CDP paketleri olmadığında, atmadan önce router'ın bilgiyi tutma süresi. |
| `Capability` | Router, switch veya repeater gibi komşunun kapasitesi. Kapasite kodları, komut çıktısının yukarısında listelenmiştir. |
| `Platform` | Direkt bağlı Cisco cihazın türü. Önceki çıktıda, bir Cisco 2500 router ve Cisco1900 switch, 2509 router'a direkt bağlıdır. 2509, sadece 1900 switch'i ve serial0 interface'inden bağlı 2500 router'ı görebilir. |
| `Port ID` | CDP paketlerinin multicast edildiği, komşu cihazın port ve interface'i. |

NOT

*Bir show cdp neighbors komutu çıktısına bakabilmeniz ve komşunun cihazını (kapasitesini, router veya switch olduğunu), model numarasını (platform), bu cihaza bağlandığınız portunuzu (lokal interface) ve komşunun size bağlandığı portu yorumlayabilmeniz çok önemlidir.*

Komşu bilgilerinin iletildiği diğer komut, `show cdp neighbors detail` komutudur (kısaca, `sh cdp nei de`). Bu komut, hem router hem de switch'lerde kullanılabilir ve komutu çalıştırdığınız cihaza bağlı her cihaz hakkında detaylı bilgileri görüntüler. Örnek olarak, şu router çıktısına bakın:

```
Corp#sh cdp neighbors detail
- - - - - - - - - - - - - - - - - - - -
Device ID: ap
Entry address(es): 10.1.1.2
Platform: cisco AIR-AP1242AG-A-K9,Capabilities: Trans-Bridge IGMP
Interface: FastEthernet0/1,  Port ID (outgoing port):
FastEthernet0
Holdtime : 122 sec

Version :
Cisco IOS Software, C1240 Software (C1240-K9W7-M), Version
12.3(8)JEA,
    RELEASE SOFTWARE (fc2)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2006 by Cisco Systems, Inc.
Compiled Wed 23-Aug-06 16:45 by kellythw

advertisement version: 2
Duplex: full
Power drawn: 15.000 Watts
- - - - - - - - - - - - - - - - - - -
Device ID: R2
Entry address(es):
  IP address: 10.4.4.2
Platform: Cisco 2801,  Capabilities: Router Switch IGMP
Interface: Serial0/1/0,  Port ID (outgoing port): Serial0/2/0
Holdtime : 135 sec

Version :
Cisco IOS Software, 2801 Software (C2801-ADVENTERPRISEK9-M),
    Experimental Version 12.4(20050525:193634) [jezhao-ani 145]
Copyright (c) 1986-2005 by Cisco Systems, Inc.
Compiled Fri 27-May-05 23:53 by jezhao

advertisement version: 2
VTP Management Domain:''
- - - - - - - - - - - - - - - - - - -
Device ID: R3
Entry address(es):
  IP address: 10.5.5.1
Platform: Cisco 1841,  Capabilities: Router Switch IGMP
Interface: Serial0/0/1,  Port ID (outgoing port): Serial0/0/1
Holdtime : 152 sec
```

```
Version :
Cisco IOS Software, 1841 Software (C1841-IPBASE-M), Version
12.4(1c),
    RELEASE SOFTWARE (fc1)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2005 by Cisco Systems, Inc.
Compiled Tue 25-Oct-05 17:10 by evmiller

advertisement version: 2
VTP Management Domain:''
- - - - - - - - - - - - - - - - -
[output cut]
Corp#
```

*Sadece, direkt bağlı cihazların IP adreslerini görebileceğinizi hatırlayın.*

NOT

Burada bize ne gösterilmektedir? İlk olarak, direkt bağlı tüm cihazların IP adresleri ve hostname'leri verilmektedir. `Show cdp neighbor` komutuyla görüntülenen bilgiye ilaveten, `show cdp neighbors detail` komutu bize, komşu cihazın IOS versiyonunu verir.

`show cdp entry *` komutu, `show cdp neighbors detail` komutu ile aynı bilgileri görüntüler. Aşağıda, `show cdp entry *` komutu kullanılarak alınan bir router çıktısı görülmektedir:

```
Corp#sh cdp entry *
- - - - - - - - - - -
Device ID: ap
Entry address(es):
Platform: cisco AIR-AP1242AG-A-K9   ,  Capabilities: Trans-Bridge
IGMP
Interface: FastEthernet0/1,  Port ID (outgoing port):
FastEthernet0
Holdtime : 160 sec

Version :
Cisco IOS Software, C1240 Software (C1240-K9W7-M), Version
12.3(8)JEA,
    RELEASE SOFTWARE (fc2)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2006 by Cisco Systems, Inc.
Compiled Wed 23-Aug-06 16:45 by kellythw

advertisement version: 2
Duplex: full
Power drawn: 15.000 Watts
- - - - - - - - - - - - -
Device ID: R2
Entry address(es):
  IP address: 10.4.4.2
Platform: Cisco 2801,  Capabilities: Router Switch IGMP
 --More--
[output cut]
```

`show cdp neighbors detail` ve `show cdp entry *` komutları arasında herhangi bir farklılık yoktur. Bununla beraber `sh cdp entry *` komutu, `show cdp neighbors detail` komutunda olmayan iki seçeneğe sahiptir:

```
Corp#sh cdp entry * ?
  protocol  Protocol information
  version   Version information
  |         Output modifiers
  <cr>
Corp#show cdp entry * protocols
Protocol information for ap :
  IP address: 10.1.1.2
Protocol information for R2 :
  IP address: 10.4.4.2
Protocol information for R3 :
  IP address: 10.5.5.1
Protocol information for R1 :
  IP address: 10.3.3.2
Protocol information for R1 :
  IP address: 10.2.2.2
```

`show cdp entry * protocols` komutunun yukarıdaki çıktısı, direkt bağlı her komşunun sadece IP adresini gösterebilir. `show cdp entry * version`, direkt bağlı komşularınızın sadece IOS versiyonlarını gösterecektir:

```
Corp#show cdp entry * version
Version information for ap :
  Cisco IOS Software, C1240 Software (C1240-K9W7-M), Version
   12.3(8)JEA, RELEASE SOFTWARE (fc2)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2006 by Cisco Systems, Inc.
Compiled Wed 23-Aug-06 16:45 by kellythw

Version information for R2 :
  Cisco IOS Software, 2801 Software (C2801-ADVENTERPRISEK9-M),
   Experimental Version 12.4(20050525:193634) [jezhao-ani 145]
Copyright (c) 1986-2005 by Cisco Systems, Inc.
Compiled Fri 27-May-05 23:53 by jezhao

Version information for R3 :
  Cisco IOS Software, 1841 Software (C1841-IPBASE-M),Version 12.4(1c),
   RELEASE SOFTWARE (fc1)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2005 by Cisco Systems, Inc.
Compiled Tue 25-Oct-05 17:10 by evmiller

 --More--
[output cut]
```

`show cdp neighbors detail` ve `show cdp entry` komutları, benzer olmalarına rağmen, `show cdp entry` komutu, direkt bağlı komşuların çıktılarını tek satırda göstermenizi sağlar. Oysaki `show cdp neighbor detail` komutu bunu yapamaz. Şimdi gelin `show cdp traffic` komutuna bir bakalım.

## Interface Traffic Bilgisinin Toplanması

`show cdp traffic` komutu, gönderilen ve alınan CDP paketlerinin sayısı ve CDP ile alınan hataları içeren, interface trafiği hakkındaki bilgiyi görüntüler.

Aşağıda, Corp router'ında kullanılan `show cdp traffic` komutu çıktısı vardır:

```
Corp#sh cdp traffic
CDP counters :
        Total packets output: 911, Input: 524
        Hdr syntax: 0, Chksum error: 0, Encaps failed: 2
        No memory: 0, Invalid packet: 0, Fragmented: 0
        CDP version 1 advertisements output: 0, Input: 0
        CDP version 2 advertisements output: 911, Input: 524
Corp#
```

Bu, gerçekten bir router'dan toplayabileceğiniz en önemli bilgi değildir, fakat bir cihazda ne kadar CDP paketinin gönderildiği ve alındığını gösterir.

## Port ve Interface Bilgisini Toplamak

`show cdp interface` komutu size, router interface'leri veya switch portlarındaki CDP durumunu verir.

Daha önce söylediğim gibi, `no cdp run` komutunu kullanarak, bir router'daki CDP'yi tamamen kapatabilirsiniz. Fakat aynı zamanda, `no cdp enable` komutu ile CDP'yi interface bazında kapatabilirsiniz. Bir portu, `cdp enable` komutu ile etkinleştirebilirsiniz. Tüm port ve interface'ler varsayılan olarak, `cdp enable` olarak ayarlıdır.

Bir router'da `show cdp interface` komutu, CDP kullanarak, her interface hakkındaki bilgiyi görüntüler. Bu, bir hattaki enkapsülasyonu, her interface için timer ve holdtime'ı içermektedir. Aşağıda, ISR router'da bu komutun çıktısıyla ilgili örnek vardır:

```
Corp#sh cdp interface
FastEthernet0/0 is administratively down, line protocol is down
  Encapsulation ARPA
  Sending CDP packets every 60 seconds
  Holdtime is 180 seconds
FastEthernet0/1 is up, line protocol is up
  Encapsulation ARPA
  Sending CDP packets every 60 seconds
  Holdtime is 180 seconds
Serial0/0/0 is up, line protocol is up
  Encapsulation HDLC
  Sending CDP packets every 60 seconds
  Holdtime is 180 seconds
Serial0/0/1 is up, line protocol is up
```

```
  Encapsulation HDLC
  Sending CDP packets every 60 seconds
  Holdtime is 180 seconds
Serial0/1/0 is up, line protocol is up
  Encapsulation HDLC
  Sending CDP packets every 60 seconds
  Holdtime is 180 seconds
Serial0/2/0 is up, line protocol is up
  Encapsulation HDLC
  Sending CDP packets every 60 seconds
  Holdtime is 180 seconds
```

Yukarıdaki çıktı, daima interface'in durumu hakkında bilgi verdiği için güzeldir. Bir router'ın interface'ini kapatmak için interface configuration moddan `no cdp enable` komutunu kullanın:

```
Corp#config t
Corp(config)#int s0/0/0
Corp(config-if)#no cdp enable
Corp(config-if)#do show cdp interface
FastEthernet0/0 is administratively down, line protocol is down
  Encapsulation ARPA
  Sending CDP packets every 60 seconds
  Holdtime is 180 seconds
FastEthernet0/1 is up, line protocol is up
  Encapsulation ARPA
  Sending CDP packets every 60 seconds
  Holdtime is 180 seconds
Serial0/0/1 is up, line protocol is up
  Encapsulation HDLC
  Sending CDP packets every 60 seconds
  Holdtime is 180 seconds
Serial0/1/0 is up, line protocol is up
  Encapsulation HDLC
  Sending CDP packets every 60 seconds
  Holdtime is 180 seconds
Serial0/2/0 is up, line protocol is up
  Encapsulation HDLC
  Sending CDP packets every 60 seconds
  Holdtime is 180 seconds
Corp(config-if)#
```

Serial0/0/0'ın router çıktısında listelenmediğine dikkat edin. Bu çıktıyı almak için serial0/0/0'da `cdp enable` çalıştırmalısınız. O zaman çıktıda görünecektir:

```
Corp(config-if)#cdp enable
Corp(config-if)#^Z
Corp#
```

## Gerçek Dünya Senaryosu

### CDP Hayat Kurtarır mı?

*Karen, Dallas, Texas'taki büyük bir hastanede uzman network danışmanı olarak yeni işe alındı. Kendisinden problemlerle ilgilenmesi beklenmektedir. Network'te bir arıza oluştuğunda Karen, insanların muhtemelen iyi bir sağlık hizmeti alamayacaklarından kaygılanmalıdır. Potansiyel bir ölüm-kalım meselesinden bahsediyorum!*

*Karen işine mutlu olarak başlar. Kısa süre sonra, tabii ki ağda bazı problemler olur. Ağdaki hatayı tespit etmek için altındaki yöneticilerin birinden, ağın haritasını ister. Bu personel, onları eski uzman yöneticisinin (işten çıkartılan) bildiğini ve kimsenin bulamadığını söyler.*

*Doktorlar, hastaları ile ilgilenmeleri gereken bilgilere ulaşamadığından, kısa aralıklarda ararlar. Karen ne yapmalıdır?*

*CDP imdada yetişmektedir! Çok şükür, bu hastanede, tamamen Cisco router ve switch'ler kullanılmakta ve tüm Cisco cihazlarında, varsayılan olarak CDP etkindir. Ayrıca, şanslı bir şekilde işten çıkartılan mutsuz yönetici, ayrılmadan önce CDP'yi kapatmamıştır.*

*Şimdi Karen'in tüm yapması gereken, hastane ağını anlamasına ve hayat kurtarmasına yardımcı olacak her cihaz hakkında, ihtiyacı olan tüm bilgileri bulmak için show cdp neighbor detail komutunu kullanmaktır.*

*Ağınızda bunu çözmeye engel olabilecek tek şey, bu cihazların şifrelerini bilip bilmemenizdir. Sizin umudunuz şimdi, birilerinin giriş şifrelerini biliyor olması veya onlarda şifre kurtarma işlemi uygulamaktır.*

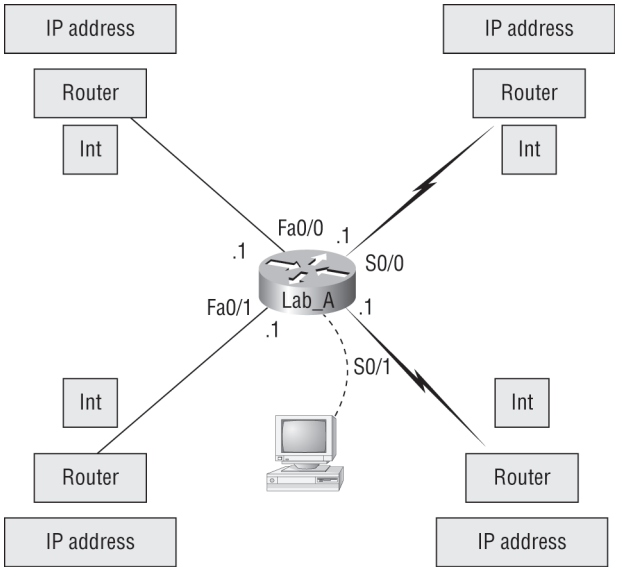
*CDP'yi kullanın. Ne zaman birinin hayatını kurtaracağınızı bilemezsiniz!*

*Bu bir gerçek hikayedir.*

## Network Topolojisini, CDP Kullanarak Belgelemek

Bu bölümün başlığının da belirttiği gibi, CDP kullanarak, örnek bir ağın nasıl belgeleneceğini göstereceğim. Sadece CDP komutları ve `show running-config` komutu kullanarak uygun router tiplerini, interface tiplerini ve farklı interfacelerin IP adreslerini belirlemeyi öğreteceğim. Ağı belgelemek için sadece Lab_A router'ına konsol bağlantısı yapabilirsiniz. Uzak router'lara, aralıktaki ardışık IP adreslerini atamalısınız. Şekil 5.2, bu dokümantasyonu tamamlamak için kullanılacaktır.

Bu çıktıda, dört interface'li bir router'ınız olduğunu görebilirsiniz: İki FastEthernet ve iki serial interface. İlk olarak, `show running-config` komutunu kullanarak, her interface'in IP adresini belirleyin:




***Şekil 5.2:*** *CDP kullanarak bir network topolojisini belgelemek.*

```
Lab_A#sh running-config
Building configuration...

Current configuration : 960 bytes
!
version 12.2
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname Lab_A
!
ip subnet-zero
!
!
interface FastEthernet0/0
 ip address 192.168.21.1 255.255.255.0
 duplex auto
!
interface FastEthernet0/1
 ip address 192.168.18.1 255.255.255.0
 duplex auto
!
interface Serial0/0
ip address 192.168.23.1 255.255.255.0
!
interface Serial0/1
ip address 192.168.28.1 255.255.255.0
!
ip classless
!
line con 0
line aux 0
line vty 0 4
!
end
```

Bu adım tamamlandığında, Lab_A router'ının dört interface'inin IP adresini yazabilirsiniz. Sonra, bu interface'lerin karşı uçlarındaki cihazların tiplerini belirlemek zorundasınız. Bunu yapmak kolaydır. Sadece `show cdp neighbors` komutunu kullanın:

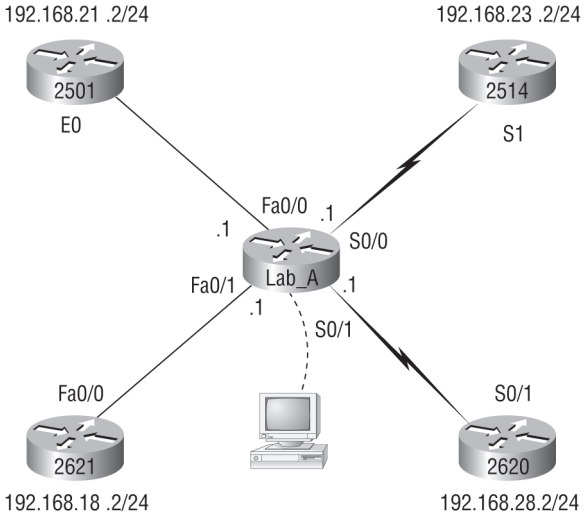
```
Lab_A#sh cdp neighbors
Capability Codes: R-Router, T-Trans Bridge, B-Source Route Bridge
  S - Switch, H - Host, I - IGMP, r - Repeater
```

```
Device ID   Local Intrfce   Holdtme   Capability   Platform   Port ID
  Lab_B        Fas 0/0        178          R          2501       E0
  Lab_C        Fas 0/1        137          R          2621       Fa0/0
  Lab_D        Ser 0/0        178          R          2514       S1
  Lab_E        Ser 0/1        137          R          2620       S0/1
  Lab_A#
```

Şimdi oldukça iyi bilgilere sahipsiniz. `show running-config` ve `show cdp neighbors` komutlarının her ikisini kullanarak, Lab_A router'ının tüm IP adreslerini ve Lab_A router'ın linklerinin her birine ve uzak router'lardaki interface'lerin hepsine bağlı router tiplerini biliyorsunuz.

`show running-config` ve `show cdp neighbors` ile toplanan tüm bilgileri kullanarak, şimdi şekil 5.3'teki topolojiyi oluşturabiliriz.

Şayet ihtiyacımız olursa, komşuların IP adreslerini görmek için, `show cdp neighbors detail` komutunu da kullanabiliriz. Fakat Lab_A router'ındaki her linkin IP adresini bildiğimizden, sonraki uygun IP adresin ne olacağını zaten biliyoruz.




***Şekil 5.3:*** *Belgelenmiş network topolojisi.*

# Telnet Kullanmak

TCP/IP protokol yığınının bir parçası olan Telnet, bilgi toplamak ve programları çalıştırmak için uzak cihazlara bağlantı kurmanızı sağlayan sanal bir terminaldir.

Router ve switch'leriniz yapılandırıldıktan sonra, bir konsol kablosu kullanmaksızın, switch ve router'larınızı tekrar yapılandırmak ve/veya kontrol etmek için Telnet programını kullanabilirsiniz. Herhangi bir komut istemcisinden (DOS veya Cisco), `telnet` yazarak Telnet programını çalıştırırsınız. Bunun çalışması için router'larda VTY şifrelerine sahip olmanız gerekir.

Cihazınıza direkt bağlı olmayan router veya switch'ler hakkında bilgi toplamak için CDP'yi kullanamayacağınızı hatırlayın. Fakat komşu cihazlarınıza bağlanmak için Telnet uygulamasını kullanabilir ve sonra, onlardaki bilgileri almak için bu uzak cihazlarda CDP'yi çalıştırabilirsiniz.

Herhangi router istemcisinden telnet komutunu şu şekilde çalıştırabilirsiniz:

```
Corp#telnet 10.2.2.2
Trying 10.2.2.2 ... Open

Password required, but none set

[Connection to 10.2.2.2 closed by foreign host]
Corp#
```

Görebileceğiniz gibi, şifrelerimi ayarlamadım (nasıl utanç verici!). Bir router'daki VTY portlarının, `login` ile yapılandırıldığını hatırlayın. Yani, ya VTY şifresi oluşturmamız ya da `no login` komutunu kullanmamız gerekir. (Şayet ihtiyacınız olursa, Bölüm 4 "Cisco's Internetworking Operating

NOT *Şayet bir cihaza telnet yapamadığınızı anlarsanız, uzak cihazda şifre oluşturulmamış olabilir. Ayrıca, bir access list'in, Telnet oturumunu engellemesi de mümkündür.*

System (IOS) ve Security Device Manager'da (SDM) şifreleri ayarlamaya tekrar göz atabilirsiniz.)

Bir Cisco router'da, telnet komutunu kullanmaya ihtiyacınız yoktur; bir komut istemcisinden bir IP adresi yazabilirsiniz, router cihaza telnet yapmayı istediğinizi düşünecektir. Aşağıda, sadece IP adresi kullanarak bunun nasıl yapıldığını göstermektedir:

```
Corp#10.2.2.2
Trying 10.2.2.2 ... Open

Password required, but none set

[Connection to 10.2.2.2 closed by foreign host]
Corp#
```

Bu noktada, telnet yapmayı istediğim router'da bu VTY şifrelerini ayarlamak oldukça iyi bir fikirdir. Aşağıda, R1 isimli router'da ne yaptığımı görebilirsiniz:

```
R1#config t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#line vty 0 ?
  <1-807>  Last Line number
  <cr>
R1(config)#line vty 0 807
R1(config-line)#password telnet
R1(config-line)#login
R1(config-line)#^Z
```

Gelin bunu bir daha deneyelim. Burada, Corp ISR konsolundan router'a bağlanıyorum:

```
Corp#10.2.2.2
Trying 10.2.2.2 ... Open

User Access Verification

Password:
R1>
```

VTY şifresinin, enable-mode değil de user-mode şifresi olduğunu hatırlayın. Router R1'e telnet yaptıktan sonra, priveleged moda gitmeye çalıştığımda ne olduğuna bakın:

```
R1>en
% No password set
R1>
```

O, aslında "asla" diyor! Bu gerçekten, iyi bir güvenlik özelliğidir. Çünkü herhangi birinin cihazınıza telnet yapmasını ve sadece `enable` yazarak privileged moda geçebilmesini istemezsiniz. Uzak cihazları yapılandırmada Telnet'i kullanmak için enable-mode şifresi veya enable secret şifresini oluşturmanız gerekmektedir.

Aşağıdaki örnekte, birçok cihaza eşzamanlı telnetin nasıl yapıldığını ve sonra, IP adresi yerine hostname'in nasıl kullanılacağını göstereceğim.

*Uzak bir cihaza telnet yaptığınızda, varsayılan olarak, konsol mesajlarını görmeyeceksiniz. Örnek olarak, debugging çıktısını görmeyeceksiniz. Telnet oturumunuza konsol mesajlarının gönderilmesine izin vermek için, terminal monitor komutunu kullanın.*

NOT

## Birçok Cihaza Eşzamanlı Telnet Yapmak

Şayet bir router veya switch'e telnet yaptıysanız, herhangi bir zaman **exit** yazarak bağlantıyı sonlandırabilirsiniz. Şayet, orijinal router konsolunuza geri dönerken, uzak cihaza bağlantınızı korumak isterseniz? Bunu yapmak için Ctrl+Shift+6 tuş kombinasyonuna basın, bırakın ve sonra X tuşuna basın.

Aşağıda, Corp router konsolumdan, birçok cihaza bağlantıyla ilgili örnek vardır:

```
Corp#10.2.2.2
Trying 10.2.2.2 ... Open

User Access Verification

Password:
R1>Ctrl+Shift+6
Corp#
```

Bu örnekte, R1 router'ına telnet yaptım ve sonra user moda girmek için şifreyi yazdım. Sonra Ctrl+Shift+6 tuş kombinasyonuna, ondan sonra da X tuşuna bastım (fakat ekran çıktısında görünmediğinden, bunu göremezsiniz). Komut istemcimin şimdi, Corp router'da olduğuna dikkat edin.

Şimdi bazı doğrulama komutlarına bakalım.

## Telnet Bağlantılarını Kontrol Etmek

Router'ınızdan uzak bir cihaza yapılan bağlantıları görmek için `show sessions` komutunu kullanın:

```
Corp#sh sessions
Conn Host               Address            Byte  Idle Conn Name
   1 10.2.2.2           10.2.2.2              0     0 10.2.2.2
*  2 10.1.1.2           10.1.1.2              0     0 10.1.1.2
Corp#
```

Connection2'nin yanındaki asterik (*)'i gördünüz mü? Bunun anlamı, 2. oturumun sizin son oturumunuz olduğudur. Enter'a iki defa basarak, son oturumunuza geri dönebilirsiniz. Ayrıca, herhangi bir oturumunuza, bağlantının numarası ve Enter'a basarak geri dönebilirseniz.

Telnet Kullanıcılarını Kontrol Etmek

Router'ınızda kullanımda olan aktif konsol ve VTY portlarını, `show users` komutu ile listeleyebilirsiniz:

```
Corp#sh users
    Line       User       Host(s)              Idle       Location
*  0 con 0                10.1.1.2             00:00:01
                          10.2.2.2             00:01:06
```

Komut çıktısında, con, lokal konsolu belirtir. Bu örnekte konsol, iki uzak IP adresine, başka bir deyişle iki cihaza bağlıdır. Sonraki örnekte, Corp router'ının telnet yaptığı ve line1 üzerinden bağlandığı ap cihazında `sh users` yazdım:

```
Corp#sh sessions
Conn Host                Address             Byte  Idle Conn Name
   1 10.1.1.2            10.1.1.2               0     0 10.1.1.2
*  2 10.2.2.2            10.2.2.2               0     0 10.2.2.2
Corp#1
[Resuming connection 1 to 10.1.1.2 ... ]
ap>sh users
    Line       User       Host(s)              Idle       Location
*  1 vty 0                idle                 00:00:00 10.1.1.1
ap>
```

Bu çıktı, konsolun aktif olduğunu ve VTY port1'in kullanıldığını gösterir. Asterisk, `show user` komutunun girildiği aktif terminal oturumunu belirtmektedir.

## Telnet Oturumlarını Sonlandırmak

Telnet oturumlarınızı, birkaç farklı yöntemle sonlandırabilirsiniz. `exit` veya `disconnect` yazmak muhtemelen en basit ve hızlı olanıdır.

Uzak bir cihazdan oturumu sonlandırmak için `exit` komutunu kullanın:

```
ap>exit
[Connection to 10.1.1.2 closed by foreign host]
Corp#
```

AP cihazı benim son oturumum olduğundan, bu oturuma dönmek için sadece iki defa Enter'a bastım.

Lokal bir cihazda oturumu sonlandırmak için `disconnect` komutunu kullanın:

```
Corp#sh session
Conn Host                Address             Byte  Idle Conn Name
   2 10.2.2.2            10.2.2.2               0     0 10.2.2.2
Corp#disconnect ?
  <0-0>  The number of an active network connection
  qdm    Disconnect QDM web-based clients
  ssh    Disconnect an active SSH connection
Corp#disconnect 2
Closing connection to 10.2.2.2 [confirm][enter]
Corp#
```

Bu örnekte, sonlandırmak istediğim R1 router'ına bağlı oturum olmasından dolayı, oturum numarası 2'yi kullandım. Gösterdiğim gibi, bağlantı numaralarını görmek için, `show sessions` komutunu kullanabilirsiniz.

Telnet üzerinden lokal cihazınıza bağlı bir cihazın oturumunu sonlandırmak isterseniz, ilk olarak router'ınıza herhangi bir cihazın telnet yapıp yapmadığını kontrol etmelisiniz. Bu bilgiyi almak için `show users` komutunu kullanın:

```
R1#sh users
    Line       User       Host(s)              Idle      Location
*  0 con 0                 idle                 00:00:00
  vty 194                  idle                 00:00:21 10.2.2.1
```

Bu çıktı, VTY'nin 10.2.2.1 IP adresine bağlı olduğunu gösterir. Bu, Corp router'ıdır. Corp router'ının, line 194'e bağlı olduğuna dikkat edin. Hangi line'a bağlanacağınızı seçemeyeceğinizi hatırlayın. Bu nedenle tüm line'lara aynı şifreyi ayarladık.

Bağlantıyı kaldırmak için `clear line#` komutunu kullanın:

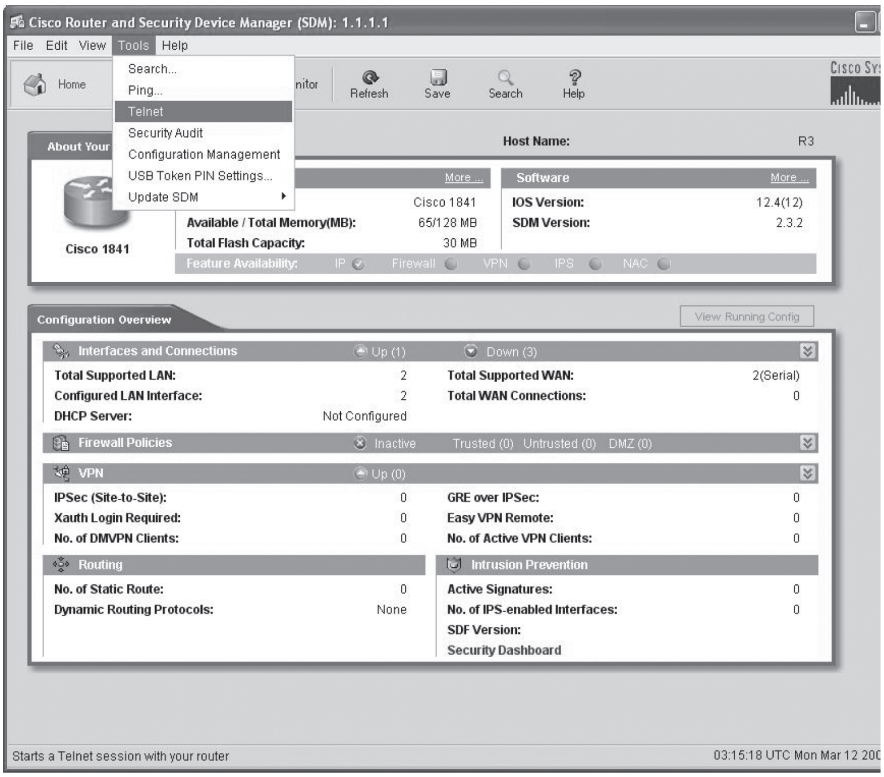
```
R1#clear line 194
[confirm][enter]
 [OK]
R1#sh users
    Line       User       Host(s)              Idle       Location
*  0 con 0                 idle                 00:00:00
```

Bu çıktı, line'nın temizlendiğini doğrular.

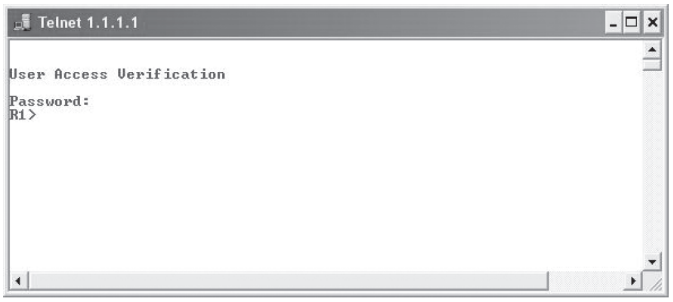
## SDM Kullanarak Router'ınıza Telnet Yapmak

SDM kullandığımızda, Telnet servisleri hakkında söyleyecek çok fazla şey yoktur. Bir menüye veya seçeneklere sahip olamazsınız. Sadece bir pop-up DOS ekranı, zaten HTTP ve HTTPS üzerinden bağlı olduğunuz router'a telnet yapar.

Tools menüsüne tıklayın ve sonra Telnet'i seçin.



Telnet'i seçince, bir DOS ekranı açılır ve user moddasınızdır (tabi ki, telnet şifrenizi girince).




SDM kullandığınızda, Telnet ile diğer seçeneklere sahip olmanız çok güzel olurdu, fakat bu durum için mümkün değildir.

# Hostname'leri Çözümlemek

Bir uzak cihaza, bağlanmak için IP adresi kullanmak yerine hostname kullanmak için bağlantı kurmaya çalıştığınız makinenin hostname'i, IP adresine çevirebilmesi gerekir.

Hostname'leri IP adreslerine çevirmenin iki yöntemi vardır: her router'da bir host tablosu oluşturmak veya bir dinamik host tablosuna benzer, Domain Name System (DNS) kurmaktır.

## Bir Host Tablosu Oluşturmak

Bir host tablosu, sadece üzerinde kurulu olduğu router'da isim çözümlemesi sağlar. Bir router'da host tablosu oluşturmak için kullanılan komut şöyledir:

```
ip host host_name tcp_port_number ip_address
```

Varsayılan, TCP port numarası 23'tür. Fakat istediğiniz farklı TCP port numarası ile Telnet kullanarak bir oturum oluşturabilirsiniz. Ayrıca, bir hostname'e sekize kadar IP adresi atayabilirsiniz.

R1 router ve ap cihazlarının isimlerini çözmek için iki girdi ile Corp router'da host tablosu oluşturmakla ilgili örnek aşağıdadır:

```
Corp#config t
Corp(config)#ip host R1 ?
  <0-65535>   Default telnet port number
  A.B.C.D     Host IP address
  additional  Append addresses
  mx          Configure a MX record
  ns          Configure an NS record
  srv         Configure a SRV record
Corp(config)#ip host R1 10.2.2.2 ?
  A.B.C.D  Host IP address
  <cr>
Corp(config)#ip host R1 10.2.2.2
Corp(config)#ip host ap 10.1.1.2
```

Yukarıdaki router konfigürasyonunda, bir host'u belirtmek için ardı ardına sekiz IP adrese kadar eklemeye devam edebileceğime dikkat edin. Yeni oluşturulan host tablosunu görmek için sadece `show hosts` komutunu kullanın:

```
Corp(config)#do show hosts
Default domain is not set
Name/address lookup uses domain service
Name servers are 255.255.255.255

Codes: UN - unknown, EX - expired, OK - OK, ?? - revalidate
       temp - temporary, perm - permanent
       NA - Not Applicable None - Not defined
Host                    Port  Flags      Age Type   Address(es)
ap                      None  (perm, OK)  0   IP    10.1.1.2
R1                      None  (perm, OK)  0   IP    10.2.2.2
Corp(config)#^Z
Corp#
```

Yukarıdaki router çıktısında, iki hostname ile onların ilgili IP adreslerini görebilirsiniz. `Flags` kolonundaki `perm`'in anlamı, girdinin manuel yapılandırıldığıdır. Şayet `temp` olarak belirtirse, DNS tarafından çözülmüş bir girdi olur.

*show hosts komutu, geçici DNS girdilerini ve ip host komutu kullanarak oluşturulan kalıcı isimden- IP adresine eşleşme bilgisini sağlar.*

NOT

Host tablosunun çözümlediği isimleri kontrol etmek için bir router komut istemcisinde, hostname'leri yazmayı deneyin. Şayet komutu belirtmezseniz, router'ın telnet istediğinizi düşüneceğini hatırlayın.

Aşağıdaki örnekte, uzak cihazlara telnet yapmak için hostname'leri kullanacağım, Ctrl+Shift+6 tuş kombinasyonuna basacağım ve sonra Corp router'ın ana konsoluna dönmek için X tuşuna basacağım:

```
Corp#r1
Trying R1 (10.2.2.2)... Open

User Access Verification

Password:
R1>Ctrl+Shift+6
Corp#ap
Trying ap (10.1.1.2)... Open

User Access Verification

Password:
ap>Ctrl+Shift+6
Corp#
```

İki cihaza oturum oluşturmak için host tablosundaki girdileri başarıyla kullandım ve iki cihaza telnet yapmak için isimleri kullandım. Host tablosundaki isimler, büyük/küçük harf duyarlılığına sahip değildir.

Aşağıdaki `show sessions` çıktısındaki girdiler, sadece IP adresleri yerine hostname'leri ve IP adreslerini görüntüler:

```
Corp#sh sessions
Conn Host            Address           Byte  Idle Conn Name
   1 r1              10.2.2.2             0     1 r1
*  2 ap              10.1.1.2             0     0 ap
Corp#
```

Bir hostname'i tablodan çıkartmak istediğinizde, `no ip host` komutunu kullanın:

```
RouterA(config)#no ip host R1
```

Host tablosu kullanmakla ilgili problem, isimleri çözümleyebilmek için her router'da bir host tablosu oluşturmaktır. Şayet çok sayıda router'a sahipseniz ve isimleri çözümlemek istiyorsanız, DNS kullanmak çok daha iyi bir çözümdür!

## İsimleri Çözümlemek İçin DNS'i Kullanmak

Şayet çok sayıda cihaza sahipseniz ve her cihazda bir host tablosu oluşturmak istemiyorsanız, hostname'leri çözümlemek için bir DNS sunucusu kullanabilirsiniz.

Bir Cisco cihazı, anlamadığı bir komutu aldığı her zaman varsayılan olarak, onu DNS üzerinden çözümlemeye çalışacaktır. Bir Cisco router'da, `todd` komutunu yazdığımda, ne olduğunu izleyin:

```
Corp#todd
Translating "todd"...domain server (255.255.255.255)
Translating "todd"...domain server (255.255.255.255)
Translating "todd"...domain server (255.255.255.255)
% Unknown command or computer name, or unable to find
  computer address
Corp#
```

Router, ismimi veya yazmaya çalıştığım komutu bilmiyor. Bu nedenle, bunu DNS yardımıyla çözmeye çalışıyor. Bu, iki sebepten dolayı gerçekten sinir bozucudur: İlki, ismimi bilmediği açıkça bellidir ve ikincisi, isim arama işlemi bitene kadar beklemek zorunda kalmamdır. Bu zaman alan DNS aramasını, global configuration moddan router'ınızda `no ip domain-lookup` komutunu kullanarak engelleyebilirsiniz.

Ağınızda bir DNS sunucusu varsa, DNS isim çözümlemesini çalışır hale getirmek için birkaç komut girmeniz gerekir:

- İlk komut, varsayılan olarak açık olan `ip domain-lookup`'tır. Sadece önceden kapatılması durumunda girilmesi gerekir. ( `no ip domain-lookup` komutu ile). Komut, tire (`-`) olmaksızın da kullanılabilir. (`ip domain lookup`).

- İkinci komut, ip name-server'dır. Bu, DNS sunucusunun IP adresini ayarlar. Altı sunucuya kadar IP adresi girebilirsiniz.

- Son komut, `ip domain-name`'dir. Bu komut, seçime bağlı olduğu halde, aslında ayarlanmalıdır. Domain adını, yazdığınız hostame'e ekler. DNS, bir fully qualified domain name (FQDN) sistemi kullandığından, `domain.com` formunda, tam bir DNS ismine sahip olmalısınız.

Aşağıda, bu üç komutun kullanıldığı bir örnek bulabilirsiniz:

```
Corp#config t
Corp(config)#ip domain-lookup
Corp(config)#ip name-server ?
  A.B.C.D  Domain server IP address (maximum of 6)
Corp(config)#ip name-server 192.168.0.70
Corp(config)#ip domain-name lammle.com
Corp(config)#^Z
Corp#
```

DNS konfigürasyonları ayarlandıktan sonra, DNS sunucunuzu bir cihaza ping atmak veya telnet yapmak için hostname kullanarak test edebilirsiniz:

```
Corp#ping R1
Translating "R1"...domain server (192.168.0.70) [OK]
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.2.2.2, timeout is
  2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max
  = 28/31/32 ms
```

İsmi çözümlemek için router'ın, DNS sunucusunu kullandığına dikkat edin.

DNS kullanarak bir isim çözümlendikten sonra, cihazın, host tablosundaki bu bilgiyi önbellekte tuttuğunu görmek için `show hosts` komutunu kullanın:

```
Corp#sh hosts
Default domain is lammle.com
Name/address lookup uses domain service
Name servers are 192.168.0.70
Host                  Flags      Age Type   Address(es)
R1                    (temp, OK)  0   IP     10.2.2.2
ap                    (perm, OK)  0   IP     10.1.1.2
Corp#
```

Çözümlenmiş girdi, `temp` olarak görünmektedir, fakat ap, hala `perm`'dir, yani bu bir statik girdidir. Hostname'in full domain name olduğuna dikkat edin. Şayet, `ip domain-name lammle.com` komutunu kullanmamış olsaydım, `ping 21.lammle.com` yazmam gerekecekti.

## Gerçek Dünya Senaryosu

### Bir Host Tablosu veya DNS Sunucusu Kullanmalı mısınız?

*Karen sonunda, CDP kullanarak ve doktorların sıkıştırmasıyla, ağın haritasını çıkarmayı tamamlamıştır. Bununla beraber Karen, ağı yönetirken zorlanmaktadır. Çünkü uzak bir router'a telnet yapması gerektiğinde, her seferinde IP adresini bulmak için network çizimine bakmak zorunda kalmaktadır.*

*Karen, her router'a host tabloları koymayı düşünmektedir, fakat yüzlerce router olduğundan, bu ürkütücü bir işlemdir.*

*Birçok network artık bir şekilde DNS sunucusuna sahiptir. Kullanmak için en kolay yol budur. Gerçekten, her router'a bu hostname'leri eklemekten çok daha kolaydır. Karen, her router'da sadece üç komut ekleyecek, o kadar. Artık isimleri çözümleyebilir.*

*Bir DNS sunucusu kullanmak, eski girdileri de güncellemeyi kolaylaştırır. Hatırlayın, şayet statik host tablosu kullanıyorsa, en küçük bir değişiklikte dahi her router'da mauel olarak güncelleme yapmak zorundadır.*

*Aklınızda olsun, bunun ağa isim çözümlemesi ile bir ilgisi yoktur ve ağdaki host'un yapmaya çalıştığıyla da ilgisi yoktur. Bu sadece, router konsolundan isimleri çözümlemeye çalıştığınızda kullanılmaktadır.*

# Network Bağlanırlığını Kontrol Etmek ve Hata Tespiti Yapmak

Uzak cihazlara bağlanabilirliği test etmek için, `ping` ve `traceroute` komutlarını kullanabilirsiniz ve her ikisi de, sadece IP ile değil, birçok protokol ile kullanılabilmektedir. Fakat `show ip route` komutunun, routing tablonuzu doğrulamak için iyi bir hata tespiti komutu olduğunu ve `show interfaces` komutunun size, her interface'in durumunu göstereceğini unutmayın.

Bölüm 4'te zaten bahsettiğimden, `show interfaces` komutuna burada girmeyeceğim. Fakat bir router'da hata tespiti için ihtiyacınız olan, `debug` ve `show processes` komutlarını anlatacağım.

## Ping Komutunu Kullanmak

Şimdiye kadar, IP bağlanırlığını test etmek için cihazları pinglemek ve DNS kullanarak isim çözümlemesi yapmak ile ilgili birçok uygulama gördünüz. Ping programı ile kullanabileceğiniz diğer tüm protokolleri görmek için `ping ?` yazın:

```
Corp#ping ?
  WORD  Ping destination address or hostname
  clns  CLNS echo
  ip    IP echo
  srb   srb echo
  tag   Tag encapsulated IP echo
  <cr>
```

`Ping` çıktısı, ping paketinin, belirli bir sistemi bulması ve geri dönmesi için geçen minimum, ortalama ve maksimum zamanları görüntüler. İşte bir örnek:

```
Corp#ping R1
Translating "R1"...domain server (192.168.0.70)[OK]
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.2.2.2, timeout
  is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max
  = 1/2/4 ms
Corp#
```

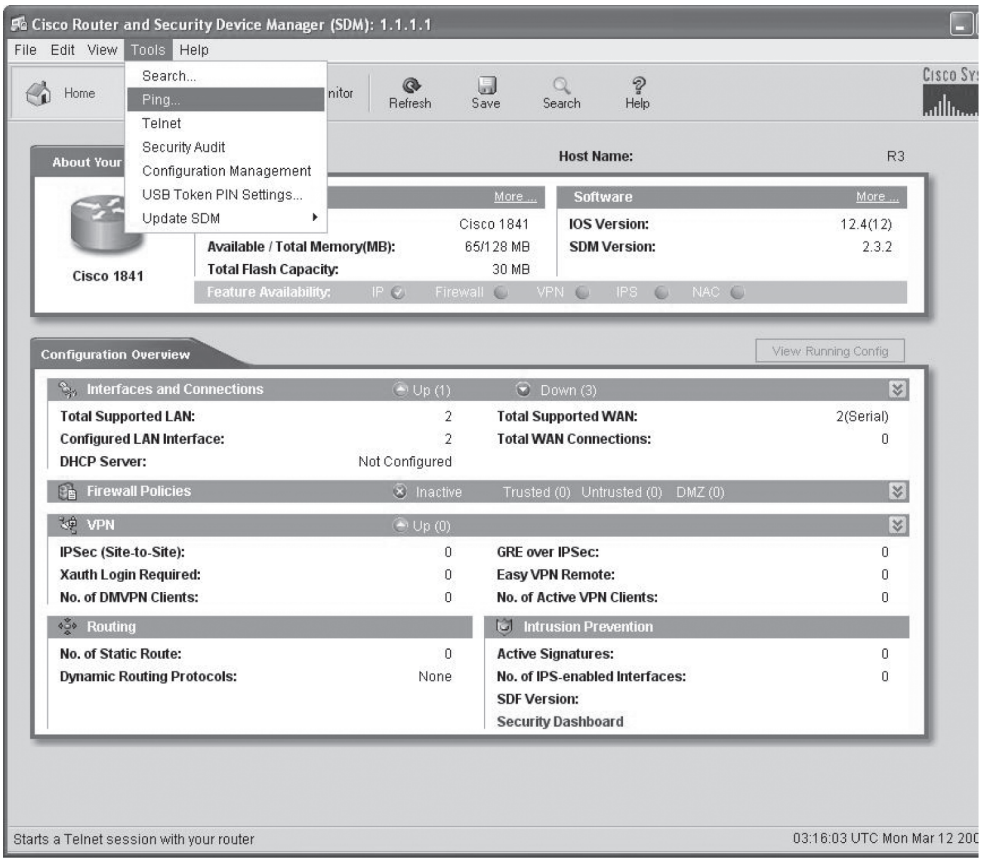
DNS sunucusunun, isimleri çözümlemek için kullanıldığını ve cihazın, minimum 1 ms'de (milisaniyede), ortalama 2ms'de ve maksimum 4ms'de pinglendiğini görebilirsiniz.

*ping komutu, user ve privileged modda kullanılabilir, fakat configuration modda kullanılamaz.*

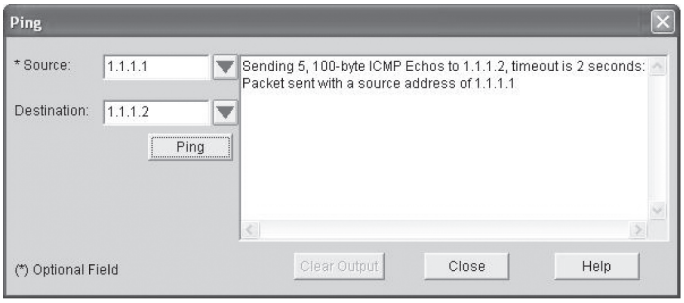
NOT

## SDM ile Pinglemek

SDM'deki Telnet seçeneğinin aksine, kullanabileceğimiz bir veya iki ekran seçeneğimiz vardır.




Tools > Ping'i seçtiğinizde, Ping ekranı görünür.




Buradan, ping atılacak kaynak interface seçilebilir. Hedefinizi girin ve Ping'e tıklayın.

## *traceroute* Komutunu Kullanmak

`traceroute` (`traceroute` komutu veya kısaca `trace`), bir paketin, uzak bir cihaza giderken geçtiği yolu gösterir. Paketin bir ağ topluluğunda uzak bir host'a ulaşmak için kat ettiği yolun ana hattını çıkarmak için time to live (TTL) time-out'lar ve ICMP hata mesajları kullanır.

User veya privileged moddan kullanılabilen *trace* (`trace` komutu), erişilemeyen bir ağa giden yoldaki hangi routerın, ağın çalışmamasının nedeni olarak yakından incelenmesi gerektiğini anlamanızı sağlar.

`Traceroute` komutu ile kullanabileceğiniz protokolleri görmek için `traceroute ?` yazın:

```
Corp#traceroute ?
  WORD       Trace route to destination address or hostname
  appletalk  AppleTalk Trace
  clns       ISO CLNS Trace
  ip         IP Trace
  ipv6       IPv6 Trace
  ipx        IPX Trace
  <cr>
```

`Trace` komutu, uzak bir cihaza giden yolda bir paketin uğradığı hop veya hop'ları gösterir.

```
Corp#traceroute r1

Type escape sequence to abort.
Tracing the route to R1 (10.2.2.2)

  1 R1 (10.2.2.2) 4 msec *  0 msec
Corp#
```

NOT *Kafanız karışmasın! Tracert komutunu kullanamazsınız, o bir Windows komutudur. Bir router için traceroute komutunu kullanın!*

Paketin, hedefi bulmak için sadece bir hop'a uğradığını görebilirsiniz.

Aşağıda, bir Windows DOS istemcisinden tracert kullanımıyla ilgili bir örnek vardır. (komutun `tracert` olduğuna dikkat!):

```
C:\>tracert www.whitehouse.gov

Tracing route to a1289.g.akamai.net [69.8.201.107]
over a maximum of 30 hops:

  1     *        *        *     Request timed out.
  2    53 ms    61 ms    53 ms  hlrn-dsl-gw15-207.hlrn.qwest.net
         [207.225.112.207]
  3    53 ms    55 ms    54 ms  hlrn-agw1.inet.qwest.net
[71.217.188.113]
  4    54 ms    53 ms    54 ms  hlr-core-01.inet.qwest.net
[205.171.253.97]
  5    54 ms    53 ms    54 ms  apa-cntr-01.inet.qwest.net
[205.171.253.26]
  6    54 ms    53 ms    53 ms  63.150.160.34
  7    54 ms    54 ms    53 ms  www.whitehouse.gov [69.8.201.107]

Trace complete.
```

Şimdi, ağınızda, debug komutu kullanarak nasıl hata tespiti yapılacağına geçelim.

## Debugging

Debug, Cisco IOS'un privileged exec modundan kullanılabilir. Farklı router operasyonlarını ve router'la alınan veya üretilen ilgili trafiği, artı herhangi bir hata mesajı ile ilgili bilgiyi görüntülemek için kullanılmaktadır.

Faydalı ve aydınlatıcı bir araçtır, fakat onun kullanımıyla ilgili bazı önemli hususları bilmeniz gerekmektedir. Debug'ın yüksek-öncelikli bir görev olduğu düşünülür, çünkü oldukça fazla miktarda kaynak kullanabilir ve router, debug edilen paketlerin anahtarlanması prosesine zorlanır. Bu nedenle, debug'ı sadece bir izleme aracı olarak kullanmazsınız. Yani, kısa bir süre için sadece hata tespiti aracı olarak kullanılır. Bunu kullanarak gerçekten, çalışan veya hatalı yazılım ve/veya donanım bileşenleri hakkında, bazı önemli bilgiler öğrenebilirsiniz.

Debugging çıktısının, diğer network çıktılarına önceliğinden ve `debug all` komutunun diğer `debug` komutlarından daha fazla çıktı üretmesinden dolayı, router'ın performansını ciddi olarak düşürebilir, hatta kullanılamaz hale getirebilir. Bu nedenle her olayda, daha spesifik `debug` komutlarının kullanılması en iyisidir.

Aşağıdaki çıktıdan görebileceğiniz gibi, debugging'i user moddan etkinleştiremezsiniz, sadece privileged moddan yapabilirsiniz:

```
Corp>debug ?
% Unrecognized command
Corp>en
Corp#debug ?
  aaa                     AAA Authentication, Authorization and
Accounting
  access-expression       Boolean access expression
  adjacency               adjacency
  all                     Enable all debugging
[output cut]
```

Şayet router'ı devre dışı bırakma olanağınız varsa ve debugging yaparak eğlenmek istiyorsanız, `debug all` komutunu yazın:

```
Corp#debug all

This may severely impact network performance. Continue? (yes/
[no]):yes

All possible debugging has been turned on

2d20h: SNMP: HC Timer 824AE5CC fired
2d20h: SNMP: HC Timer 824AE5CC rearmed, delay = 20000
2d20h: Serial0/0: HDLC myseq 4, mineseen 0, yourseen 0, line down
2d20h:
2d20h: Rudpv1 Sent: Pkts 0,  Data Bytes 0,  Data Pkts 0
2d20h: Rudpv1 Rcvd: Pkts 0,  Data Bytes 0,  Data Pkts 0
2d20h: Rudpv1 Discarded: 0,  Retransmitted 0
2d20h:
2d20h: RIP-TIMER: periodic timer expired
```

```
2d20h: Serial0/0: HDLC myseq 5, mineseen 0, yourseen 0, line down
2d20h: Serial0/0: attempting to restart
2d20h: PowerQUICC(0/0): DCD is up.
2d20h: is_up: 0 state: 4 sub state: 1 line: 0
2d20h:
2d20h: Rudpv1 Sent: Pkts 0,  Data Bytes 0,  Data Pkts 0
2d20h: Rudpv1 Rcvd: Pkts 0,  Data Bytes 0,  Data Pkts 0
2d20h: Rudpv1 Discarded: 0,  Retransmitted 0
2d20h: un all
All possible debugging has been turned off
Corp#
```

Bir router'da debugging'i kapatmak için debug komutunun önüne no yazın:

```
Corp#no debug all
```

Fakat ben, kısa yol kullanmak kolay olduğundan, genellikle `undebug all` komutunu kullanırım:

```
Corp#un all
```

`Debug all` komutu kullanmak yerine, spesifik komutları, sadece kısa bir süre için kullanmanın her zaman daha iyi olduğunu hatırlayın. Aşağıda bir router'da gönderilen ve alınan RIP güncellemelerinin gösterecek, `debug ip rip` komutuyla ilgili örnek vardır:

```
Corp#debug ip rip
RIP protocol debugging is on
Corp#
1w4d: RIP: sending v2 update to 224.0.0.9 via Serial0/0
(192.168.12.1)
1w4d: RIP: build update entries
1w4d:   10.10.10.0/24 via 0.0.0.0, metric 2, tag 0
1w4d:   171.16.125.0/24 via 0.0.0.0, metric 3, tag 0
1w4d:   172.16.12.0/24 via 0.0.0.0, metric 1, tag 0
1w4d:   172.16.125.0/24 via 0.0.0.0, metric 3, tag 0
1w4d: RIP: sending v2 update to 224.0.0.9 via Serial0/2
(172.16.12.1)
1w4d: RIP: build update entries
1w4d:   192.168.12.0/24 via 0.0.0.0, metric 1, tag 0
1w4d:   192.168.22.0/24 via 0.0.0.0, metric 2, tag 0
1w4d: RIP: received v2 update from 192.168.12.2 on Serial0/0
1w4d:    192.168.22.0/24 via 0.0.0.0 in 1 hops
Corp#un all
```

debug komutunun etkili bir komut olduğunu görebileceğinize eminim. Ve bundan dolayı, herhangi bir debugging komutunu kullanmadan önce, router'ınızın kullanımını kontrol ettiğinizden emin olmanız gerektiğinin farkında olduğunuzu biliyorum. Bu önemlidir, çünkü birçok olayda, ağ topluluğunu boyunca paketlerin işlemden geçmesinde cihazların kapasitesini olumsuz etkilemek

istemezsiniz. `show processes` komutu kullanarak, belirli bir router'ın kullanım bilgilerini belirleyebilirsiniz.

*Bir uzak cihaza telnet yaptığınızda, varsayılanda, konsol mesajlarını görmeyeceğinizi hatırlayın. Örnek olarak, debugging çıktısı görmeyeceksiniz. Konsol mesajlarının, Telnet oturumunuza gönderilmesini sağlamak için terminal monitor komutunu kullanın.*

NOT

### *show processes* Komutunu Kullanmak

Önceki bölümde belirttiğim gibi, cihazlarınızda `debug` komutunu kullandığınızda, gerçekten dikkatli olmanız gerekir. Şayet, router'ınızın CPU kullanımı, sürekli olarak %50 veya üzerinde ise, router'ın arızalandığında neye benzediğini görmek istemeniz dışında, `debug all` komutunu kullanmak, muhtemelen iyi bir fikir değildir.

Peki, hangi araçları kullanabilirsiniz? `Show processes` (veya `show processes cpu`), belirtilen router'ın CPU kullanımını belirlemek için iyi bir araçtır. İlave olarak, bu size, ilgili process ID'lerini, priority'i, scheduler test'i (status), kullanılan CPU zamanını, prosesin çalıştırılma sayısı v.s. ile birlikte aktif proseslerin listesini verecektir. Bu komut, router'ınızın performansını ve CPU kullanımını değerlendirmek istediğinizde, çok kullanışlıdır.

Aşağıdaki çıktıda ne görüyorsunuz? İlk satır, son 5 saniye, 1 dakika ve 5 dakika için CPU kullanım çıktısını gösterir. Çıktı, son 5 saniye için CPU kullanımı önünde 2%/0% değerini gösterir. İlk sayı, toplam kullanıma eşittir ve ikincisi, rutinlerin kesilmesinden dolayı, kullanımı sınırlandırır:

```
Corp#sh processes
CPU utilization for five seconds: 2%/0%; one minute: 0%; five
minutes: 0%
 PID QTy PC Runtime (ms)    Invoked   uSecs    Stacks TTY Process
   1 Cwe 8034470C    0          1       0 5804/6000   0 Chunk Manager
   2 Csp 80369A88    4       1856       2 2616/3000 0 Load Meter
   3 M*          0    112     14    800010656/12000  0 Exec
   5 Lst 8034FD9C  268246    52101 5148 5768/6000   0 Check heaps
   6 Cwe 80355E5C  20          3    6666 5704/6000   0 Pool Manager
   7 Mst 802AC3C4   0          2       0 5580/6000   0 Timers
[output cut]
```

Basit olarak, `show processes` komutunun çıktısı, router'ın, aşırı yüklenme olmaksızın, debugging komutlarını prosesten geçirebileceğini gösterir.

## Özet

Bu bölümde, Cisco router'ların nasıl yapılandırıldığını ve bu konfigürasyonların nasıl yönetileceğini öğrendiniz.

Bu bölüm, router'ın ROM, RAM, NVRAM ve flash gibi iç bileşenlerini kapsadı.

İlave olarak, bir router boot olduğunda ve dosyalar yüklendiğinde ne olduğunu da anlattım. Configuration register, router'a nasıl boot edeceğini ve dosyaları nereden bulacağını söyler. Siz, configuration register ayarlarının, şifre kurtarma amacıyla, nasıl değiştirildiğini ve doğrulandığını öğrendiniz.

Sonra, hem bir Cisco IOS'unun nasıl yedeklendiğini ve geri yüklendiğini hem de bir Cisco router konfigürasyon yedeğinin nasıl alındığını ve geri yüklendiğini öğrendiniz. Bu dosyaları CLI, IFS ve SDM kullanarak nasıl yöneteceğinizi gösterdim.

Daha sonra, uzak cihazlar hakkında bilgi toplamak için CDP ve Telnet'in nasıl kullanıldığını öğrendiniz. Son olarak bölümde, hem hostname'lerin nasıl çözümlendiği ve ağın bağlanabilirliğini

test etmek için `ping` ve `trace` komutlarının nasıl kullanıldığı hem de `debug` ve `show processes` komutlarının nasıl kullanıldığı anlatıldı.

# Sınav Gereklilikleri

**Farklı configuration register komutlarını ve ayarlarını hatırlamak:** 0x2102 ayarı, tüm Cisco router'larda varsayılandır ve router'a, boot sıralaması için NVRAM'e bakmasını söyler. 0x2101 router'a, ROM'dan boot etmesini ve 0x2142 router'a, şifre kurtarması için NVRAM'deki startup-config'i yüklememesini söyler.

**Bir IOS imajın nasıl yedekleneceğini hatırlamak:** Privileged-mod `copy flash tftp` komutunu kullanarak, bir dosyayı, flash bellekten TFTP (network) sunucusuna yedekleyebilirsiniz.

**Bir IOS imajının nasıl geri yükleneceğini veya upgrade edileceğini hatırlamak:** Privileged-mode `copy tftp flash` komutunu kullanarak, bir dosyayı, bir TFTP (network) sunucusundan flash belleğe geri yükleyebilir veya upgrade edebilirsiniz.

**Bir IOS imajını, bir network sunucusuna yedek almadan önce neyi tamamlamanız gerektiğini hatırlamak:** Network sunucusuna erişebileceğinize emin olun, network sunucusunun, imaj için yeterli alana sahip olduğundan emin olun ve dosya isimlendirmesi ve yol gereksinimlerinin doğruluğunu kontrol edin.

**Bir router'ın konfigürasyonunun nasıl kaydedileceğini hatırlamak:** Bunu yapmak için birkaç yöntem vardır. Hem en yaygını hem de en çok deneneni, `copy running-config startup-config` dir.

**Bir router konfigürasyonunun nasıl silindiğini hatırlamak:** `Erase startup-config privileged-mode` komutunu yazın ve router'ı yeniden başlatın.

**CDP'nin ne zaman kullanılacağını anlamak:** Cisco Discovery Protocol'ü, hem ağın dökümanını çıkarmakta hem de ağınızda hata tespitinde kullanılabilmektedir.

`Show cdp neighbors` **komutunu çıktısının ne gösterdiğini hatırlamak:** `Show cdp neighbors` komutu, şu bilgileri sağlar: device ID'si, lokal interface, holdtime, kapasite, platform ve port ID'si (uzak interface).

**Bir router'a nasıl telnet yapılacağını ve orijinal konsolunuza dönseniz bile bağlantınızı nasıl koruyacağınızı anlamak:** Şayet bir router veya switch'e telnet yapıyorsanız, istendiğinde, exit yazarak bağlantıyı sonlandırabilirsiniz. Bununla beraber, orijinal router konsolunuza geri dönerken, uzak cihaza bağlantınızı korumak isterseniz, Ctrl+Shift+6 tuş kombinasyonuna basıp, onu bırakarak, X'e basabilirsiniz.

**Telnet oturumlarınızı doğrulayan komutu hatırlamak:** `Show session` komutu, diğer router'larla router'ınız arasındaki tüm oturumlar hakkında bilgi sağlayacaktır.

**Bir router'da statik host tablosunun nasıl oluşturulduğunu hatırlamak:** `ip host` ***`host_name ip_address`*** global configuration komutunu kullanarak, router'ınızda, statik host tablosu oluşturabilirsiniz. Aynı host girdisine, birçok IP adresi atayabilirsiniz.

**Bir router'daki host tablonuzu nasıl doğrulayabileceğinizi hatırlamak:** Host tablosunun, `show hosts` komutuyla doğruluğunu kontrol edebilirsiniz.

**Ping komutunu ne zaman kullanacağınızı anlamak:** Packet Internet Groper (Ping), bir ağdaki aktif IP adresini kontrol etmek için ICMP echo request ve ICMP echo reply'lar kullanır.

**Geçerli bir host ID'sine nasıl ping atıldığını hatırlamak:** Bir IP adresine, router'ın user veya privileged modundan (configuration moddan olmaz) ping atabilirsiniz. 1.1.1.1 gibi geçerli bir adresi pinglemelisiniz.

# Yazılı Lab 5

Aşağıdaki soruların cevaplarını yazın:

1. Bir Cisco IOS'unu TFTP sunucusuna kopyalamak için hangi komut kullanılır?
2. Bir Cisco startup-config'i, TFTP sunucusuna kopyalamak için hangi komut kullanılır?
3. Startup-config'i, DRAM'e kopyalamak için hangi komut kullanılır?
4. Startup-config'i, DRAM'e kopyalamak için kullanabileceğiniz eski komut nedir?
5. Router komut istemcisinden, komşu router'ın IP adresini görmek için kullanabileceğiniz komut nedir?
6. Bir komşu router'ın hostname'ini, lokal interface'ini, platformunu ve uzak portunu görmek için kullanabileceğiniz komut nedir?
7. Birçok cihaza eşzamanlı telnet yapmak için kullanabileceğiniz tuşlar nelerdir?
8. Hangi komut, komşuya veya uzak cihazlara aktif Telnet bağlantılarını gösterecektir?
9. Bir Cisco IOS'unu upgrade etmek için hangi komutu kullanabilirsiniz?
10. Bir yedek konfigürasyonu, RAM'deki konfigürasyonla birleştirmek için hangi komutu kullanabilirsiniz?

*(Yazılı lab5'in cevapları, bu bölüm için gözden geçirme sorularına cevaptan sonra bulunabilir.)*

# Pratik Lab'lar

Bu bölümdeki labları tamamlamak için, en az bir router'ınız (üç tane olması en iyisidir) ve TFTP sunucusu olarak çalışan en az bir PC'niz olması gerekmektedir.

*İlk iki lab'ı, SDM demosunu kullanarak çalıştırabilirsiniz ve bu lab'ların TFTP host kısımlarını atlayabilirsiniz. Bununla beraber, her iki yöntemin nasıl kullanıldığını bilmelisiniz.*

NOT

Aşağıda, bu modüldeki lab'ların bir listesi vardır:

Lab 5.1: Router IOS'unuzu yedeklemek

Lab 5.2: Router IOS'unuzu upgrade etmek ve geri yüklemek

Lab 5.3: Router konfigürasyonunu yedeklemek

Lab 5.4: Cisco Discovery Protocol'ü (CDP) kullanmak

Lab 5.5: Telnet'i kullanmak

Lab 5.6: Hostname'leri çözümlemek

## Pratik Lab 5.1: Router IOS'unuzu yedeklemek

1. Router'ınıza bağlanın ve `en` veya `enable` yazarak privileged moda girin.
2. Router konsolundan IP adresini pingleyerek, ağınızdaki TFTP sunucunuza bağlanabileceğinizden emin olun.
3. Flash belleğin içindekileri görmek için `show flash` yazın.
4. Router'da çalışan IOS'un ismini öğrenmek için privileged-modda `show version` yazın. Şayet flash bellekte sadece bir dosya varsa, `show flash` ve `show version` komutları, aynı dosyayı gösterir. `Show version` komutunun, aktif olarak çalışan dosyayı ve `show flash` komutunun, flash bellekteki tüm dosyaları gösterdiğini hatırlayın.
5. TFTP sunucusuna Ethernet bağlantınızın iyi olduğundan eminseniz ve IOS'un dosya ismini biliyorsanız, `copy flash tftp` yazarak IOS'unuzu yedekleyin. Bu komut, router'a flash

belleğin (bu, IOS'un varsayılan olarak tutulduğu yerdir) içindekileri bir TFTP sunucusuna kopyalamasını söyler.

6. TFTP sunucusunun IP adresini ve kaynak IOS dosya adını girin. Dosya şimdi, TFTP sunucusunun varsayılan dizinine kopyalanıp, saklanmaktadır.

## Pratik Lab 5.2: Router IOS'unuzu Upgrade Etmek ve Geri Yüklemek

1. Router'ınıza bağlanın ve en veya `enable` yazarak privileged moda girin.
2. Router konsolundan IP adresini pingleyerek, ağınızdaki TFTP sunucunuza bağlanabileceğinizden emin olun.
3. TFTP sunucusuna, Ethernet bağlantınızın iyi olduğundan emin olunca, `copy tftp flash` komutunu çalıştırın.
4. Router konsolundan sağlanan istekleri takip ederek, geri yükleme ve upgrade esnasında router'ın başka bir işlem yapmadığını kontrol edin.
5. TFTP sunucusunun IP adresini girin.
6. Geri yüklemeyi veya upgrade etmeyi istediğiniz IOS dosyasının adını girin.
7. Flash belleğin içindekilerin silineceğini anladığınızdan emin olun.
8. IOS'unuzun flash bellekten silindiğini ve yeni IOS'unuzun flash belleğe kopyalanmasının güzelliğini izleyin.

Şayet, flash bellekteki dosya silindiyse ve yeni versiyon, flash'a kopyalanmadıysa router, ROM monitor moddan boot edecektir. Kopyalama işleminin neden gerçekleşmediğini anlamanız gerekir.

## Pratik Lab 5.3: Router Konfigürasyonunun Yedeklenmesi

1. Router'ınıza bağlanın ve en veya `enable` yazarak privileged moda girin.
2. IP bağlantınız olduğundan emin olmak için TFTP sunucusunu pingleyin.
3. RouterB'den `copy run tftp` yazın.
4. TFTP sunucusunun IP adresini (örneğin 172.16.30.2) yazın ve Enter'a basın.
5. Router, sizden bir dosya adı isteyecektir. Router'ın dosya adı, `-confg` sonekiyle devam eder. İstediğiniz bir ismi kullanabilirsiniz.

```
Name of configuration file to write [RouterB-confg]?
```

Varsayılan ismi kabul etmek için Enter'a basın.

```
Write file RouterB-confg on host 172.16.30.2? [confirm]
```

Enter'a basın.

## Pratik Lab 5.4: Cisco Discovery Protocol (CDP) Kullanmak

1. Router'ınıza bağlanın ve en veya `enable` yazarak privileged moda girin.
2. Router'dan, `sh cdp` yazın ve Enter'a basın. CDP paketlerinin, her 60 saniyede tüm aktif interface'lere gönderildiğini ve holdtime'ın 180 saniye olduğunu görmelisiniz (bunlar varsayılan değerlerdir).
3. CDP güncelleme periyodunu, 90 saniye olarak değiştirmek için global configuration modda, `cdp timer 90` yazın.

```
RouterC#config t
Enter configuration commands, one per line.  End with
   CNTL/Z.
RouterC(config)#cdp timer ?
   <5-900>  Rate at which CDP packets are sent (in sec)
RouterC(config)#cdp timer 90
```

4. Privileged modda `show cdp` komutunu yazarak, CDP timer frekansınızın değiştiğini kontrol edin.

```
RouterC#sh cdp
Global CDP information:
Sending CDP packets every 90 seconds
Sending a holdtime value of 180 seconds
```

5. Şimdi, komşular hakkında bilgi toplamak için CDP'yi kullanın. Kullanılabilir komutların listesini, `sh cdp ?` yazarak alabilirsiniz.

```
RouterC#sh cdp ?
   entry     Information for specific neighbor entry
   interface CDP interface status and configuration
   neighbors CDP neighbor entries
   traffic   CDP statistics
   <cr>
```

6. Interface bilgisini ve interfece'in kullandığı varsayılan enkapsülasyonu görmek için `sh cdp int` yazın. Bu, ayrıca CDP timer bilgisini de gösterir.
7. Tüm cihazlardan alınan CDP bilgisini görmek için `sh cdp entry *` yazın.
8. Tüm bağlı komşular hakkında bilgi toplamak için, `show cdp neighbors` yazın.(Bu komutla alınan çıktıdaki özel bilgiyi bilmelisiniz).
9. `Show cdp neighbors detail` yazın. Bunun, `show cdp entry *` ile aynı çıktıyı verdiğine dikkat edin.

## Pratik Lab 5.5: Telnet'i Kullanmak

1. Router'ınıza bağlanın ve `en` veya `enable` yazarak privileged moda girin.
2. RouterA'dan, komut satırından `telnet` ***`ip_address`*** yazarak, uzak router'ınıza telnet yapın.
3. RouterA'nın komut satırından RouterB'nin IP adresini yazın. Router'ın otomatik olarak, sizin yazdığınız IP adresine telnet yapmaya çalıştığına dikkat edin. Telnet komutunu kullanabilir veya sadece IP adresini yazarsınız.
4. RouterA'nın komut istemcisine dönmek için, RouterB'den, Ctrl+Shift+6 ya basın ve sonra X'e basın. Şimdi, üçüncü router'ınız olan RouterC'ye telnet yapın. RouterA'nın komut istemcisine dönmek için, Ctrl+Shift+6 tuş kombinasyonuna basın ve sonra X'e basın.
5. RouterA'dan, `show sessions` yazın. İki oturumunuz olduğuna dikkat edin. Oturumun solunda görünen numaraya basın ve bu oturuma geri dönmek için iki defa Enter'a basın. Asterisk, varsayılan oturumu gösterir. Bu oturuma dönmek için Enter'a iki kez basabilirsiniz.

6. RouterB ile olan oturumunuza gidin. Show `users` yazın. Bu, konsol bağlantısını ve uzak bağlantıyı gösterir. Oturumu temizlemek için `disconnect` komutunu kullanabilir veya RouterB ile oturumunuzu sonlandırmak için komut istemcisinden exit yazabilirsiniz.
7. İlk router'da `show sessions` yazarak ve RouterC'ye geri dönmek için bağlantı numarasını kullanarak RouterC'nin konsol portuna gidin. `Show user` yazın ve ilk router'ınız RouterA'ya bağlandığınıza dikkat edin.
8. Telnet oturumunu sonlandırmak için `clear line` yazın.

## Pratik Lab 5.6: Hostname'leri Çözümlemek

1. Router'ınıza bağlanın ve en veya enable yazarak privileged moda girin.
2. RouterA'dan, komut satırında `todd` yazın ve Enter'a basın. Aldığınız hataya ve gecikmeye dikkat edin. Router, bir DNS sunucusu arayarak, hostname'i bir IP adresine çözümlemeye çalışmaktadır. Bu özelliği, global configuration moddan, `no ip domain-lookup`'ı kullanarak kapatabilirsiniz.
3. Bir host tablosu oluşturmak için `ip host` komutunu kullanın. RouterA'dan, aşağıdaki komutları girerek, RouterB ve RouterC için bir host tablosu girişi ekleyin.

```
ip host routerb ip_address
ip host routerc ip_address
```

İşte bir örnek:

```
ip host routerb 172.16.20.2
ip host routerc 172.16.40.2
```

4. Komut istemcisinden (config istemcisinden değil), `ping routerB` yazarak host tablonuzu test edin.

```
RouterA#ping routerb
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.20.2, timeout
  is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip
  min/avg/max = 4/4/4 ms
```

5. Host tablonuzu, `ping routerc` yazarak test edin.

```
RouterA#ping routerc
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.40.2, timeout
  is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip
  min/avg/max = 4/6/8 ms
```

6. RouterB'ye oturumunuzu açık tutmaya devam edin ve sonra Ctrl+Shift+6 ve devamında X'e basarak, RouterA'ya dönün

7. Komut satırından `routerc` yazarak, RouterC'ye telnet yapın.
8. RouterA'ya dönün ve Ctrl+Shift+6 ve sonra X'e basarak RouterC'ye oturumunuzu koruyun.
9. `Show hosts` yazıp Enter'a basarak, host tablosuna bakın.

```
Default domain is not set
Name/address lookup uses domain service
Name servers are 255.255.255.255
Host                 Flags      Age Type   Address(es)
routerb             (perm, OK)  0   IP    172.16.20.2
routerc             (perm, OK)  0   IP    172.16.40.2
```

# Gözden Geçirme Soruları

NOT *Aşağıdaki sorular, bu bölümün materyallerini anladığınızı test etmek için tasarlanmıştır. İlave sorulara nasıl sahip olacağınızla ilgili daha fazla bilgi için lütfen bu kitabın Giriş bölümüne bakın.*

1. o/r 0x2142 komutu ne sağlar?
   A. Router'ı yeniden başlatmak için kullanılmaktadır.
   B. NVRAM'deki konfigürasyonu göz ardı etmek için kullanılmaktadır.
   C. ROM monitor moda girmek için kullanılmaktadır.
   D. Kayıp şifrelere bakmak için kullanılmaktadır.
2. Hangi komut, IOS'u, ağınızdaki host'a yedekleyecektir?
   A. IOS'u, 172.16.10.1'e transfer edin
   B. copy run start
   C. copy tftp flash
   D. copy start tftp
   E. copy flash tftp
3. Firma ağınızdaki bağlantı probleminde hata tespiti yapıyorsunuz ve problemi izole etmek istiyorsunuz. Erişilemeyen bir ağa giden route'daki bir router'ın hatalı olduğundan şüpheleniyorsunuz. Hangi IOS user exec komutu kullanılmalıdır?
   A. Router>ping
   B. Router>trace
   C. Router>show ip route
   D. Router>show interface
   E. Router>show cdp neighbors
4. Konfigürasyonu, bir network host'undan, router'ın RAM'ine kopyalıyorsunuz. Yapılandırma, doğru görünüyor, fakat çalışmamaktadır. Problem ne olabilir?
   A. RAM'e yanlış konfigürasyon kopyaladınız.
   B. Konfigürasyonu, RAM yerine, flash belleğe kopyaladınız.
   C. Kopya, running-config'teki `shutdown` komutunun üzerine yazılmadı.
   D. `copy` komutu başlatıldıktan sonra, IOS bozuldu.
5. Bir network yöneticisi, mevcut kurulu imajı silmeden, bir router'ın IOS'unu upgrade etmek istiyor. Hangi komut, mevcut IOS imajının kullandığı bellek boyutunu gösterir ve yeni ile mevcut imajları tutmak için yeterli alan olup olmadığını belirtir?
   A. show version
   B. show flash
   C. show memory
   D. show buffers
   E. show running-config
6. Firma ofisi, bağlanmak için yeni bir router gönderir, fakat konsol kablosuyla bağlandığınızda, router'da zaten konfigürasyon olduğunu gördünüz. Router'a yeni konfigürasyonu girmeden önce, ne yapılmalıdır?

A. RAM silinmeli ve router yeniden başlatılmalıdır.

B. Flash silinmeli ve router yeniden başlatılmalıdır.

C. NVRAM silinmeli ve router yeniden başlatılmalıdır.

D. Yeni konfigürasyon girilmeli ve kaydedilmelidir.

7. Hangi komut, Cisco IOS'un yeni versiyonunu bir router'a yükler?

   A. copy flash ftp

   B. copy ftp flash

   C. copy flash tftp

   D. copy tftp flash

8. Hangi komut size, router'ınızda çalışan IOS versiyonunu gösterir?

   A. sh IOS

   B. sh flash

   C. sh version

   D. sh running-config

9. Şifre kurtarma prosedürünü başarıyla tamamladıktan ve router'u normal çalışmasına döndürdükten sonra, hangi configuration register değeri olmalıdır?

   A. 0x2100

   B. 0x2101

   C. 0x2102

   D. 0x2142

10. Bir router'daki konfigürasyonu, `copy running-config startup-config` komutuyla kaydettiniz. Ancak router'da, boş bir konfigürasyon belirmektedir. Problem ne olabilir?

    A. Router'ı, doğru komutla boot etmediniz.

    B. NVRAM bozuldu.

    C. configuration register ayarı yanlıştır.

    D. Yeni upgrade edilen IOS, router'ın donanımı ile uyumlu değildir.

    E. Kaydettiğiniz konfigürasyon, donanımınızla uyumlu değildir.

11. Şayet, aynı zamanda birden fazla Telnet oturumu açmak istiyorsanız, hangi tuş kombinasyonunu kullanmalısınız?

    A. Tab+spacebar

    B. Ctrl+X, sonra 6

    C. Ctrl+Shift+X, sonra 6

    D. Ctrl+Shift+6, sonra X

12. Bir uzak cihaza telnet yapmakta başarısız oldunuz. Problem ne olabilir? (İki şık seçin.)

    A. IP adresi yanlıştır.

    B. Access control list, Telneti engelliyordur.

    C. Arızalı seri kablo vardır.

    D. VTY şifresi kayıptır.

13. Show hosts komutuyla hangi bilgi görüntülenmektedir? (İki şık seçin.)
    A. Geçici DNS girdileri,
    B. hostname komutu kullanarak oluşturulan router isimleri
    C. Router'a erişimine izin verilen workstation'ların IP adresleri
    D. ip host komutu kullanarak oluşturulan kalıcı isim-IP adresi eşleşmeleri.
    E. Bir hostun, Telnet üzerinden router'a bağlanması için geçen zaman.
14. Bir router'daki bağlantı problemini kontrol etmek için kullanılabilecek üç komut nedir? (Üç şık seçin.)
    A. show interfaces
    B. show ip route
    C. tracert
    D. ping
    E. dns lookups
15. Bir router'a telnet'le bağlandınız ve ihtiyacınız olan değişiklikleri yaptınız; şimdi Telnet oturumunuzu sonlandırmak istiyorsunuz. Hangi komutu girmelisiniz?
    A. close
    B. disable
    C. disconnect
    D. exit
16. Bir uzak cihaza telnet yaptınız ve debug ip rip yazdınız, fakat debug komutundan hiçbir çıktı alamadınız. Problem ne olabilir?
    A. İlk olarak, show ip rip komutunu yazmalısınız.
    B. Ağdaki IP adreslemesi yanlıştır.
    C. terminal monitor komutunu girmelisiniz.
    D. Debug çıktısı sadece konsola gönderilmektedir.
17. Hangi komut, configuration register ayarını göstermektedir?
    A. show ip route
    B. show boot version
    C. show version
    D. show flash
18. Hawaii'de bulunan bir uzak router'ın IP adresini almanız gerekmektedir? Adresi bulmak için ne yapabilirsiniz?
    A. Hawaii'ye uçun, switch'e konsol ile bağlanın, sonra bir içecek ile rahatlayın.
    B. Switch'e bağlı router'da, show ip route komutunu çalıştırın.
    C. Switch'e bağlı router'da, show cdp neighbor komutunu çalıştırın.
    D. Switch'e bağlı router'da, show ip arp komutunu çalıştırın.
    E. Switch'e bağlı router'da, show cdp neighbors komutunu çalıştırın.
19. Router'ın Ethernet portuna direkt bağlı laptop'ınız var. Aşağıdakilerden hangisi, copy flash tftp komutunun başarılı olması için gereklidir?

A. TFTP sunucu yazılımı, router'da çalışmalıdır.

B. TFTP sunucu yazılımı, laptop'ınızda çalışmalıdır.

C. Router'ın Ethernet portuna direkt bağlı Ethernet kablosu, bir düz kablo olmalıdır.

D. Laptop, router'ın Ethernet interface'i ile aynı subnette olmalıdır.

E. `copy flash tftp` komutu, laptop'ın IP adresini sağlamalıdır.

F. Dosyanın kopyalanması için router'ın flash belleğinde yeterli alanın olması gerekmektedir.

20. 0x2102 configuration register ayarı, bir router'a hangi işlevi sağlar?

A. Router'a, ROM monitor moda boot etmesini söyler.

B. Şifre kurtarma sağlar.

C. Router'a, boot sıralaması için NVRAM'e bakmasını söyler.

D. Bir TFTP sunucusundan IOS'u boot eder.

E. ROM'da tutulan bir IOS imajını boot eder.

# Gözden Geçirme Sorularının Cevapları

1. B Varsayılan konfigürasyon ayarı, 0x2102'dir. Router'a, IOS'u flash'tan, konfigürasyonu, NVRAM'den yüklemesini söyler. 0x2142, router'a, NVRAM'deki konfigürasyonu göz ardı etmesini söyler. Böylece şifre kurtarma prosesini çalıştırabilirsiniz.
2. E Varsayılan olarak flash bellekte tutulan IOS'u yedek bir host'a kopyalamak için, copy flash tftp komutunu kullanın.
3. B user veya privileged-moddan kullanılabilen `traceroute` (kısaca `trace`) komutu, bir paketin bir ağ topluluğu boyunca izlediği yolu bulmak için kullanılır ve aynı zamanda, bir router'daki arıza nedeniyle paketin nerede durduğunu da gösterecektir.
4. C Konfigürasyon doğru göründüğünden, kopyalama işini muhtemelen yanlış yapmadınız. Ancak, bir network host'undan, router'a kopyalama yaptığınızda interface'ler otomatik olarak kapalıdır ve `no shutdown` komutuyla manuel olarak etkinleştirilmelidir.
5. B `show flash` komutu size, mevcut IOS'un adını, boyutunu ve flash belleğin boyutunu verecektir.
6. C Router'ı yapılandırmaya başlamadan önce, NVRAM'i, `erase startup-config` komutu ile silmeniz ve sonra `reload` komutunu kullanarak yeniden yüklemeniz gerekir.
7. D `copy tftp flash` komutu, router'ınızdaki flash belleğe yeni bir IOS kopyalamanızı sağlar.
8. C En iyi cevap `show version` komutudur. Router'ınızda halihazırda çalışan IOS dosyasını gösterir. `Show flash` komutu, flash belleğin içindekileri gösterir, çalışan IOS dosyasını göstermez.
9. C Tüm Cisco router'lar, 0x2102 varsayılan configuration register değerine sahiptir. Router'a, IOS'u flash bellekten, konfigürasyonu, NVRAM'den yüklemesini söyler.
10. C Şayet bir konfigürasyonu kaydedip router'ı reload ettiyseniz ve setup mod veya boş bir konfigürasyon gelirse, configuration register ayarlarınız yanlıştır.
11. D Birden fazla Telnet oturumunu açık tutmaya devam etmek için Ctrl+Shift+6 ve sonra X tuş kombinasyonunu kullanın.
12. B, D Cevapların en iyileri, Telnet oturumunun bir access control list ile engellenmesi ve uzak cihazda VTY şifresinin ayarlanmamış olmasıdır.
13. A, D `show hosts` komutu, geçici DNS girdilerindeki ve `ip host` komutu kullanarak oluşturulan kalıcı isimden-IP adresi eşleşme bilgilerini sağlar.
14. A, B, D `tracert` komutu, bir Windows komutudur ve bir router'da çalışmayacaktır. Bir router, `traceroute` komutunu kullanır.
15. D Soru, askıdaki oturumlardan bahsetmediğinden, Telnet oturumunun hala açık olduğunu düşünebilirsiniz. Oturumu sonlandırmak için `exit` yazın.
16. C Konsol mesajlarını, Telnet oturumunuz üzerinden görebilmeniz için `terminal monitor` komutunu girmeniz gerekmektedir.
17. C `show version` komutu, size, mevcut configuration register ayarını sağlar.
18. E A cevabı, kesinlikle en iyi cevap olmasına rağmen, E cevabı maalesef çalışacaktır ve patronunuz muhtemelen `show cdp neighbors detail` komutunu kullanmanızı isteyecektir.
19. B, D, E Bir IOS imajını bir router Ethernet portuna direkt bağlı laptop'a yedeklemede önce, TFTP sunucu yazılımının laptopunuzda çalıştığına, Ethernet kablosunun çapraz olduğuna ve laptop'ın router'ın Ethernet portu ile aynı subnette olduğuna emin olun. Bundan sonra, laptop'ınızdan `copy flash tftp` komutunu kullanabilirsiniz.
20. C 0x2102 varsayılan configuration ayarı, router'a, boot sıralaması için NVRAM'e bakmasını söyler.

# Yazılı Lab 5 Cevapları

1. `copy flash tftp`
2. `copy start tftp`
3. `copy start run`
4. `config mem`
5. `show cdp neighbor detail veya show cdp entry *`
6. `show cdp neighbor`
7. Ctrl+Shift+6, sonra X
8. `show sessions`
9. `copy tftp flash`
10. Hem `copy tftp run` hem de `copy start run`

# 6 IP Routing

# 6 IP Routing

- Routing Temelleri
- IP Routing Prosesi
- Network'ümüzde IP Routing'i Yapılandırmak
- Dinamik Routing
- Distance-Vector Routing Protokolleri
- Routing Information Protocol (RIP)
- Interior Gateway Routing Protocol (IGRP)
- Konfigürasyonlarınızın Doğruluğunu Kontrol Etmek
- Özet
- Sınav Gereklilikleri
- Yazılı Lab 6
- Pratik Lab'lar
- Gözden Geçirme Soruları
- Gözden Geçirme Sorularının Cevapları
- Yazılı Lab 5 Cevapları

# IP Routing

Bu bölümde, IP routing işleyişinden bahsedeceğim. Tüm router ve konfigürasyonların IP ile ilgili olmasından dolayı, bu anlaşılması açısından önemli bir konudur. IP routing, router'ları kullanarak paketlerin bir network'ten diğerine taşınması prosesidir. Daha önceki gibi, tabii ki Cisco router'lardan bahsediyorum.

Bu bölümü okumadan önce, routing protokol ile routed protokol arasındaki farkı bilmek zorundasınız. Bir routing protokol, ağ topluluğundaki tüm network'lerin tamamının dinamik olarak bulunması ve tüm router'ların aynı routing tablosunda olmasını sağlamak için router'lar tarafından kullanılır. Aslında bir routing protokolü, ağ topluluğu boyunca bir paketin izleyeceği yolu belirler. Routing protokolüne örnek, RIP, RIPv2, EIGRP ve OSPF'tir.

Router'lar, tüm network'leri öğrenince, kurulu yapı üzerinde kullanıcı verisi (paketleri) göndermek için routed protokolü kullanılabilmektedir. Routed protokole örnek, IP ve IPv6'dır.

Bunun, öğrenmek için çok önemli bir konu olduğunu söyleyemeyeceğim. Muhtemelen çoğunuz şimdiye kadar ne söylediğimi anlamışsınızdır. Bu bölüm, tamamen temel materyallerle ilgilenir. Şayet bu bölümün içerdiği konuları anlamak istiyorsanız, bunları mutlaka bilmeniz gerekir.

Bu bölümde size, Cisco router'larla IP routingi, nasıl yapılandırıp, doğruluğunu kontrol edeceğinizi göstereceğim. Aşağıdakiler işlenecektir:

- Routing temelleri
- IP routing prosesi
- Statik routing
- Default routing
- Dinamik routing

*Bu bölümle ilgili son güncellemeler için www.lammle.com ve/veya www.sybex.com adresine bakınız.*

NOT

"Enhanced IGRP (EIGRP) ve Open Shortest Path First (OSPF)" başlıklı bölüm 7'de, EIGRP ve OSPF ile daha gelişmiş, dinamik routing'e geçeceğim. Fakat öncelikle, bir ağ topluluğu boyunca paketlerin nasıl dolaştığıyla ilgili temel bilgileri öğrenmeniz gerekir. Öyleyse, hadi başlayalım.

## Routing Temelleri

WAN ve LAN'larınızı bir router'a bağlayarak, bir ağ topluluğu oluşturunca, bu ağ topluluğu boyunca bağlanabilmeleri için tüm host'lara, IP adresi gibi mantıksal network adresleri vermeniz gerekir.

Routing terimi, paketi bir cihazdan alıp ağ boyunca, farklı bir network'teki diğer bir cihaza göndermek için kullanılır. Router'lar, host'larla ilgilenmez, onlar sadece network'ler ve her network'e giden en iyi yolla ilgilenir. Hedef host'un mantıksal ağ adresi, paketleri, route edilmiş network üzerinden bir ağa göndermek ve host'un donanım adresi, paketi, bir router'dan, doğru hedef host'una taşımak için kullanılmaktadır.

Şayet ağınızda router yoksa routing yapmayacağınız aşikardır. Router'lar trafiği, ağ topluluğunuzdaki tüm ağlara route eder. Paketleri route edebilmek için bir router, en azından şunları bilmelidir:

- Hedef adresi
- Uzak ağları öğrenebileceği komşu router'ları
- Uzak ağların hepsine mümkün route'ları
- Uzak ağların her birine en iyi route'u
- Routing bilgisini nasıl koruyup, doğrulayacağını

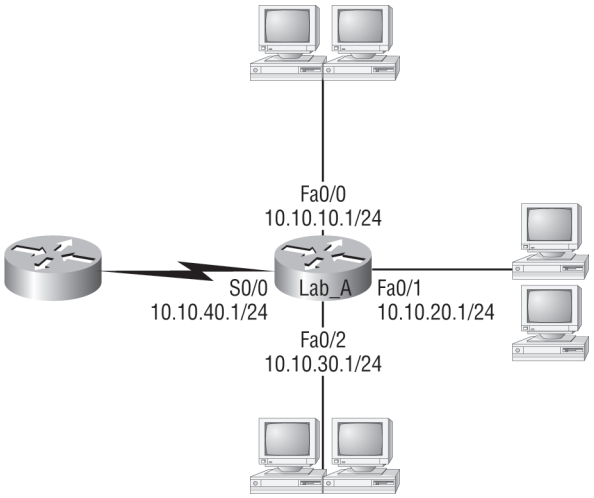
Router, uzak ağları komşu router'lardan ya da yöneticiden öğrenir. Router bundan sonra uzak ağlara nasıl ulaşılacağını belirten bir routing tablosu oluşturur (ağ topluluğunun bir haritası). Şayet network doğrudan bağlıysa router, ona nasıl ulaşacağını zaten bilir.

Şayet network bir router'a doğrudan bağlı değilse, uzak ağa nasıl ulaşacağını öğrenmek için iki yoldan birini kullanmak zorundadır: Bunlar statik routing veya dinamik routing'tir. Statik routing işleminde routing tablosuna tüm network'ler el ile girilir.

Dinamik routing'de bir router'daki protokol, komşu router'larda çalışan aynı protokolle iletişime geçer. Router'lar,daha sonra bildikleri ve roting tablosuna yerleştirdikleri tüm ağlar hakkındaki bilgileri birbirleriyle günceller. Şayet ağda değişiklik olursa dinamik routing protokolü, değişiklikle ilgili tüm router'ları otomatik olarak bilgilendirir. Şayet statik routing kullanılıyorsa yönetici, güncellemelerin tüm router'lara elle girilmesinden sorumludur. Büyük network'lerde, genel olarak dinamik ve statik routing'in ikisi de kullanılır.

IP routing'in işleyişine geçmeden önce bir router'ın, paketleri bir interface'inden route etmek için routing tablosunu nasıl kullandığını gösteren basit bir örneğe göz atalım. Bu bölümde detaylı bir çalışma proses'ine girmiş olacağız.

Şekil 6.1, iki router'dan oluşan basit bir ağı göstermektedir. Lab_A, bir seri ve üç LAN interface'ine sahiptir.




***Şekil 6.1:*** *Basit bir routing örneği.*

Şekil 6.1'e bakarak hangi interface'in bir IP datagram'ını, 10.10.10.10 IP adresli bir host'a iletmek için kullanılacağını görebiliyor musunuz?

`Show ip route` komutunu kullanarak, routing tablosunu görebiliriz. Lab_A, yönlendirme kararlarını vermektedir:

```
Lab_A#sh ip route
[output cut]
Gateway of last resort is not set
C       10.10.10.0/24 is directly connected, FastEthernet0/0
C       10.10.20.0/24 is directly connected, FastEthernet0/1
C       10.10.30.0/24 is directly connected, FastEthernet0/2
C       10.10.40.0/24 is directly connected, Serial 0/0
```

Routing tablosu çıktısındaki C'nin anlamı şudur; listelenen network'ler direk bağlıdır ve ağ topluluğumuzdaki router'lara RIP ve EIGRP gibi bir routing protokolü ekleyene (ya da statik route kullanana) kadar sadece routing tablosundaki direkt bağlı network'lere sahip olacağız.

Şimdi, orijinal soruya geri dönelim: Şekle ve routing tablosu çıktısına bakarak, IP'nin, 10.10.10.10 hedef IP adresine sahip olan bir paketi ne yapacağını söyleyebilir misiniz? Router, paketi FastEthernet interface'ine gönderecek, bu interface, paketi frame'leyecek ve sonra onu network segment'ine gönderecektir.

Başka bir örnek daha yapalım: Aşağıdaki routing tablosu çıktısına göre 10.10.10.14 hedef adresli bir paket, hangi interface'ten iletilecektir?

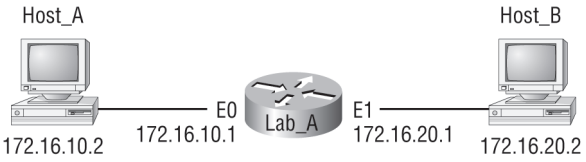
```
Lab_A#sh ip route
[output cut]
Gateway of last resort is not set
C       10.10.10.16/28 is directly connected, FastEthernet0/0
C       10.10.10.8/29 is directly connected, FastEthernet0/1
C       10.10.10.4/30 is directly connected, FastEthernet0/2
C       10.10.10.0/30 is directly connected, Serial 0/0
```

İlk olarak, network'lerin subnet'lendiğini ve her interface'in farklı bir maska sahip olduğunu görebilirsiniz. Subnet'lemeyi bilmiyorsanız, bu soruyu cevaplayamayacağınızı söylemek zorundayım. 10.10.10.14, FastEthernet0/1 interface'ine bağlı 10.10.10.8/29 subnet'indeki bir host olabilir. Anlamadıysanız, paniğe kapılmayın. Sadece geriye gidip bölüm 3'ü tekrar okuyun. Eğer zorlanıyorsanız, bunu yaptıktan sonra çok iyi anladığınızı farkedeceksiniz.

Herkes hazırsa, bu proses'in detaylarına geçelim.

# IP Routing Prosesi

IP routing proses'i, yeterince basittir ve ağınızın boyutuna bağlı olarak değişiklik göstermez. Örnek olarak, Host_A, farklı ağdaki Host_B'yle haberleşmek istediğinde ne olduğunu adım adım açıklamak için Şekil 6.2'yi kullanacağız.




***Şekil 6.2:*** *İki host ve bir router kullanarak bir IP routing örneği.*

Bu örnekte, Host_A'daki bir kullanıcı, Host_B'nin IP adresini ping'lemektedir. Routing bundan daha basit bir işlem yapamaz, fakat yine de birçok adım içerir. Bunları bir inceleyelim:

1. Internet Control Message Protocol (ICMP), bir echo request payload'u oluşturur (bu, data alanının henüz alfabesidir).
2. ICMP, bu payload'u Internet Protokolüne (IP) gönderir, böylece bir paket oluşturulur. Bir paket, en az bir IP kaynak adresi, IP hedef adresi ve 01h olarak Protokol alanı içerir. (Cisco'nun hexadecimal karakterlerin başına *0x* koymaktan hoşlandığını hatırlayın, bu nedenle bu değer 0x01 olarak kullanılır.) Tüm bu bilgiler, hedefe ulaşıldığında alıcı host'a, payload'u kimin ele alacağını söyler (bu örnekte, ICMP).
3. Paket oluşturulunca, IP hedef adresinin lokalde mi yoksa uzak network'te mi olduğunu belirler.
4. IP, bunun uzak bir istek olduğunu belirleyince, paketin uzak ağa route edilebilmesi için varsayılan ağgeçidine gönderilmesi gerekir. Windows'taki registery, yapılandırılmış varsayılan ağgeçidini bulmak için kullanılmaktadır.

5. Host 172.16.10.2'nin (Host_A)'nın varsayılan ağgeçidi, 172.16.10.1 olarak ayarlanmıştır. Bu paketin varsayılan ağgeçidine gönderilmesi için, router'ın Ethernet 0 interface'inin donanım adresinin bilinmesi gerekir. Neden? Paket, Data Link katmanına iletilip frame'lendi ve 172.16.10.0 network'üne bağlı router interface'ine gönderildi. Host'ların sadece lokal LAN'daki donanım adresleri üzerinden haberleşebilmesinden dolayı, Host_A'nın, Host_B ile iletişime geçmesi için paketleri, lokal ağdaki default gateway'in Media Access Control (MAC) adresine göndermesi gerekmektedir.

NOT *MAC adresleri daima, LAN'da lokaldedir ve asla bir router'u geçemezler.*

6. Sonra, host'un Address Resolution Protocol (ARP) önbelleğidefault gateway'in IP adresinin, bir donanım adresine çözümlenip çözümlenmediğini kontrol eder:
   - Şayet yapıldıysa paket, frame'lenmek için Data Link katmanı için hazırdır. (Donanım adresi de paketle birlikte gönderilir.) Host'unuzdaki ARP önbelleğini görmek için aşağıdaki komutu kullanın:

```
C:\>arp -a
Interface: 172.16.10.2 —- 0x3
Internet Address      Physical Address      Type
172.16.10.1           00-15-05-06-31-b0     dynamic
```

   - Şayet donanım adresi, henüz host'un ARP önbelleğinde değilse, lokal ağa 172.16.10.1'in donanım adesini bulmak için bir ARP broadcast'i gönderilir. Router, Ethernet 0'ın donanım adresini cevap olarak gönderir ve host bu adresi cache'e ekler.

7. Paket ve hedef donanım adresi, Data Link katmanına gönderilince LAN sürücüsü, kullanılan LAN tipi üzerinden (bu örnekte, Ethernet) medya erişimi sağlaması için kullanılır. Sonra, kontrol bilgisiyle enkapsüle edilerek, bir frame oluşturulur. Bu frame'in içindekiler, donanım hedef ve kaynak adresleri artı, bu olayda paketi Data Link katmanına gönderen Network katmanı protokolünü (bu örnekte, IP) belirten bir Ether-Type alanıdır. Frame'in sonundaki Frame Check Sequence (FCS) alanı, cyclic redundancy check (CRC) sonucunu tutar. Frame, Şekil 6.3'te detaylandırdığım şekilde görünmektedir. O, Host_A'nın donanım (MAC) adresini ve default gateway'in hedef donanım adresini içermektedir. Uzak host'un MAC adresini içermeyeceğini hatırlayın.


***Şekil 6.3:*** *Host_B ping'lendiğinde, HostA'dan Lab_A'ya kullanılan frame.*

8. Frame tamamlanınca, her seferinde bir bit olacak şekilde fiziksel ortam aracına (bu örnekte, sarmal-çift kablo) konması için Physical katmana gönderilir.
9. Collision domaindeki her cihaz, bu bit'leri alır ve frame oluşturur. Her biri, CRC çalıştırır ve FCS alanındaki cevabı karşılaştırır. Şayet cevaplar eşleşmezse, frame atılır.
   - Şayet CRC eşleşirse, donanım hedef adreslerinin eşleşip eşleşmediği kontrol edilir (bu örnekte, router'ın Ethernet0 interface'idir).
   - Şayet donanım adresi de eşleşirse, Network katmanında kullanılan protokolün bulunması için Ether-Type alanı kontrol edilir.
10. Paket, frame'den çekilir ve frame'in solundakiler atılır. Paket, Ether-Type alanında listelenen (IP adresi olarak verilen) protokole gönderilir.
11. IP paketi alır ve IP hedef adresini kontrol eder. Paketin hedef adresi, alan router'ın kendisinde yapılandırılmış bir adresle eşleşmediğinden router, kendi routing tablosundaki hedef IP network adresine bakar.

12. Routing tablosu, 172.16.20.0 network'ü için bir kayıta sahip olmalıdır, yoksa paket, hemen atılacaktır ve network erişilemez şeklinde bir ICMP mesajı, ping'i göndermeye çalışan cihaza gönderilecektir.
13. Şayet router, tablosunda hedef network için bir kayıt bulamazsa, paket çıkış interface'ine (burada Ethernet1 interface'idir) gönderilir. Aşağıdak çıktı, Lab_A router'ının routing tablosunu gösterir. C, direkt bağlı anlamındadır. Tüm network'ler, direkt bağlı olduğundan, bu ağda bir routing protokolüne ihtiyaç yoktur.

```
Lab_A>sh ip route
Codes:C - connected,S - static,I - IGRP,R - RIP,M - mobile,B –
   BGP, D - EIGRP,EX - EIGRP external,O - OSPF,IA - OSPF inter
   area, N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external
   type 2, E1 - OSPF external type 1, E2 - OSPF external type 2,
   E – EGP,i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia
   - IS-IS intearea * - candidate default, U - per-user static
   route, o – ODR P - periodic downloaded static route

Gateway of last resort is not set
    172.16.0.0/24 is subnetted, 2 subnets
C   172.16.10.0 is directly connected, Ethernet0
C   172.16.20.0 is directly connected, Ethernet1
```

14. Router paketi, Ethernet 1'in arabelleğine gönderir.
15. Ethernet 1 arabelleği, hedef host'un donanım adresini bilmek zorundadır ve ilk olarak ARP belleğini kontrol eder.
    - Host_B'nin donanım adresi, zaten çözümlendiyse ve router'ın ARP önbelleğinde ise, o zaman paket ve donanım adresi, frame'lenmesi için Data Link katmanına gönderilir. Şimdi, `show ip arp` komutunu kullanarak, Lab_A router'ının ARP önbelleğine bir bakalım:

```
Lab_A#sh ip arp
   Protocol  Address      Age(min) Hardware Addr  Type   Interface
   Internet  172.16.20.1    -     00d0.58ad.05f4  ARPA   Ethernet0
   Internet  172.16.20.2    3     0030.9492.a5dd  ARPA   Ethernet0
   Internet  172.16.10.1    -     00d0.58ad.06aa  ARPA   Ethernet0
   Internet  172.16.10.2   12     0030.9492.a4ac  ARPA   Ethernet0
```

      Tire (-), bunun, router'daki fiziksel bir bellek olduğunu belirtir. Yukardaki çıktıdan router'ın, 172.16.10.2'yi (Host B) ve 172.16.20.2'nin (Host B) donanım adresini bildiğini görebiliriz. Cisco router'lar, ARP tablosundaki bir kaydı 4 saat saklar.

    - Donanım adresi henüz çözümlenmediyse, router, 172.16.20.2'nin donanım adresini bulmak için E1'den bir ARP isteği gönderir. Host_B, kendi donanım adresi ile cevap verir ve paket ile hedef donanım adresi, framlenmek için, Data Link katmanına gönderilir.
16. Data Link katmanı, hedef ve kaynak donanım adresi, Ether-Type alanı ve sonunda FCS olan bir frame oluşturur. Frame, her seferinde bir bit olarak fiziksel ortam aracına gönderilmesi için, Physical katmana gönderilir.
17. Host_B, frame'i alır ve hemen bir CRC çalıştırır. Şayet netice, FCS alanındaki ile eşleşirse, sonra donanım hedef adresi kontrol edilir. Eğer, host bir eşleşme bulursa, paketi Network katmanında ele alması gereken protokolü belirlemek için Ether-Type alanı kontrol edilir.

18. Network katmanında IP, paketleri alır ve IP hedef adresini kontrol eder. Son olarak bir eşleştirme olduğundan protokol alanı payload'un kime verilmesi gerektiğini kontrol eder.
19. Payload, ICMP'ye gönderilir. O, bunun bir echo request olduğunu anlar. ICMP, paketi hemen atarak ve echo reply olarak yeni bir payload üreterek cevap verir.
20. Sonra, kaynak ve hedef adresleri, protokol alanı ve payload içeren bir paket oluşturulur. Hedef cihaz şimdi Host_A'dır.
21. IP, hedef IP adresinin lokaldeki veya uzak ağdaki bir cihazın olup olmadığını anlamak için kontrol eder. Hedef cihaz, uzak ağda olduğunda, paketin varsayılan ağgeçidine gönderilmesi gerekir.
22. Varsayılan ağgeçidinin IP adresi, Windows cihazının Registery'sinde bulunur ve ARP önbelleği, donanım adresinin, IP adresinden çözümlenip çözümlenmediğini kontrol eder.
23. Varsayılan ağgeçidinin IP adresi bulununca, paket ve hedef donanım adresi, frame'lenmek için Data Link katmanına gönderilir.
24. Data Link katmanı, bilgi paketlerini frame'ler ve başlığında şunları içerir:
    - Hedef ve kaynak donanım adresleri.
    - 0x0800 (IP) ile Ether-Type alanı.
    - Beraberinde CRC cevabı ile FCS alanı.
25. Frame, şimdi, her seferinde bir bit olarak ortam aracı üzerinden aktarılmak için Physical katmana gönderilir.
26. Router'ın Ethernet1 interface'i, bit'leri alır ve bir frame oluşturur. CRC çalışır ve FCS alanı, cevabın eşleştiğinden emin olmak için kontrol edilir.
27. CRC'nin tamam olduğu anlaşılınca, donanım hedef adresi, kontrol edilir. Router'ın hedef adresi eşleştiğinden, paket, frame'den çıkarılır ve Ether-Type alanı, paketin taşınması gereken, Network katmanı protokolünün anlaşılması için kontrol edilir.
28. Protokolün, IP olduğu tespit edilir, bu yüzden paketi o alacaktır. IP, önce IP başlığında CRC kontrolü yapar ve sonra hedef IP adresini kontrol eder.

    Hedef IP adresi, router'ın interface'i ile eşleşmediğinden, routing tablosu, 172.16.10.0'a bir route'a sahip olup olmadığını kontrol eder. Şayet, hedef network için bir route'a sahip değilse, paket hemen atılır. (Bu, yöneticilerin çoğunun kafasının karıştığı noktadır. Şayet ping başarısızsa, birçok kişi, paketin, hedef host'una asla ulaşamayacağını düşünür. Fakat burada gördüğümüz gibi, bu herzaman olan bir durum değildir.. Paket, dönüş yolunda atılır, host'a giden yolda değil.).

NOT *IP, Data Link katmanının yaptığı gibi komple bir CRC çalıştırmaz. Sadece hatalar için başlığı kontrol eder.*

NOT *Anlaşılması için ufak bir not: paket, orijinal host'a dönişünde kaybolursa, genelde bilinmeyen bir hata olduğundan, "request timed-out" mesajı alırsınız. Şayet, bir router'ın, hedef cihaza giden yolda routing tablosunda olmaması gibi bilinen bir sebepten hata olursa, bir "destination unreachable" mesajı görürsünüz. Bu, problemin hedefe giden yolda mı, dönüş yolunda mı olduğunu anlamanıza yardımcı olur.*

29. Bu olayda router, 172.16.10.0 network'üne nasıl gideceğini bilemez. Çıkış interface'i, E0'dır, bu nedenle paketi, E0'a gönderir.
30. Router, 172.16.10.2, için donanım adresinin çözümlenip çözümlenmediğini anlamak için ARP önbelleğini kontrol eder.
31. 172.16.10.2 nin donanım adresi, Host_B'ye dönüşte önbelleğe atıldığından, donanım adresi ve paket, Data Link katmanına gönderilir.
32. Data Link katmanı, hedef ve kaynak donanım adresleri ile bir frame oluşturur ve sonra Ether-Type alanına IP koyar. Frame'de bir CRC çalışır ve cevap, FCS alanına yerleştirilir.

33. Frame, sonra, her seferinde bir bit olarak gönderilmesi için Physical katmana gönderilir.
34. Hedef host frame'i alır, bir CRC çalıştırır, hedef donanım adresini kontrol eder ve paketi kime yollayacağını anlamak için Ether-Type alanına bakar.
35. IP, belirtilen alıcıdır ve paket, Network katmanındaki IP'ye gönderildikten sonra, IP, payload'u ICMP'ye vermek için yönergeler bulur ve ICMP, paketin, bir ICMP echo reply olduğunu belirler.
36. ICMP, kullanıcı interaface'ine bir ünlem işareti göndererek cevap alındığını onaylar. Bundan sonra, ICMP, hedef host'a dört echo request'i daha gönderir.

IP routing'i anlamak için Todd'un 36 basit adımını gördünüz. Burada anlaşılması gereken en önemli nokta, büyük bir ağa sahip olsanız da prosesin aynı olacağıdır. Gerçekten büyük bir ağ topluluğunda, hedef host'unu bulmadan önce, paket, daha çok hop'u geçecektir.

Host_A, Host_B'ye bir paket gönderdiğinde, hedef donanım adresinin, default gateway'in Ethernet interface'ini kullandığını bilmek çok önemlidir. Neden? Çünkü Frame'ler, hedef ağlarda bulunamazlar, sadece lokal ağda olur. Bu yüzden, uzak ağlara hedeflenen paketler, default gateway'e gitmek zorundadır.

Şimdi, Host_A'nın ARP önbelleğine bir bakalım:

```
C:\ >arp -a
Interface: 172.16.10.2 —- 0x3
  Internet Address      Physical Address      Type
  172.16.10.1           00-15-05-06-31-b0     dynamic
  172.16.20.1           00-15-05-06-31-b0     dynamic
```

Host_A'nın, Host_B'ye gitmek için kullandığı donanım adresinin (MAC), Lab_A'nın E0 interface'i olduğunu farkettiniz mi? Donanım adresleri daima lokaldir ve onlar, bir router'ın interface'ini geçmez.

Bu prosesi anlamak çok önemlidir, bu yüzden, bunu hafızanıza kazıyın.

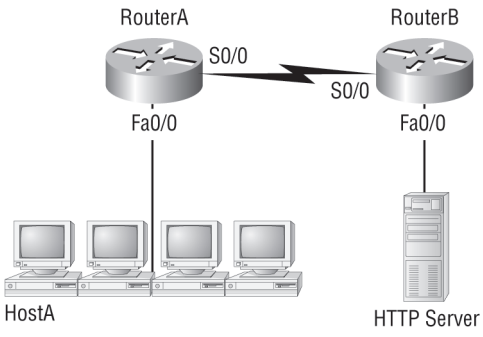
## IP Routing'i Anladığınızı Kontrol Etmek

Çok önemli olduğundan, IP routing'i anladığınızdan emin olmak istiyorum. Bu bölümü, birkaç basit şekile bakarak ve bazı basit IP routing sorularına cevap vererek, IP routing prosesinin anladığınızı test etmek için kullanacağım.

Şekil 6.4, RouterA'ya bağlı bir LAN'ı gösterir. RouterA, diğer taraftan, bir WAN linki üzerinden, RouterB'ye bağlıdır. RouterB, bir HTTP sunucusu barındıran bir LAN'a bağlıdır.

Bu şekilden çıkarmanız gereken önemli bilgi, bu örnekte, IP routing'in tam olarak nasıl olacağıdır. Biraz hile yapacağız. Size cevabı vereceğim, fakat sonra, şekle dönüp, cevaplarıma bakmadan örnek2'nin cevaplarını verip veremeyeceğinizi anlayacaksınız.

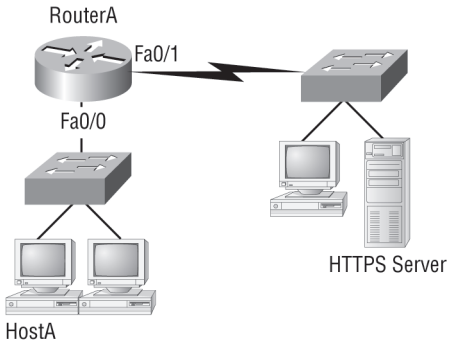
1. HostA'dan, bir frame'in hedef adresi, RouterA'nın F0/0 interface'inin MAC adresidir.
2. Bir paketin hedef adresi, HTTP sunucusunun network interface card'ının (NIC) IP adresidir.
3. Segment başlığındaki hedef port numarası, 80'dir.




***Şekil 6.4:*** *IP routing örneği 1.*

Bu örnek oldukça basitti ve konuyla tam ilgiliydi. Şayet, HTTP'yi kullanarak çok sayıda host sunucuyla haberleşiyorsa, hepsinin farklı bir kaynak port numarasını kullandığı hatırlanması gereken bir noktadır. Bu, sunucunun Transport katmanında ayrılmış veriyi korumasının sebebidir.

Biraz karıştıralım ve network'e, başka bir ağ topluluğu cihazı ekleyelim, sonrada soruları cevaplayıp cevaplayamayacağımıza bakalım. Şekil 6.5'de, bir router ve iki switchten oluşan bir network görünmektedir.

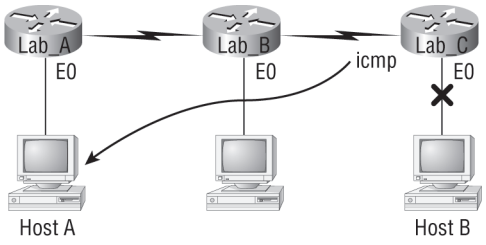



***Şekil 6.5:*** *IP routing örneği 2.*

Burada, HostA, HTTPS sunucusuna bir veri gönderdiğinde, IP routing prosesinde neler olduğunu görebilirsiniz:

1. HostA'dan, bir frame'in hedef adresi, RouterA'nın F0/0 interface'inin MAC adresi olacaktır.
2. Bir paketin hedef adresi, HTTPS sunucusunun network interface card'ının (NIC) IP adresi olacaktır.
3. Segment'teki hedef port numarası, 443 olacaktır.

Switchlerin, default gateway ya da başka hedef adresi olarak kullanılmadığına dikkat edin. Çünkü switch'lerin routig'le bir ilgisi yoktur. HostA için, default gateway olarak switch'i kaçınızın seçtiğini merak ediyorum. Şayet böyle yaptıysanız, kendinizi kötü hissetmeyin, sadece neden böyle düşündüğünüzü gözden geçirin. Hedef MAC adresinin daima, router'ın interface'i olacağını hatırlayın. (son iki örnekte olduğu gibi, şayet paketler LAN'ın dışında bir hedefe gönderiliyorsa).




***Şekil 6.6:*** *ICMP hata örneği.*

IP routing'in daha gelişmiş konularına geçmeden önce, ICMP ve onun bir ağ topluluğunda nasıl kullanıldığına detaylı bakalım. Şekil 6.6'da gösterilen network'e bir bakalım. Şayet Lab_C'nin LAN interface'i giderse, ne olacağını kendinize sorun.

Lab_C, HostB'nın erişilemez olduğu bilgisini HostA'ya göndermek için ICMP kullanır ve bunu bir ICMP destination unreachable mesajı yollayarak yapacaktır. Birçok insan, bu mesajı, Lab_A'nın gönderdiğini düşünür. Bu yanlıştır, çünkü mesajı interface'i arızalanan router gönderir.

Gelin başka bir probleme bakalım: bir firma router'ının routing tablosuna bakalım:

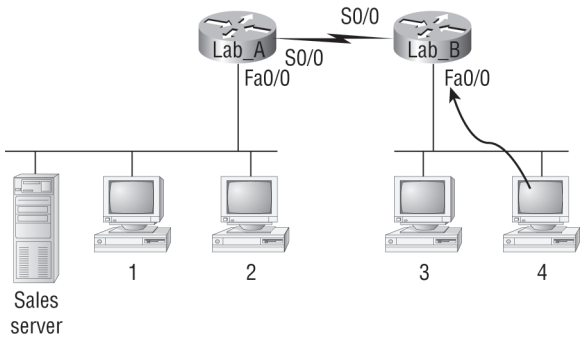
```
Corp#sh ip route
[output cut]
```

```
R    192.168.215.0 [120/2] via 192.168.20.2, 00:00:23, Serial0/0
R    192.168.115.0 [120/1] via 192.168.20.2, 00:00:23, Serial0/0
R    192.168.30.0 [120/1] via 192.168.20.2, 00:00:23, Serial0/0
C    192.168.20.0 is directly connected, Serial0/0
C    192.168.214.0 is directly connected, FastEthernet0/0
```

Burada ne görüyoruz? Şayet firma router'ının, 192.168.214.20 kaynak IP adresiyle bir IP paketi aldığını söylersem, Corp router'ının bu paketi ne yapacağını düşünürsünüz?

Şayet, "paket, FastEthernet0/0 interface'inden geldi, fakat routing tablosu, 192.168.22.0 network'üne (ya da bir default route) bir route göstermediğinden, router paketi atacak ve FastEthernet0/0 interface'ine, bir ICMP "destination unreacable mesajı gönderir" derseniz, siz bir dahisiniz! Bunu yapmasının sebebi, bunun, paketin geldiği kaynak LAN olmasıdır.

Şimdi, diğer bir şekle bakalım ve frame ile paketlerden detaylı olarak bahsedelim. Aslında, yeni birşeylerden bahsetmiyorum. Sadece temel IP routing'ini tamamıyla anladığınızdan emin olmaya çalışıyorum. Çünkü bu kitap ve hedeflenen sınav konularının hepsi IP routing'le ilgilidir. Şekil 6.7'yi, aşağıdaki sorular için kullanacağız.




***Şekil 6.7:*** *MAC ve IP adreslerini kullanarak, basit IP routing'i.*

Şekil 6.7'yi referans alarak, cevaplarına ihtiyacınız olan soruların bir listesini bulabilirsiniz:

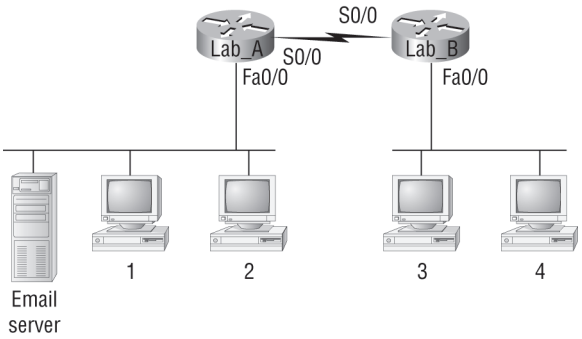
1. Satış sunucusuyla iletişime başlamak için Host4, bir ARP request gönderir. Topolojide görünen cihazlar, bu isteğe nasıl cevap vereceklerdir?
2. Host 4, bir ARP reply aldı. Host 4, şimdi, bir paket oluşturacak ve sonra bu paketi frame'e yerleştirecektir. Host4, Satış sunucusuna bağlanacaksa, Host4'ün bırakacağı hangi bilgi, paket başlığında yer alacaktır?
3. Sonunda, Lab_A router'ı, paketi alır ve onu, Fa0/0'dan, sunucuya doğru LAN'a yollar. Frame, başlıkta hangi hedef ve kaynak adreslerine sahip olacaktır?
4. Host 4, Satış sunucusunda, aynı anda iki browser penceresinde iki web dökümanı görüntülüyor. Data, doğru browser penceresine giden yolu nasıl bulur?

Muhtemelen şimdi yazacaklarımı, zor okunur bir yazı tipinde yazıp kitabın başka bir bölümünde baş aşağıya yerleştirmeliyim ki, sizin için kopya çekmesi ve gizlice göz atması zor olsun. Fakat gizlice göz attığınız takdirde siz kaybedeceğiniz için işte cevaplarınız:

1. Sunucuyla iletişime başlamak için, Host4, bir ARP request gönderir. Topolojide gösterilen cihazlar, bu isteğe nasıl cevap verecektir? MAC adresinin, lokalde kalması gerektiğinden, Lab_B router'ı, Fa0/0 interface'i ile cevap verecektir ve paketler Satış sunucusuna gönderileceği zaman, Host4, tüm frame'leri, Lab_B Fa0/0 interface'inin MAC adresine gönderecektir.
2. Host 4, bir ARP reply aldı. Host 4, şimdi, bir paket oluşturacak ve sonra bu paketi frame'e yerleştirecektir. Host 4, Satış sunucusuna bağlanacaksa, Host4'ün bırakacağı hangi bilgi, paket

başlığında yer alacaktır? Şimdi, paketler sözkonusu olduğundan, kaynak adresi, Host 4'ün IP adresi olacaktır ve hedef IP adresi, Satış sunucusunun IP adresi olacaktır.

3. Sonunda, Lab_A router'ı, paketi alır ve onu, Fa0/0'dan, sunucuya doğru LAN'a yollar. Frame, başlıkta hangi hedef ve kaynak adreslerine sahip olur? Kaynak MAC adresi, Lab_A router'ının Fa0/0 interface'i ve hedef MAC adresi, Satış sunucusunun MAC adresi olacaktır. (Tüm MAC adresleri, LAN'daki lokal adresler olmalıdır.)
4. Host 4, Satış sunucusunda, aynı anda iki browser penceresinde iki web dökümanı görüntülüyor. Data, doğru browser penceresine giden yolu nasıl bulur? TCP port numaraları, datanın doğru uygulama penceresine yönlendirilmesi için kullanılmaktadır.Size, gerçek bir ağda routing konfigürasyonu yapmadan önce, birkaç sorum daha olacak. Hazırmısınız? Şekil 6.8, basit bir network'ü göstermektedir ve Host4'ün, e-mail alması gerekmektedir. O, Host 4'ten ayrıldığında hedef adres alanında hangi adresin olması gerekir?




***Şekil 6.8:*** *Temel routing bilgisini test etmek.*

Cevap, Host 4'ün Lab_B router'ının, Fa0/0 interface'ini hedef MAC adresi olarak kullanacağıdır. Bildiğinize eminim. Doğru mu? Tekrar Şekil 6.8'e bakalım: Host 4'ün, Host 1 ile bağlantıya geçmesi gerekmektedir. Host 1'e ulaştığında paket başlığında, hangi OSI katman 3 adresi yer almalıdır?

Bunu bildiğinizi umuyorum: Katman 3'te, kaynak IP adresi, Host 4 olacaktır ve paketteki hedef adresi, Host1'in IP adresi olacaktır. Tabiiki, Host 4'ten hedef MAC adresi, daima, Lab_B router'ının Fa0/0 adresi olacaktır, değil mi? Birden fazla router'a sahip olduğumuzdan, onların birbiriyle haberleşebilmesi için bir routing protokolüne ihtiyacımız vardır. Böylece, trafik, Host 1'in bağlı olduğu ağa ulaşmak için doğru yöne gönderilebilir.

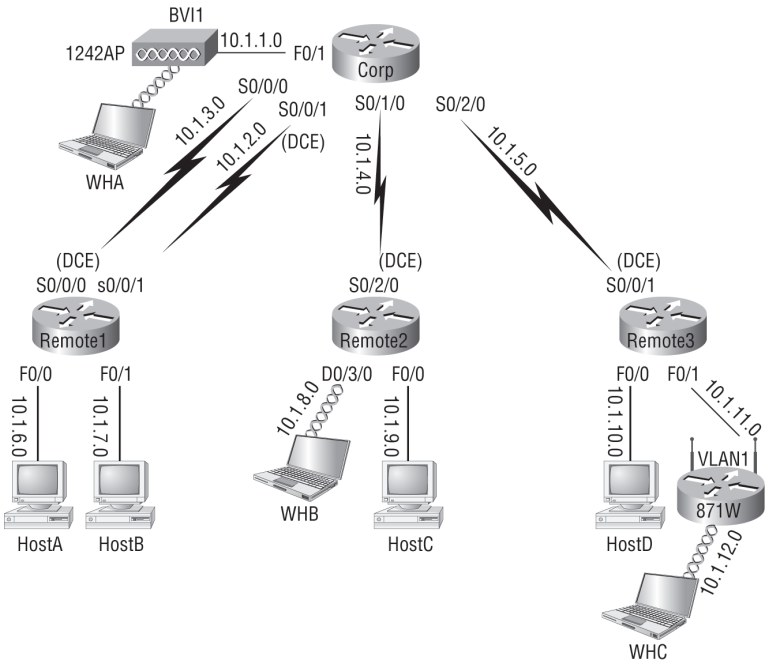
Bir soru daha çözelim, artık IP routing dehası olma yolundasınız. Yine Şekil 6.8'i kullanalım. Host 4, Lab_A router'ına bağlı bir email sunucusuna bir dosya transfer ediyor. Host 4'ten ayrıldığında, katman 2 hedef adresi ne olur? Evet, bu soruyu daha önce sordum. Fakat şunu sormadım: Frame, email sunucusu tarafından alındığında, kaynak MAC adresi ne olacaktır?

Host4'ten ayrılan katman2 hedef adresinin, Fa0/0 interface'inin MAC adresi ve email sunucusunun kaynak katman adresinin, Lab_A router'ının Fa0/0 interface'i olduğunu bildiğinizi umuyorum.

Şayet doğru cevapladıysanız, IP routing'in, geniş bir ağda nasıl ele alındığı konusunu kolaylaştırdınız demektir.

## IP Routing Konfigürasyonu

Ciddi olma ve gerçek bir network'ü yapılandırma zamanı geldi. Şekil 6.9, beş router'ı göstermektedir: Corp, Remote1, Remote2, Remote3 ve 871W (bir wireless router). Varsayılan olarak, router'ların, sadece kendilerine direk bağlı network'leri bildiklerini hatırlayın. Ayrıca, şekilde gösterilen 1242'nin, bir access point olduğunu, 871W gibi wireless bir router olmadığını aklınızda tutun. Access point, bir router'dan ziyade, hub gibidir.




***Şekil 6.9:*** *IP Routing konfigürasyonu.*

Tahmin edebileceğiniz gibi çalışmak için oldukça güzel bir router kolleksiyonu hazırladım. Corp router, Wireless Controller modülü olan bir 2811'dir. (bölüm 12'de göreceksiniz). Remote1 ve 3, 1841 ISR router'dır ve Remote2, wireless WIC kartı ve switch modülü olan bir 2801'dir. Bunları kısaca, R1, R2 ve R3 uzak router grubu olarak belirteceğim. (Bu kitapta, eski router'larda kullandığım birçok komutu çalıştırabilirsiniz, fakat SDM kullanmak için en azından, yeni bir 800 ya da 1800 serisine ihtiyacınız var.)

Bu proje için ilk adım, her interface'ine bir IP adresi vererek doğru bir şekilde yapılandırmaktır. Tablo 6.1, network'ü yapılandırmak için kullanacağım IP adres şemasını göstermektedir. Network'ün nasıl yapılandırıldığını tamamladıktan sonra, IP routing'in nasıl yapılandırılacağını göstereceğim. Aşağıdaki tablodaki her network, 24-bit'lik bir subnet maskına (255.255.255.0) sahiptir. Bu, dikkate değer (subnet) oktet'in, üçüncü olduğunu belirtir.

**Tablo 6.1:** IP Network'ü için Network Adreslemesi

| Router | Network Adresi | Interface | Adres |
|---|---|---|---|
| **CORP** | | | |
| Corp | 10.1.1.0 | F0/1 | 10.1.1.1 |
| Corp | 10.1.2.0 | S0/0/0 | 10.1.2.1 |
| Corp | 10.1.3.0 | S0/0/1(DCE) | 10.1.3.1 |
| Corp | 10.1.4.0 | s0/1/0 | 10.1.4.1 |
| Corp | 10.1.5.0 | s0/2/0 | 10.1.5.1 |
| **R1** | | | |
| R1 | 10.1.2.0 | S0/0/0 (DCE) | 10.1.2.2 |
| R1 | 10.1.3.0 | S0/0/1 | 10.1.3.2 |
| R1 | 10.1.6.0 | F0/0 | 10.1.6.1 |
| R1 | 10.1.7.0 | F0/1 | 10.1.7.1 |
| **R2** | | | |
| R2 | 10.1.4.0 | S0/2/0 (DCE) | 10.1.4.2 |
| R2 | 10.1.8.0 | D0/3/0 | 10.1.8.1 |
| R2 | 10.1.9.0 | F0/0 | 10.1.9.1 |
| **R3** | | | |

**Tablo 6.1:** IP Network'ü için Network Adreslemesi (devam)

| Router | Network Adresi | Interface | Adres |
|---|---|---|---|
| R3 | 10.1.5.0 | S0/0/0/ (DCE) | 10.1.5.2 |
| R3 | 10.1.10.0 | F0/0 | 10.1.10.1 |
| R3 | 10.1.11.0 | F0/1 | 10.1.11.1 |
| **871W** | | | |
| 871W | 10.1.11.0 | Vlan 1 | 10.1.11.2 |
| 871W | 10.1.12.0 | Dot11radio0 | 10.1.12.1 |
| **1242 AP** | | | |
| 1242 AP | 10.1.1.0 | BVI 1 | 10.1.1.2 |

Sadece interface'lere IP adreslerini eklemek ve sonra aynı interface'lerde `no shutdown` komutunu çalıştırmak yeterli olduğundan, router konfigürasyonu, aslında oldukça basit bir prosestir. Devamında biraz daha karmaşık olacaktır, fakat şimdi network'teki IP adreslerini yapılandıralım.

## Corp Konfigürasyonu

Corp router'ını yapılandırmak için beş interface'i yapılandırmamız gerekir. Her router'ın hostname'lerini yapılandırmak, router'ları ayırdetmemizi kolaylaştıracaktır. Bunu yaparken, neden interface açıklamalarını, banner'ları ve router şifrelerini de ayarlamayalım? Her router'da bu komutları kullanmayı alışkanlık haline getirmek oldukça iyi bir fikirdir.

Başlamak için, router'da, `erase startup-config` komutunu kullandım ve reload ettim. Böylece setup moda geçtim. Setup moda girmemek için "no" dedim ve bu bizi, doğruca konsolun kullanıcı adı istemcisine götürecektir. R3 dışında tüm router'larımı bu yolla yapılandıracağım. R3'ü sırf değişiklik olsun diye SDM kullanarak yapılandıracağım. Sizde router'larınızı aynı şekilde yapılandırabilirsiniz.

Yaptıklarımın hepsi aşağıdadır:

```
—- System Configuration Dialog —-

Would you like to enter the initial configuration dialog? [yes/
no]: n
[output cut]
Press RETURN to get started!
Router>en
Router#config t
Router(config)#hostname Corp
Corp(config)#enable secret todd
Corp(config)#interface fastEthernet 0/1
Corp(config-if)#ip address 10.1.1.1 255.255.255.0
Corp(config-if)#description Connection to 1242 AP
Corp(config-if)#no shutdown
Corp(config-if)#int s0/0/0
Corp(config-if)#ip address 10.1.2.1 255.255.255.0
Corp(config-if)#description 1st Connection to R1
Corp(config-if)#no shut
Corp(config-if)#int s0/0/1
Corp(config-if)#ip address 10.1.3.1 255.255.255.0
Corp(config-if)#description 2nd Connection to R1
```

```
Corp(config-if)#no shut
Corp(config-if)#int s0/1/0
Corp(config-if)#ip address 10.1.4.1 255.255.255.0
Corp(config-if)#description Connection to R2
Corp(config-if)#no shut
Corp(config-if)#int s0/2/0
Corp(config-if)#ip address 10.1.5.1 255.255.255.0
Corp(config-if)#description Connection to R3
Corp(config-if)#no shut
Corp(config-if)#line con 0
Corp(config-line)#password console
Corp(config-line)#login
Corp(config-line)#logging synchronous
Corp(config-line)#exec-timeout 0 0
Corp(config-line)#line aux 0
Corp(config-line)#password aux
Corp(config-line)#login
Corp(config-line)#exit
Corp(config)#line vty 0 ?
  <1-1180>  Last Line number
  <cr>
Corp(config)#line vty 0 1180
Corp(config-line)#password telnet
Corp(config-line)#login
Corp(config-line)#exit
Corp(config)#no ip domain-lookup
Corp(config)#banner motd # This is my Corp 2811 ISR Router #
Corp(config-if)#^Z
Corp#copy running-config startup-config
Destination filename [startup-config]?[enter]
Building configuration...
[OK]
Corp#
```

Bir Cisco router'da oluşturulan routing tablosuna bakmak için `show ip route` komutunu kullanın. Komut çıktısı aşağıdaki gibidir:

*Bu konfigürasyon prosesini anlamakta güçlük çekiyorsanız "Cisco Internetworking Operating System (IOS) ve Security Device Manager (SDM)." Başlıklı bölüm 4'e bakın.*

NOT

```
Corp#sh ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
   D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
   N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
   E1 - OSPF external type 1, E2 - OSPF external type 2
   i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS
   level-2, ia - IS-IS inter area, * - candidate default, U -
per-user
```

```
      static route, o - ODR, P - periodic downloaded static route
Gateway of last resort is not set

      10.0.0.0/24 is subnetted, 1 subnets
C        10.1.1.0 is directly connected, FastEthernet0/1
Corp#
```

Sadece yapılandırılmış, direk bağlı network'lerin görüneceğini hatırlayın. Bundan dolayı mı, routing tablosunda sadece FastEthernet0/1 interface'ini görebiliyorum? Endişelenmeyin, diğer uçtaki seri interface'ler çalışır duruma gelene kadar, seri interface'leri göremeyeceksiniz. R1, R2 ve R3 router'larımızı yapılandırır yapılandırmaz, bu interface'ler hemen orada olacaklardır.

NOT *Kısaltma amacıyla, bu modülün kalanında, komutlar kısaltılacaktır.*

Routing tablosu çıktısının sol tarafındaki ***C***'yi fark ettiniz mi? Bunu orada gördüğünüzde, network'ün direk bağlı olduğu anlamına gelir. Her bağlantı tipinin kodu, kısaltmalarıyla beraber `show ip route` komutunun üst tarafında listelenmektedir.

Corp serial0/0/1 interface'i, bir DCE bağlantısıdır. Yani, interface'e `clock rate` komutunu eklememiz gerekmektedir. Gerçek network'lerde, `clock rate` komutunu kullanmaya ihtiyacınız olmayacağını hatırlayın. Bu doğru olduğu halde, onu nasıl/ne zaman kullanacağınızı bilmek ve CCNA sınavınıza hazırlandığınızda, iyice anlamanız hala önemlidir.

Saat denetimimizi, `show controllers` komutu ile görebiliriz:

```
Corp#sh controllers s0/0/1
Interface Serial0/0/1
Hardware is GT96K
DCE V.35, clock rate 2000000
```

Remote router'ların konfigürasyonuna geçmeden önce son bir şey: Corp router'ının s0/0/1 interface'inde, clock rate'in 2000000 olduğunu farkettiniz mi? Şayet, Corp router'ını yapılandırırken, clock rate'i ayarlamadığınız aklınıza gelecektir. Bunun sebebi, ISR router'ların, DCE kablo tipini otomatik algılaması ve clock rate'i otomatik yapılandırmasıdır. Gerçekten güzel bir özellik!

## R1 Konfigürasyonu

Şimdi R1 router'ımızı yapılandırmaya hazırız. Bunu doğru bir şekilde yapmak için, ilgilenmemiz gereken dört interface'imiz olduğunu unutmayın: serial0/0/0, serial0/0/1, FastEthernet0/0 ve FastEthernet0/1. Hostname'i, banner'ı, şifreleri ve interface açıklamalarını router konfigürasyonuna eklemeyi unutmayalım. Corp router'ında yaptığım gibi, üzerindeki konfigürasyonu sildim ve reload ettim.

Kullandığım konfigürasyon şöyledir:

```
R1#erase start
% Incomplete command.
R1#erase startup-config
Erasing the nvram filesystem will remove all configuration files!
   Continue? [confirm][enter]
[OK]
Erase of nvram: complete
R1#reload
Proceed with reload? [confirm][enter]
[output cut]
```

```
%Error opening tftp://255.255.255.255/network-confg (Timed out)
%Error opening tftp://255.255.255.255/cisconet.cfg (Timed out)

         –- System Configuration Dialog –-

Would you like to enter the initial configuration dialog? [yes/
no]: n
```

Devam etmeden önce, yukardaki çıktıyla ilgili konuşmak istiyorum. İlk olarak, yeni ISR router'ların, artık `erase start` komutunu almayacaklarına dikkat edin. Router `erase`'den sonra, s ile başlayan, sadece bir komuta sahiptir, aşağıda göründüğü gibi:

```
Router#erase s?
startup-config
```

IOS'un komutu kabul etmeye devam edeceğini düşündüğünüzü biliyorum, ama değil. İkinci işaret etmek istediğim nokta, çıktının bize, router'ın, konfigürasyon indirebileceği bir TFTP host'unu aradığını söylemesidir. Bu gerçekleşmediğinde, doğru setup moda geçilir. Bu size Modül 5'te gördüğümüz, Cisco router'ın varsayılan boot sıralaması ile ilgili genel bir bakış açısı kazandırır.

Router'ımızı yapılandırmaya devam edelim:

```
Press RETURN to get started!
Router>en
Router#config t
Router(config)#hostname R1
R1(config)#enable secret todd
R1(config)#int s0/0/0
R1(config-if)#ip address 10.1.2.2 255.255.255.0
R1(config-if)#Description 1st Connection to Corp Router
R1(config-if)#no shut
R1(config-if)#int s0/0/1
R1(config-if)#ip address 10.1.3.2 255.255.255.0
R1(config-if)#no shut
R1(config-if)#description 2nd connection to Corp Router
R1(config-if)#int f0/0
R1(config-if)#ip address 10.1.6.1 255.255.255.0
R1(config-if)#description Connection to HostA
R1(config-if)#no shut
R1(config-if)#int f0/1
R1(config-if)#ip address 10.1.7.1 255.255.255.0
R1(config-if)#description Connection to HostB
R1(config-if)#no shut
R1(config-if)#line con 0
R1(config-line)#password console
R1(config-line)#login
R1(config-line)#logging synchronous
R1(config-line)#exec-timeout 0 0
```

```
R1(config-line)#line aux 0
R1(config-line)#password aux
R1(config-line)#login
R1(config-line)#exit
R1(config)#line vty 0 ?
  <1-807>  Last Line number
  <cr>
R1(config)#line vty 0 807
R1(config-line)#password telnet
R1(config-line)#login
R1(config-line)#banner motd # This is my R1 ISR Router #
R1(config)#no ip domain-lookup
R1(config)#exit
R1#copy run start
Destination filename [startup-config]?[enter]
Building configuration...
[OK]
R1#
```

Interface'lerin konfigürasyonuna bir göz atalım:

```
 R1#sh run | begin interface
interface FastEthernet0/0
 description Connection to HostA
 ip address 10.1.6.1 255.255.255.0
 duplex auto
 speed auto
!
interface FastEthernet0/1
 description Connection to HostB
 ip address 10.1.7.1 255.255.255.0
 duplex auto
 speed auto
!
interface Serial0/0/0
 description 1st Connection to Corp Router
 ip address 10.1.2.2 255.255.255.0
!
interface Serial0/0/1
 description 2nd connection to Corp Router
 ip address 10.1.3.2 255.255.255.0
!
 show ip route komutu aşağıdakileri gösterir:
R1#show ip route
     10.0.0.0/24 is subnetted, 4 subnets
C       10.1.3.0 is directly connected, Serial0/0/1
```

```
C       10.1.2.0 is directly connected, Serial0/0/0
C       10.1.7.0 is directly connected, FastEthernet0/1
C       10.1.6.0 is directly connected, FastEthernet0/0
R1#
```

R1 router'u, 10.1.3.0, 10.1.2.0, 10.1.7.0 ve 10.1.6.0 network'lerine nasıl ulaşacağını bilmektedir. Şimdi, R1'den Corp router'ını ping'leyebiliyoruz:

```
R1#10.1.2.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.2.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4
ms
R1#
 Hadi, Corp router'una geri dönelim ve routing tablosuna bakalım:
Corp#sh ip route
[output cut]
     10.0.0.0/24 is subnetted, 4 subnets
C       10.1.3.0 is directly connected, Serial0/0/1
C       10.1.2.0 is directly connected, Serial0/0/0
C       10.1.1.0 is directly connected, FastEthernet0/1
Corp#
```

Seri linkler "up" olunca, şimdi her üçünü de görebiliyoruz. (DCE, ISR router'lar tarafından otomatik olarak algılandı ve interface konfigürasyonuna clock rate otomatik olarak eklendi.) Ve R2 ile R3'ü yapılandırınca, Corp router'ının routing tablosunda iki network daha göreceğiz. Corp router, henüz yapılandırılmadığından, 10.1.6.0 ve 10.1.7.0 network'lerini göremez.

## R2 Konfigürasyonu

R2'yi yapılandırmak için, diğer iki router'da yaptıklarımızla neredeyse aynı şeyleri yapacağız. Üç interface vardır: serial0/2/0, FastEthernet0/0/0 ve Dot11radio0/3/0. Router konfigürasyonuna, hostname, şifre, interface açıklaması ve bir banner eklediğimizden emin olalım:

```
Router>en
Router#config t
Router(config)#hostname R2
R2(config)#enable secret todd
R2(config)#int s0/2/0
R2(config-if)#ip address 10.1.4.2 255.255.255.0
R2(config-if)#description Connection to Corp ISR Router
R2(config-if)#no shut
R2(config-if)#int f0/0
R2(config-if)#ip address 10.1.9.1 255.255.255.0
R2(config-if)#description Connection to HostC
R2(config-if)#no shut
R2(config-if)#int dot11radio 0/3/0
R2(config-if)#ip address 10.1.8.1 255.255.255.0
```

```
R2(config-if)#description Admin WLAN
R2(config-if)#ssid ADMIN
R2(config-if-ssid)#guest-mode
R2(config-if-ssid)#authentication open
R2(config-if-ssid)#infrastructure-ssid
R2(config-if-ssid)#no shut
R2(config-if)#line con 0
R2(config-line)#password console
R2(config-line)#login
R2(config-line)#logging sync
R2(config-line)#exec-timeout 0 0
R2(config-line)#line aux 0
R2(config-line)#password aux
R2(config-line)#login
R2(config-line)#exit
R2(config)#line vty 0 ?
  <1-807>  Last Line number
  <cr>
R2(config)#line vty 0 807
R2(config-line)#password telnet
R2(config-line)#login
R2(config-line)#exit
R2(config)#banner motd # This is my R2 ISR Router #
R2(config)#no ip domain-lookup
R2(config)#^Z
R2#copy run start
Destination filename [startup-config]?[enter]
Building configuration...
[OK]
R2#
```

Wireless interface'i dışında, her şey oldukça basitti. Wireless interface'i, router'daki farklı bir interface'dir ve routing tablosunda da böyle görünür. Fakat wireless interface'ini aktifleştirmek için basit bir FastEthernet'ten daha fazla konfigürasyon gerekmektedir. Aşağıdaki çıktıyı kontrol edin, sonra, bu wireless interface için gerekli özel konfigürasyondan bahsedeceğim:

```
R3(config-if)#int dot11radio0/3/0
R2(config-if)#ip address 10.1.8.1 255.255.255.0
R3(config-if)#description Connection to Corp ISR Router
R3(config-if)#no shut
R3(config-if)#ssid ADMIN
R3(config-if-ssid)#guest-mode
R3(config-if-ssid)#authentication open
R3(config-if-ssid)#infrastructure-ssid
R2(config-if-ssid)#no shut
```

Buradaki her şey, SSID konfigürasyonuna kadar oldukça sıradandır. Bu, host'ların bağlanacağı bir wireless network'ü oluşturan, Service Set Identifier'dır. Acces point'lerin aksine, R2 router'ındaki interface, IP adresi fiziksel interface altında girildiğinden, aslında bir routed interface'dir. Genelde IP adresi, yönetim VLAN'ı ya da Bridge-Group Virtual Interface (BVI) altında girilir.

Guest-mode satırının anlamı şudur: ınterface, SSID'yi broadcast edecektir, böylece, kablosuz makinalar bu interface'e bağlanabileceklerini anlayacaktır. Authentication open, kimlik doğrulaması olmadığı anlamına gelir. (Böyle olsa bile interface'in çalışır duruma gelmesi için hala en azından bu komutu yazmak zorundasınız.) Son olarak, `infrastructure-ssid`, bu interface'in, bu altyapıdaki diğer access point'ler ya da cihazlarla (aktif kablolu network'ün kendisiyle) iletişimde kullanılabileceğini belirtir.

Fakat bekleyin, henüz tamamlamadık. Wireless istemcileri için DHCP havuzu oluşturmamız gerekmektedir:

```
R2#config t
R2(config)#ip dhcp pool Admin
R2(dhcp-config)#network 10.1.8.0 255.255.255.0
R2(dhcp-config)#default-router 10.1.8.1
R2(dhcp-config)#exit
R2(config)#ip dhcp excluded-address 10.1.8.1
R2(config)#
```

Bir router'da DHCP havuzu oluşturmak aslında oldukça basit bir prosestir. Bunu yapmak için bir havuz adı belirleyin, network/subnet ile varsayılan ağ geçidi ekleyin ve kullanılmasını istemediğiniz (default gateway gibi) adresleri hariç tutun. Genelde, bir DNS sunucusu da ekleyebilirsiniz.

Aşağıdaki `show ip route` komutunun çıktısı, 10.1.9.0, 8.0 ve 4.0 doğrudan bağlı network'leri gösterir:

```
R2#sh ip route
     10.0.0.0/24 is subnetted, 3 subnets
C       10.1.9.0 is directly connected, FastEthernet0/0
C       10.1.8.0 is directly connected, Dot11Radio0/3/0
C       10.1.4.0 is directly connected, Serial0/2/0
R2#
```

Corp, R1ve R2 router'larının şimdi tüm linkleri ***up***'tır. Fakat hala R3 (871W router) ve 1241 AP'yi yapılandırmamız gerekmektedir.

*Wireless ağları "Cisco Wireless Teknolojileri" başlıklı bölüm 12'de detaylı olarak ele alınacaktır.*

NOT

## R3 Konfigürasyonu

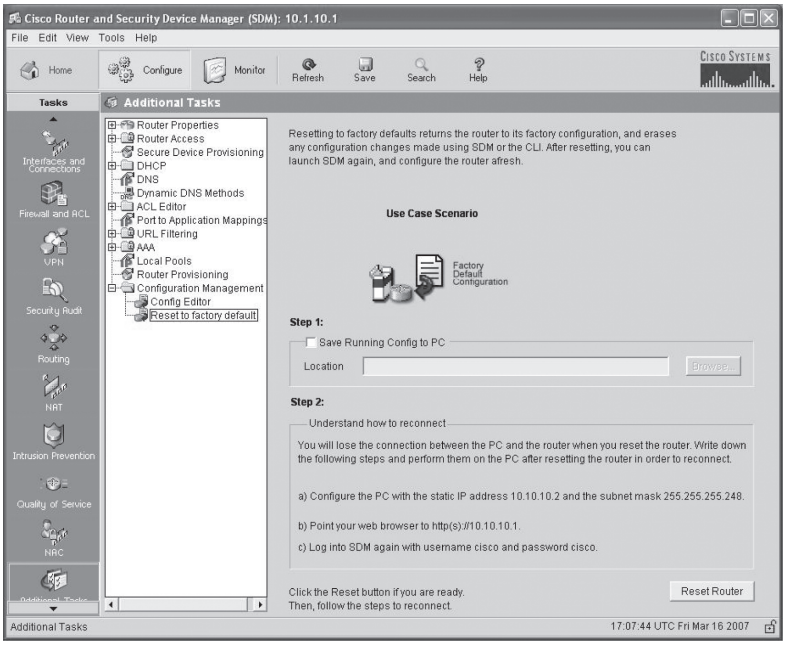
Söylediğim gibi, R3 router'ı için SDM'i kullanacağım. İlk adımım, F0/0 interface'ine bir IP adresi vermektir. PC'mi, f0/0 port'una direk bağlamak için çapraz bir kablo kullandım.

Router'ı, güvenli bir şekilde kurmak istediğim için tekrar fabrika ayarlarına döndürmeliyim. Bunu, Modül4'te gösterdiğim gibi CLI üzerinden yapabilirsiniz, fakat SDM kullanarak yapmak gerçekten çok daha basittir.

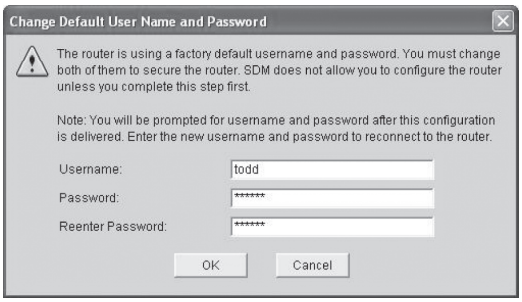
HTTP'yi kullanarak, R3 router'ına erişip Configure sayfasına gittim ve Additional Tasks'ı seçtim. Sonra, Configuration Management ve Reset to Factory Default'a tıkladım.

Alt-sağ köşedeki Reset Router butonunu tıkladım ve sonra, yukarıdaki ekran çıktısındaki yönergeleri kullanarak PC'mi yapılandırdım.

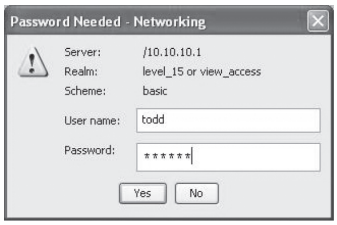
HTTPS'i kullanarak, yönergelerde belirtilen 10.10.10.1 adresini kullanarak, SDM'e tekrar bağlandım. SDM ile cisco kullanıcı adı ve cisco şifresi ile ikinci kez bağlandım. Şimdi güvenli bir bağlantım var.



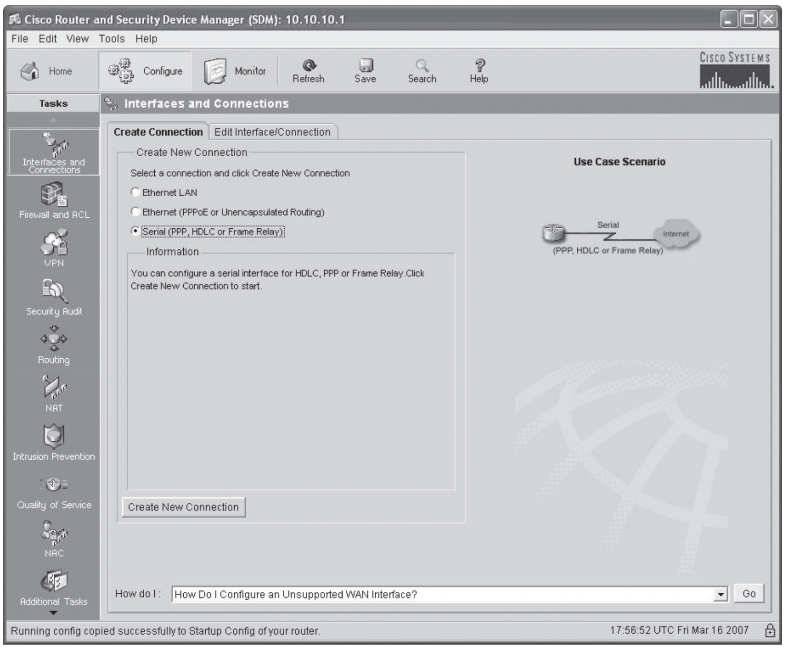

SDM yüklendikten sonra router'ın yaptığı ilk şey, kullanıcı adı ve şifreyi değiştirmek oldu.



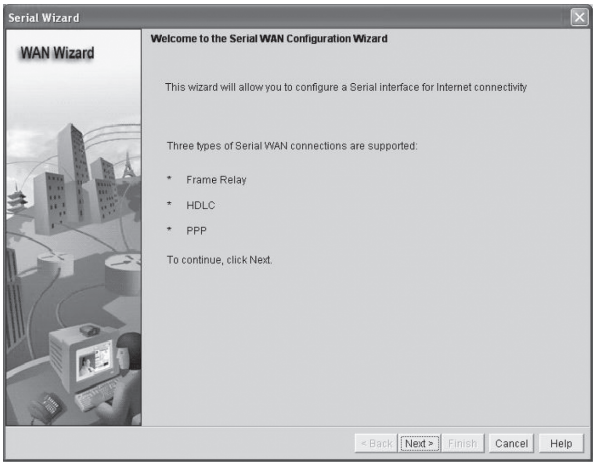

Sonra, yeni kullanıcı adım ve şifremle tekrar oturum açmam gerekti.



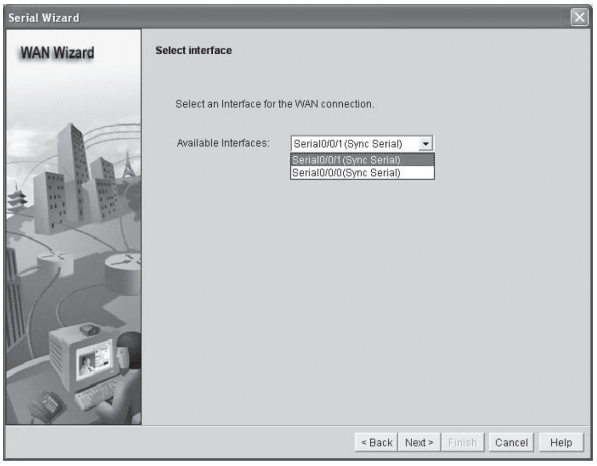

Bundan sonra, Configure ve sonra Interfaces ve Connections'ı seçtim (Üst-sol köşede, Home'un altında). Serial (PPP, HDLC veya Frame Relay) butonuna basarak, Create New Connection'ı seçtiğim yere gittim.



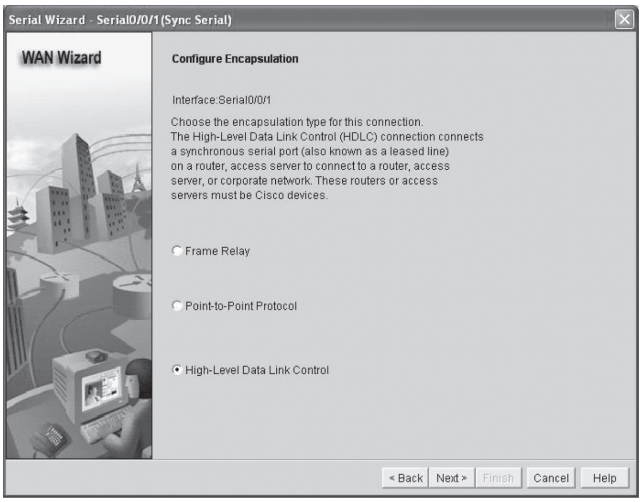

Create New Connection butonu beni WAN Wizard'ına götürdü.



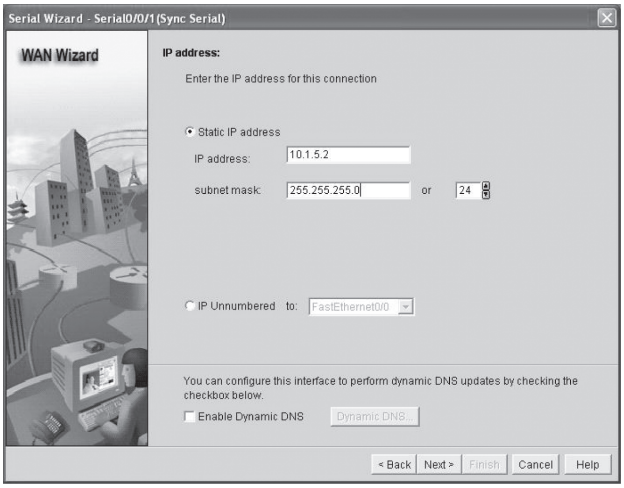

Interface'imi seçtim ve Next butonuna tıkladım.




Sonra, High-Level Data Link Control'u seçtim ve Next butonuna tıkladım.

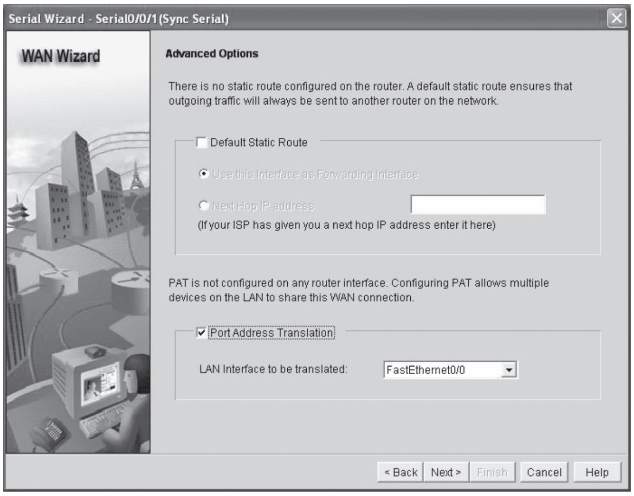



Şimdi, IP adresimi ve mask'ımı ekleyebildim.

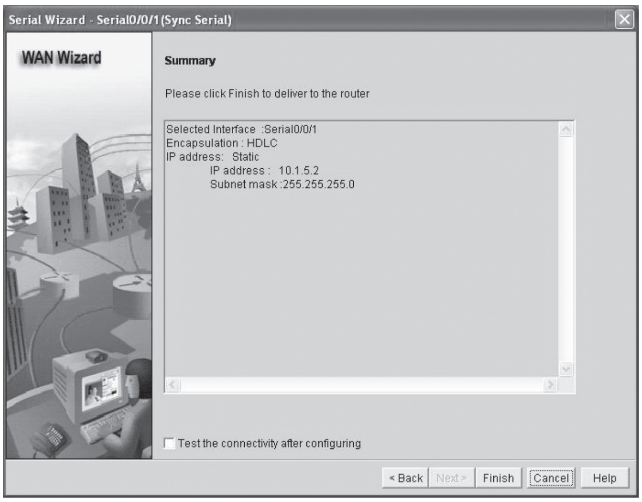



IP Unnumbered, aslında ilginç bir konfigürasyondur. Çünkü size, bir IP adresi kullanmadan network bağlantısı kurmanızı sağlar. Onun yerine, diğer aktif interface'te bir IP adresi borç alırsınız. Subnet'lerde bir bit kısaltması yaparsanız, bu oldukça kullanışlıdır.

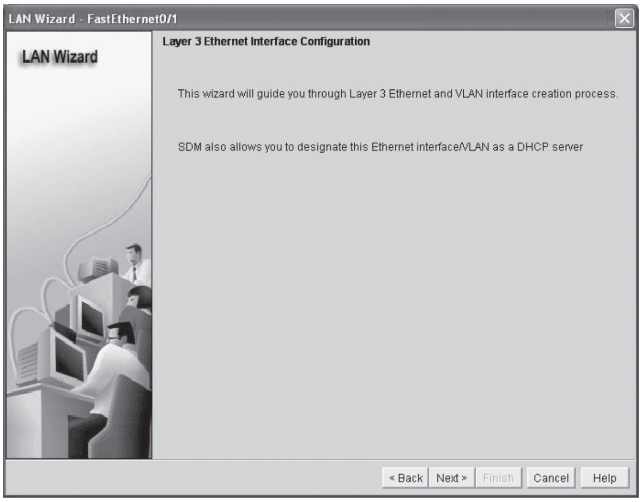
Şimdiki ekran, statik routing ya da NAT kurmayı isteyip istemediğinizi sorar. Bu ilerde üzerinde duracağımız bir şey olduğundan, onu şimdi yapılandırmayacağız.



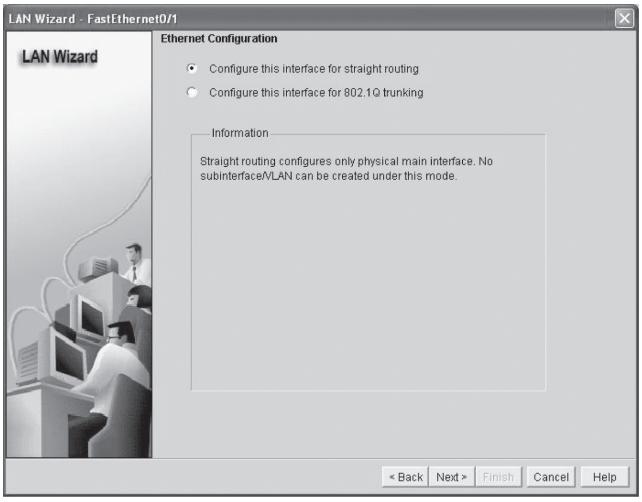

Next butonuna tıkladım ve serial0/0/1 konfigürasyonumun bir özetini aldım.



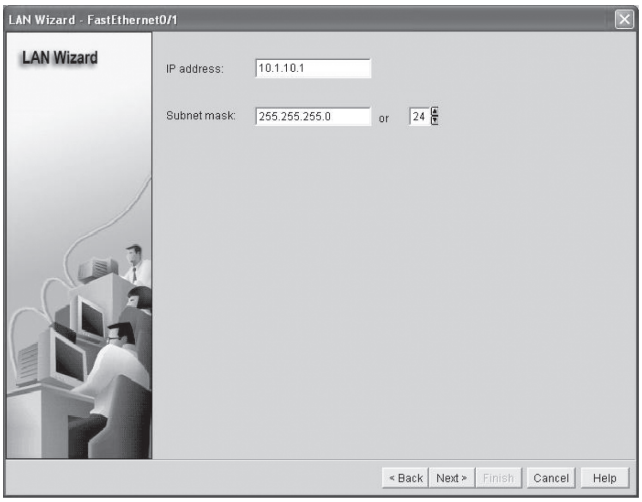

Finish butonuna tıkladım ve komutlar R3 router'ıma gönderildi. (F0/0 ve F0/1 interface'lerimi de aynı yolla yapılandıracağım.)



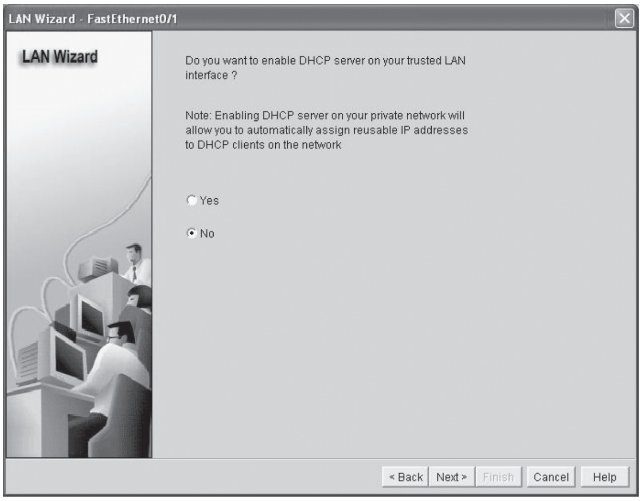

S0/0/1 interface'ini yapılandırmaya başladığım aynı lokasyondan FastEthernet0/1 interface'ini seçtikten sonra, Create New Connection'ı seçtim ve LAN Wizard'ına ulaştım.



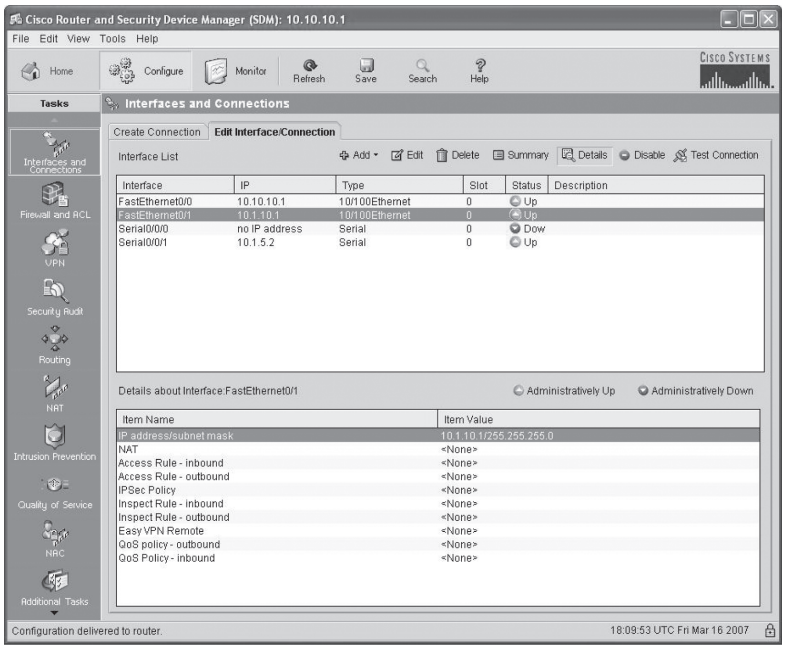

LAN Wizard'ı size, düz routing'i seçmeye (burada yapmak istediğimiz) veya 802.1Q trunking'i yapılandırmaya ("Virtual LAN'lar" başlıklı bölüm 9'da anlatacağım) izin verir. IP adresi ve maskı ayarladım ve Next butonuna tıkladım.




Şayet istersem, bu LAN için bir DHCP sunucusu oluşturması da, SDM'in çok güzel bir özelliğidir. Bu oldukça kolaydır.




Yanlışlıkla F0/1 interface için yanlış IP adresi girdiğim için onu değiştirmenin tek yolu, SDM'de Configure and Edit Interface/Connection'ı seçmek ya da CLI'ı kullanmaktır.



Buradan, FastEthernet0/1 interface'ine çift-tıklayıp, IP adresini değiştirebiliyorum.

F0/0'ı yapılandırmak için LAN Wizard'ı kullandıktan sonra, konfigürasyonu kaydettim. Sonra, PC'mi doğru network için yeniden yapılandırdım ve konfigürasyonumun doğruluğunu kontrol etmek için SDM'e tekrar bağlandım.

R3 şimdi yapılandırıldı. Konsol ve VTY şifrem, todd kullanıcısı oluşturduğumda, otomatik ayarlandığı halde, hala Configure, sonra Additional Tasks'ı seçmek ve sonrada Router Properties'inden hostname ve enable secret şifresi ayarlamak zorundayım.

### 871W Konfigürasyonu

871 router'ımı, SDM ile yapabileceğim halde, onu CLI kullanarak yapılandıracağım. İlk olarak, R3 router hariç, diğer router'larda yaptığım gibi varsayılan konfigürasyonu sileceğim ve sonra reload edeceğim.

```
Router>en
Router#config t
Router(config)#hostname 871W
871W(config)#int vlan 1
871W(config-if)#ip address 10.1.11.2 255.255.255.0
871W(config-if)#no shut
871W(config-if)#int dot11radio 0
871W(config-if)#ip address 10.1.12.1 255.255.255.0
871W(config-if)#no shut
871W(config-if)#ssid R3WLAN
871W(config-if-ssid)#guest-mode
871W(config-if-ssid)#authentication open
871W(config-if-ssid)#infrastructure-ssid
871W(config-if-ssid)#line con 0
871W(config-line)#password console
871W(config-line)#logging sync
871W(config-line)#exec-timeout 0 0
871W(config-line)#exit
871W(config)#line vty 0 ?
  <1-4>  Last Line number
  <cr>
871W(config)#line vty 0 4
871W(config-line)#password telnet
871W(config-line)#login
871W(config-line)#ip dhcp pool R3WLAN
871W(dhcp-config)#network 10.1.12.0 255.255.255.0
871W(dhcp-config)#default-router 10.1.12.1
871W(dhcp-config)#exit
871W(config)#ip dhcp excluded-address 10.1.12.1
871W(config)#exit
871W#copy run start
Destination filename [startup-config]?[enter]
Building configuration...
```

```
[OK]
871W#
```

871W, dört-port'lu bir switch'tir. Yani, IP adresini, yönetim VLAN interface'inin altında girmelisiniz. Katman 2 switch'lerinin interface'lerine basitçe IP adresi verip gidemezsiniz.

Dürüst olmak gerekirse, bu SDM kullanımından daha hızlı bir konfigürasyondu. Fakat gerçek ağlarda HTTPS ile SDM, router'ı yönetmek için daha güvenli bir yoldur. Wireless güvenliği kurmak istediğinizde, SDM'in neden kolay bir yöntem olduğunu (bölüm 12'de) göstereceğim.

Şimdi routing tablosuna bakalım:

```
 871W#sh ip route
     10.0.0.0/24 is subnetted, 2 subnets
C       10.1.11.0 is directly connected, Vlan1
C       10.1.12.0 is directly connected, Dot11Radio0
```

Her iki network'ümüz de direkt bağlı görünmektedir. Son cihazımızı da yapılandıralım ve sonra routing yapılandırmasına başlayacağız.

## 1242AP Konfigürasyonu

Router değil de Access point olmasından dolayı, 1242AP konfigürasyonu biraz farklıdır. Bu cihazı, CLI'dan yapılandıracağım, fakat bir HTTP interface'i de kullanabilirsiniz. Güvenlik eklemeye başladığımızda ve daha karmaşık konfigürasyonlara geçtiğimizde, HTTP interface kullanımı daha kolay olabilir.

Çıktıyı kontrol edin:

```
 ap>en
Password:
ap#config t
ap(config)#hostname 1242AP
1242AP(config)#enable secret todd
242AP(config)#int dot11Radio 0
1242AP(config-if)#description CORPWLAN
1242AP(config-if)#no shutdown
1242AP(config-if)#ssid CORPWLAN
1242AP(config-if-ssid)#guest-mode
1242AP(config-if-ssid)#authentication open
1242AP(config-if-ssid)#infrastructure-ssid
1242AP(config-if-ssid)#exit
1242AP(config-if)#exit
1242AP(config)#line con 0
1242AP(config-line)#password console
1242AP(config-line)#login
1242AP(config-line)#logging synchronous
1242AP(config-line)#exec-timeout 0 0
1242AP(config-line)#exit
1242AP(config)#line vty 0 ?
  <1-15>  Last Line number
  <cr>
```

```
1242AP(config)#line vty 0 15
1242AP(config-line)#password telnet
1242AP(config-line)#login
1242AP(config-line)#int bvi 1
1242AP(config-if)#ip address 10.1.1.2 255.255.255.0
1242AP(config-if)#no shut
1242AP(config-if)#exit
1242AP(config)#ip default-gateway 10.1.1.1
1242AP(config)#ip dhcp pool CORPWLAN
1242AP(dhcp-config)#network 10.1.1.0 255.255.255.0
1242AP(dhcp-config)#default-router 10.1.1.1
1242AP(dhcp-config)#exit
1242AP(config)#ip dhcp excluded-address 10.1.1.1
1242AP(config)#ip dhcp excluded-address 10.1.1.2
1242AP(config)#no ip domain-lookup
1242AP(config)#^Z
1242AP#copy run start
Destination filename [startup-config]?[enter]
Building configuration...
[OK]
1242AP#
```

SSID konfigürasyonu, R2 routed radio interface'i ile aynı olmasına rağmen Dot11radio 0 interface'i altında IP adresi olmadığına dikkat edin. Neden? Çünkü o, routed interface değildir, bu sebeple IP adresi, Bridge Virtual Interface (BVI) altında girilmiştir. Bu cihazın, LAN dışından da yönetilmesi için default gateway'de verdim.

Bilmeniz gereken, bir switch'te olduğu gibi çalışması için AP'ye IP adresi eklemeniz gerekmediğidir. Corp router'ına, wireless LAN için kolayca DHCP havuzu ekleyebiliyorum. AP'ye IP adresi ya da DHCP havuzu eklemedim ve aynı şekilde çalıştı.

# Network'ümüzde IP Routing'i Yapılandırmak

Network'ümüz iyi gidiyor, değil mi? Bunlardan sonra, IP adreslemesi, yönetimsel fonksiyonlar ve hatta saat denetimi (ISR router'larda otomatik olarak) doğru bir şekilde yapılandırılmıştır. Fakat uzak ağlara nasıl gideceğini anlamak için, routing tablosuna bakmanın, paketleri göndermek için tek yöntem olduğunda, bir router uzak ağlara paketleri nasıl gönderir? Yapılandırılan router'larımız, routing tablolarında sadece doğrudan bağlı network'ler hakkında bilgiye sahiptirler. Bir router, routing tablosunda yer almayan bir network için paket aldığında ne olur? Uzak ağı bulmak için broadcast göndermez, onu atar.

Bu nedenle, henüz tam olarak hazır değiliz. Fakat endişe etmeyin. Paketlerin gönderilmesi için, küçük ağ topluluğumuzda tüm ağları içermesi için routing tablomuzu yapılandırmanın birkaç yolu vardır. Bir network için en iyi olan, başkası için gerekli olmayabilir. Farklı routing türlerini anlamak, belirli ortam ve iş gerekliliklerinize için en iyi çözümü bulmanızda gerçekten yardımcı olacaktır.

Aşağıdaki bölümlerde şu routing türlerini öğreneceksiniz:

- Statik routing
- Default routing
- Dinamik routing

Statik routing'i oluşturup onu çalışır hale getirmeniz, ağ topluluğunu iyi bildiğiniz anlamına geldiğinden, network'ümüzdeki statik routingi açıklayıp uygulayarak başlayacağım. Hadi başlayalım!

## Statik Routing

Her router'ın, routing tablosuna manuel olarak route'lar girdiğinizde, statik routing olur. Statik route'a lehte ve aleyhte olanlar vardır, fakat tüm routing prosesleri için bu böyledir.

Statik routing, aşağıdaki faydaları sağlamaktadır:

- Router CPU'suna ek yük getirmez. Yani, dinamik routing kullandığınızdan daha ucuz bir router almanız mümkündür.
- Router'lar arasında bant genişliği kullanımı yoktur. Yani, WAN linklerinde, para tasarrufu sağlayabilirsiniz.
- Yöneticiler, sadece belirli network'lere routing erişimine izin vermeyi seçebildiklerinden, güvenlik sağlar.

Statik routing aşağıdaki dezavantajlara sahiptir:

- Yönetici, ağ topluluğunu ve doğru yapılandırmak için her router'ın nasıl bağlandığını iyi bilmelidir.
- Ağ topluluğuna bir network eklenirse yönetici elle, her router'da ona bir route eklemelidir.
- Büyük network'lerde uygulanabilir değildir, çünkü onun devamını sağlamak, tam-zamanlı bir iştir.

Aşağıda, bir routing tablosuna statik route eklemek için kullandığınız komut dizisi vardır:

```
ip route [destination_network] [mask] [next-hop_address or
  exitinterface] [administrative_distance] [permanent]
```

Dizgide ki her komut aşağıda açıklanmaktadır:

**`ip route`:** Statik route oluşturmak için kullanılan komut.

***`destination_network`:*** Routing tablosuna yerleştirilen network.

***`mask`:*** Ağda kullanılan subnet maskı.

***`next-hop_address`:*** Paketi alıp onu uzak network'e gönderecek next-hop router'ın adresidir. Bu, direk bağlı network'lerdeki bir router interface'idir. Route eklemeden önce router interface'ini pingleyebilmelisiniz. Şayet yanlış next-hop adresi girerseniz yada bu router'a interface'i down ise, statik route, router konfigürasyonunda görünecektir, fakat routing tablosunda görünmeyecektir.

***`exitinterface`:*** İsterseniz, next-hop adresinin yerine kullanılır ve direkt bağlı bir route gibi görünür.

***`administrative_distance`:*** Varsayılan olarak, statik route'un administrative distance'ı 1'dir (next-hop adresi yerine bir çıkış interface'i kullanırsanız, 0'dır) Komutun sonuna bir administrative weight ekleyerek varsayılan değeri değiştirebilirsiniz. Dinamik routing bölümüne geçince, bu konu hakkında daha çok bahsedeceğim.

**`permanent`:** Şayet interface kapalıysa yada router, next-hop router ile haberleşemiyorsa, route, routing tablosundan otomatik olarak atılacaktır. *Permanent* seçeneği, ne olursa olsun, routing tablosundaki kaydı korur.

Statik route konfigürasyonuna girmeden önce, örnek bir statik route'a ve ondan ne anladığımıza bir bakalım.

```
Router(config)#ip route 172.16.3.0 255.255.255.0 192.168.2.4
```

- `ip route` komutu, bunun bir statik route olduğunu söyler.
- 172.16.3.0, paketleri göndermek istediğimiz uzak network'tür.
- 255.255.255.0, uzak network'ün maskıdır.
- 192.168.2.4, paketleri göndereceğimiz next hop ya da router'dır.

Ancak, statik route'ın şöyle olması düşünülür:

```
Router(config)#ip route 172.16.3.0 255.255.255.0 192.168.2.4 150
```

Sondaki 150, 1 olan varsayılan administrative distance(AD)'ı, 150 olarak değiştirir. Endişelenmeyin, Dinamik routing'e geçtiğimizde, AD'den daha fazla bahsedeceğim. Şimdi, AD'nin bir route'ın güvenilirliği olduğunu hatırlayın (0, en iyisi, 255 en kötüsüdür).

Bir örnek daha yapalım, sonra konfigürasyona geçeceğim:

```
Router(config)#ip route 172.16.3.0 255.255.255.0 s0/0/0
```

Next-hop adresi kullanmak yerine, bir çıkış interface'i kullanabiliriz. Bu, route'u, direk bağlı network gibi gösterir. İşlevsel olarak, next hop ve çıkış interface'i tamamen aynıdır. Statik route'ın nasıl çalıştığını anlamanıza yardımcı olması için, daha önce Şekil 6.9'da gösterilen ağ topluluğundaki konfigürasyonu göstereceğim.

## Corp

Her routing tablosu, direk bağlı network'leri otomatik olarak görür. Ağ topluluğundaki tüm network'e route olabilmesi için, routing tablosunun, diğer network'lerin nerede olduklarını ve onlara nasıl erişileceğini tanımlayan bilgiye sahip olması gerekir.

Corp router, beş network'e bağlıdır. Corp router'ın tüm network'lere route'a sahip olması için, aşağıdaki network'ler, routing tablosunda yapılandırılmalıdır:

- 10.1.6.0
- 10.1.7.0
- 10.1.8.0
- 10.1.9.0
- 10.1.10.0
- 10.1.11.0
- 10.1.12.0

Aşağıdaki router çıktısı, Corp router'ındaki statik route'ları ve konfigürasyon sonrası routing tablosunu gösterir. Corp router'ının, uzak ağları bulması için, uzak network'ü, uzak maskı ve paketlerin nereye gönderileceğini açıklayan bir kaydı routing tablosuna girdim. Administrative distance'ı yükseltmek için her satırın sonuna "150" ekleyeceğim. (dinamik routinge geçtiğimde, bunu neden kullandığımı göreceksiniz.)

```
Corp(config)#ip route 10.1.6.0 255.255.255.0 10.1.2.2 150
Corp(config)#ip route 10.1.6.0 255.255.255.0 10.1.3.2 151
Corp(config)#ip route 10.1.7.0 255.255.255.0 10.1.3.2 150
Corp(config)#ip route 10.1.7.0 255.255.255.0 10.1.2.2 151
Corp(config)#ip route 10.1.8.0 255.255.255.0 10.1.4.2 150
Corp(config)#ip route 10.1.9.0 255.255.255.0 10.1.4.2 150
Corp(config)#ip route 10.1.10.0 255.255.255.0 10.1.5.2 150
```

```
Corp(config)#ip route 10.1.11.0 255.255.255.0 10.1.5.2 150
Corp(config)#ip route 10.1.12.0 255.255.255.0 10.1.5.2 150
Corp(config)#do show run | begin ip route
ip route 10.1.6.0 255.255.255.0 10.1.2.2 150
ip route 10.1.6.0 255.255.255.0 10.1.3.2 151
ip route 10.1.7.0 255.255.255.0 10.1.3.2 150
ip route 10.1.7.0 255.255.255.0 10.1.2.2 151
ip route 10.1.8.0 255.255.255.0 10.1.4.2 150
ip route 10.1.9.0 255.255.255.0 10.1.4.2 150
ip route 10.1.10.0 255.255.255.0 10.1.5.2 150
ip route 10.1.11.0 255.255.255.0 10.1.5.2 150
ip route 10.1.12.0 255.255.255.0 10.1.5.2 150
```

10.1.6.0 ve 10.1.7.0 network'leri için, her network'e iki yolu ekledim, fakat linklerden birinin AD'sini 151 yaptım. Diğer linkin arızalanması durumunda bu yedek bir route olacaktır. Şayet ikisine de aynı AD'yi verseydim, bir routing kısır döngüsü oluşacaktı. (Statik routing, aynı hedefe çok sayıda linki kullanamaz.) Router yapılandırıldıktan sonra, statik route'ları görmek için `show ip route` yazabilirsiniz:

```
Corp(config)#do show ip route
     10.0.0.0/24 is subnetted, 12 subnets
S       10.1.11.0 [150/0] via 10.1.5.2
S       10.1.10.0 [150/0] via 10.1.5.2
S       10.1.9.0 [150/0] via 10.1.4.2
S       10.1.8.0 [150/0] via 10.1.4.2
S       10.1.12.0 [150/0] via 10.1.5.2
C       10.1.3.0 is directly connected, Serial0/0/1
C       10.1.2.0 is directly connected, Serial0/0/0
C       10.1.1.0 is directly connected, FastEthernet0/1
S       10.1.7.0 [150/0] via 10.1.3.2
S       10.1.6.0 [150/0] via 10.1.2.2
C       10.1.5.0 is directly connected, Serial0/2/0
C       10.1.4.0 is directly connected, Serial0/1/0
```

Corp router, route etmek için yapılandırıldı ve tüm network'lere giden route'ları bilmektedir. R1'de, her uzak network'e iki route yapılandırdım, fakat routing tablosu sadece düşük AD olanını gösterecektir. Diğer link sadece, kullanılan düşük AD'li link arızalandığında routing tablosunda görünecektir.

Şayet route'lar, routing tablosunda görünmezse, router'ın, yapılandırdığınız next-hop router ile haberleşemediğini anlamanızı istiyorum. Next-hop cihazıyla bağlantı kurulamasa dahi, routing tablosundaki route'u saklamaya devam etmek için permanent parametresini kullanabilirsiniz.

Yukarıdaki routing tablosundaki S'in anlamı, network'ün bir statik kayıt olduğudur. [1/0], uzak network'e, administrative distance ve metriği belirtir. Burada, netxt-hop interface'inin 0 olması, onun direk bağlı olduğunu işaret eder.

NOT *Statik konfigürasyonun sonundaki 150/151 için kaygılanmayın. Bu modülde, en kısa sürede bundan bahsedeceğim. Bu noktada, bundan endişelenmemeniz gerektiğinden emin olabilirsiniz.*

Tamam, iyi gidiyoruz. Corp router şimdi, tüm uzak ağlarla haberleşmesi için gerekli bilgilere sahiptir. Fakat unutmayın ki, R1,

R2, R3 ve 871W router'ları, aynı bilgilerle yapılandırılmadıysa, paketler, atılacaktır. Bunu, statik route'lar yapılandırarak düzeltebiliriz.

## R1

R1 router'ı, 10.1.2.0, 10.1.3.0, 10.1.6.0 ve 10.1.7.0 ağlarına direk bağlıdır, bu nedenle, R1 router'ında, aşağıdaki statik route'ları oluşturduk:

- 10.1.1.0
- 10.1.4.0
- 10.1.5.0
- 10.1.8.0
- 10.1.9.0
- 10.1.10.0
- 10.1.11.0
- 10.1.12.0

R1 router'ı için konfigürasyon şöyle olacaktır; direk bağlı olduğumuz network'ler için statik route oluşturmadığımızı ve Corp ve R1 router'ları arasında iki linkimiz olduğundan, next hop olarak, 10.1.2.1 ya da 10.1.3.1'i kullanabileceğimizi hatırlayın. Next hop'ları değiştireceğim, böylece, tüm veri, tek linkten gitmeyecek. Statik routing ile load-balance yapmadığımdan, hangi linki kullanacağım önemli değildir. RIP, EIGRP ve OSPF gibi dinamik routing kullandığımızda, load-balance yapabileceğiz, fakat şimdi, linkler, her network'e bir yedek route sağlayacaktır. Çıktıyı kontrol edelim:

```
R1(config)#ip route 10.1.1.0 255.255.255.0 10.1.2.1 150
R1(config)#ip route 10.1.1.0 255.255.255.0 10.1.3.1 151
R1(config)#ip route 10.1.4.0 255.255.255.0 10.1.2.1 150
R1(config)#ip route 10.1.4.0 255.255.255.0 10.1.3.1 151
R1(config)#ip route 10.1.5.0 255.255.255.0 10.1.2.1 150
R1(config)#ip route 10.1.5.0 255.255.255.0 10.1.3.1 151
R1(config)#ip route 10.1.8.0 255.255.255.0 10.1.3.1 150
R1(config)#ip route 10.1.8.0 255.255.255.0 10.1.2.1 151
R1(config)#ip route 10.1.9.0 255.255.255.0 10.1.3.1 150
R1(config)#ip route 10.1.9.0 255.255.255.0 10.1.2.1 151
R1(config)#ip route 10.1.10.0 255.255.255.0 10.1.3.1 150
R1(config)#ip route 10.1.10.0 255.255.255.0 10.1.2.1 151
R1(config)#ip route 10.1.11.0 255.255.255.0 10.1.3.1 150
R1(config)#ip route 10.1.11.0 255.255.255.0 10.1.2.1 151
R1(config)#ip route 10.1.12.0 255.255.255.0 10.1.3.1 150
R1(config)#ip route 10.1.12.0 255.255.255.0 10.1.2.1 151
R1(config)#do show run | begin ip route
ip route 10.1.1.0 255.255.255.0 10.1.2.1 150
ip route 10.1.1.0 255.255.255.0 10.1.3.1 151
ip route 10.1.4.0 255.255.255.0 10.1.2.1 150
ip route 10.1.4.0 255.255.255.0 10.1.3.1 151
ip route 10.1.5.0 255.255.255.0 10.1.2.1 150
ip route 10.1.5.0 255.255.255.0 10.1.3.1 151
```

```
ip route 10.1.8.0 255.255.255.0 10.1.3.1 150
ip route 10.1.8.0 255.255.255.0 10.1.2.1 151
ip route 10.1.9.0 255.255.255.0 10.1.3.1 150
ip route 10.1.9.0 255.255.255.0 10.1.2.1 151
ip route 10.1.10.0 255.255.255.0 10.1.3.1 150
ip route 10.1.10.0 255.255.255.0 10.1.2.1 151
ip route 10.1.11.0 255.255.255.0 10.1.3.1 150
ip route 10.1.11.0 255.255.255.0 10.1.2.1 151
ip route 10.1.12.0 255.255.255.0 10.1.3.1 150
ip route 10.1.12.0 255.255.255.0 10.1.2.1 151
```

Her network'e iki yol yapılandırdığımdan, bu oldukça uzun bir konfigürasyon oldu. Routing tablosuna bakarak, R1 router'ının, tüm network'lere nasıl ulaşacağını bildiğini görebilirsiniz:

```
R1(config)#do show ip route
     10.0.0.0/24 is subnetted, 12 subnets
S       10.1.11.0 [150/0] via 10.1.3.1
S       10.1.10.0 [150/0] via 10.1.3.1
S       10.1.9.0 [150/0] via 10.1.3.1
S       10.1.8.0 [150/0] via 10.1.3.1
S       10.1.12.0 [150/0] via 10.1.3.1
C       10.1.3.0 is directly connected, Serial0/0/1
C       10.1.2.0 is directly connected, Serial0/0/0
S       10.1.1.0 [150/0] via 10.1.2.1
C       10.1.7.0 is directly connected, FastEthernet0/1
C       10.1.6.0 is directly connected, FastEthernet0/0
S       10.1.5.0 [150/0] via 10.1.2.1
S       10.1.4.0 [150/0] via 10.1.2.1
```

NOT *Yüksek administrative distance 'a sahip route'un, düşük olanın kullanım dışı kalması dışında routing tablosunda görünmeyeceğini hatırlayın.*

R1 router, artık komple routing tablosuna sahiptir. Ağ topluluğundaki diğer router'lar, routing tablolarında bütün ağlarla ilgili bilgiye sahip olur olmaz, R1, tüm uzak ağlarla haberleşebilecektir.

## R2

R2 router'ı, 10.1.4.0, 10.1.8.0 ve 10.1.9.0 ağlarına direk bağlıdır, bu nedenle, şu route'lar eklenmelidir:

- 10.1.1.0
- 10.1.2.0
- 10.1.3.0
- 10.1.5.0
- 10.1.6.0
- 10.1.7.0
- 10.1.10.0
- 10.1.11.0
- 10.1.12.0

R2 router konfigürasyonu aşağıdaki gibidir:

```
R2(config)#ip route 10.1.1.0 255.255.255.0 10.1.4.1 150
R2(config)#ip route 10.1.2.0 255.255.255.0 10.1.4.1 150
R2(config)#ip route 10.1.3.0 255.255.255.0 10.1.4.1 150
R2(config)#ip route 10.1.5.0 255.255.255.0 10.1.4.1 150
R2(config)#ip route 10.1.6.0 255.255.255.0 10.1.4.1 150
R2(config)#ip route 10.1.7.0 255.255.255.0 10.1.4.1 150
R2(config)#ip route 10.1.10.0 255.255.255.0 10.1.4.1 150
R2(config)#ip route 10.1.11.0 255.255.255.0 10.1.4.1 150
R2(config)#ip route 10.1.12.0 255.255.255.0 10.1.4.1 150
R2(config)#do show run | begin ip route
ip route 10.1.1.0 255.255.255.0 10.1.4.1 150
ip route 10.1.2.0 255.255.255.0 10.1.4.1 150
ip route 10.1.3.0 255.255.255.0 10.1.4.1 150
ip route 10.1.5.0 255.255.255.0 10.1.4.1 150
ip route 10.1.6.0 255.255.255.0 10.1.4.1 150
ip route 10.1.7.0 255.255.255.0 10.1.4.1 150
ip route 10.1.10.0 255.255.255.0 10.1.4.1 150
ip route 10.1.11.0 255.255.255.0 10.1.4.1 150
ip route 10.1.12.0 255.255.255.0 10.1.4.1 150
```

Aşağıdaki çıktı, R2 router'ındaki routing tablosunu göstermektedir:

```
R2(config)#do show ip route
     10.0.0.0/24 is subnetted, 12 subnets
S       10.1.11.0 [150/0] via 10.1.4.1
S       10.1.10.0 [150/0] via 10.1.4.1
C       10.1.9.0 is directly connected, FastEthernet0/0
C       10.1.8.0 is directly connected, Dot11Radio0/3/0
S       10.1.12.0 [150/0] via 10.1.4.1
S       10.1.3.0 [150/0] via 10.1.4.1
S       10.1.2.0 [150/0] via 10.1.4.1
S       10.1.1.0 [150/0] via 10.1.4.1
S       10.1.7.0 [150/0] via 10.1.4.1
S       10.1.6.0 [150/0] via 10.1.4.1
S       10.1.5.0 [150/0] via 10.1.4.1
C       10.1.4.0 is directly connected, Serial0/2/0
```

R2, şimdi ağ topluluğundaki 12 network'ün hepsini göstermektedir, bu nedenle artık o da tüm network ve router'larla (şimdiye kadar yapılandırılmış olanlarla) haberleşebilecektir.

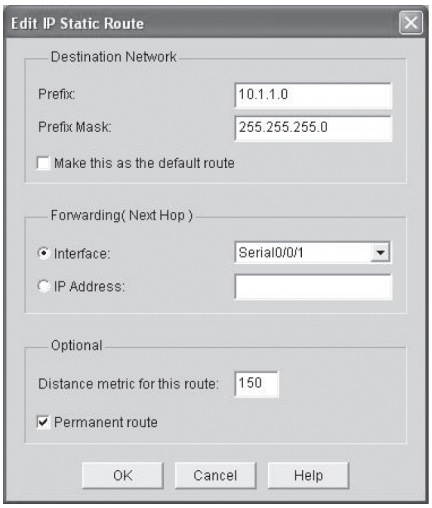
## R3

R3 router'ı, 10.1.5.0, 10.1.10.0 ve 10.1.11.0 ağlarına direk bağlıdır, bu nedenle, şu routeları eklememiz gerekir:

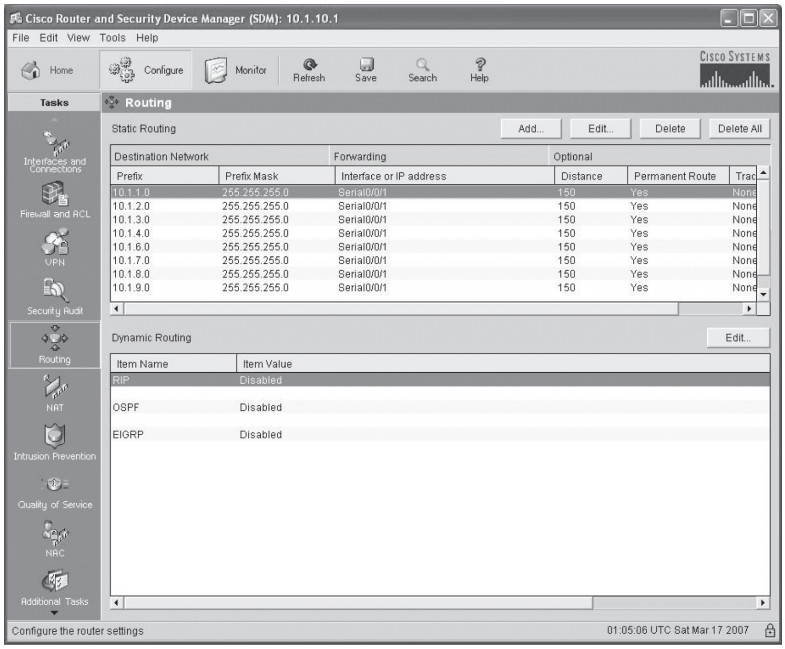
- 10.1.1.0
- 10.1.2.0

- 10.1.3.0
- 10.1.4.0
- 10.1.6.0
- 10.1.7.0
- 10.1.8.0
- 10.1.9.0
- 10.1.12.0

Önceden yaptığım gibi, R3 router'ına statik routing yapılandırmak için SDM'i kullanacağım. Konfigürasyon oldukça basittir. Next-hop adresi ya da çıkış interface'ini kullanabilirim. Mümkün olduğu kadar kısa yazmayı sevdiğimden, sadece bir mouse tıklaması olduğundan, çıkış interface'ini kullanacağım.




Tüm route'ları yapılandırdıktan sonra, onları routing ekranında görebiliriz.




Bu ekrandan, statik route'ları düzenlemek kolaydır.

Şimdi, SDM'den router'a yüklenen konfigürasyona ve routing tablosuna bir bakalım:

```
R3#show run | begin ip route
ip route 10.1.1.0 255.255.255.0 Serial0/0/1 150 permanent
ip route 10.1.2.0 255.255.255.0 Serial0/0/1 150 permanent
ip route 10.1.3.0 255.255.255.0 Serial0/0/1 150 permanent
ip route 10.1.4.0 255.255.255.0 Serial0/0/1 150 permanent
ip route 10.1.6.0 255.255.255.0 Serial0/0/1 150 permanent
ip route 10.1.7.0 255.255.255.0 Serial0/0/1 150 permanent
ip route 10.1.8.0 255.255.255.0 Serial0/0/1 150 permanent
ip route 10.1.9.0 255.255.255.0 Serial0/0/1 150 permanent
ip route 10.1.12.0 255.255.255.0 FastEthernet0/1 150 permanent
R3#show ip route
     10.0.0.0/24 is subnetted, 12 subnets
C       10.1.11.0 is directly connected, FastEthernet0/1
C       10.1.10.0 is directly connected, FastEthernet0/0
S       10.1.9.0 is directly connected, Serial0/0/1
S       10.1.8.0 is directly connected, Serial0/0/1
S       10.1.12.0 is directly connected, FastEthernet0/1
S       10.1.3.0 is directly connected, Serial0/0/1
S       10.1.2.0 is directly connected, Serial0/0/1
S       10.1.1.0 is directly connected, Serial0/0/1
S       10.1.7.0 is directly connected, Serial0/0/1
S       10.1.6.0 is directly connected, Serial0/0/1
C       10.1.5.0 is directly connected, Serial0/0/1
S       10.1.4.0 is directly connected, Serial0/0/1
R3#
```

`Show ip route` komut çıktısına bakarak, statik route'ların direkt bağlı olarak listelendiğini görebilirsiniz. Garip? Değil, çünkü next-hop adres yerine, çıkış interface'i kullandığım ve işlevsel olarak fark yoktur. Gerçekte, `permanent` komutu kullanmamıza gerek yoktur. Çünkü tek yapacağı, bu linke giden link arızalansa bile, route'un, routing tablosunda kalmasını sağlamaktır. `Permanent` komutunu sırf SDM ile yapması kolay olduğundan (tek fare tıklaması) yapılandırdım. Neredeyse tamamlamak üzereyiz, sadece 871W kaldı.

## 871W

Şimdi, bu router için 871W'i stub olarak yapılandıracağımdan dolayı default routing kullanacağım. Stub, bu tasarımdaki kablosuz ağların, diğer ağlara ulaşmak için sadece bir yola sahip olduklarını belirtir. Şimdiki bölümde, konfigürasyonu size gösterip network'ün doğruluğunu kontrol ettikten sonra default routing'ten detaylı bahsedeceğim. İşte konfigürasyon:

```
871W(config)#ip route 0.0.0.0 0.0.0.0 10.1.11.1
871W(config)#ip classless
871W(config)#do show ip route
     10.0.0.0/24 is subnetted, 2 subnets
C       10.1.11.0 is directly connected, Vlan1
C       10.1.12.0 is directly connected, Dot11Radio0
S*   0.0.0.0/0 [1/0] via 10.1.11.1
871W(config)#
```

Bu, çok daha kolay gözüküyor, değil mi? Evet öyledir, fakat bunu tüm router'larda yapamazsınız, sadece stub network'lerde yapabilirsiniz. R1 ve R2 router'larında da, default routing kullanabilirdim ve kolayca yapabileceğim halde, 150'yi bu default route'a eklemedim. Sonradan dinamik route kullandığımızda route'u kaldırmak oldukça basit olduğundan, bunu yapmadım.

Böylece tamamladık. Tüm router'lar doğru routing tablosuna sahiptir. Bu nedenle, tüm router ve host'lar problemsiz haberleşebilmelidir. Şayet ağ topluluğuna bir tane daha network ya da router eklersek, her router'ın routing tablosunu elle güncellemek zorundasınız. Şayet küçük bir network'ünüz varsa, bu bir problem sayılmaz. Fakat büyük bir ağ topluluğuyla uğraşıyorsanız, çok fazla zaman alacaktır.

## Konfigürasyonunuzun Doğruluğunu Kontrol Etmek

Henüz tamamlayamadık. Tüm router'ların routing tablosu yapılandırılınca, doğruluklarının kontrol edilmesi gerekir. Bunu yapmanın en iyi yolu, `show ip route` komutunu kullanmanın yanında, Ping programını kullanmaktır. 1242 AP'den 871W router'ını pingleyerek başlayacağım.

İşte çıktısı:

```
871W#ping 10.1.1.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.1.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4
ms

Router871W'dan HostA, B, C ve D'ye ping atmak, iyi bir IP
bağlanırlığı testi olacaktır. Router çıktısı şöyledir:
871W#ping 10.1.6.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.6.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max =
4/6/12 ms
871W#ping 10.1.7.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.7.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/4/4
ms
871W#ping 10.1.9.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.9.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/4/4
ms
871W#ping 10.1.10.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.10.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5)
```

Ayrıca, paketin HostA'ya gidene kadarki uğradığı noktaları görmek için, 871W router'dan trace kullanabiliriz:

```
871W#trace 10.1.6.2
Type escape sequence to abort.
Tracing the route to 10.1.6.2
  1 10.1.11.1 0 msec 0 msec 0 msec
  2 10.1.5.1 4 msec 0 msec 4 msec
  3 10.1.2.2 0 msec 0 msec 4 msec
  4 10.1.6.2 4 msec 4 msec *
```

Uçtan uca ve her host'la, problemsiz haberleşebildiğimiz için statik route konfigürasyonumuz başarılıdır!

## Default Routing

Default routing'i, routing tablosunda olmayan uzak bir hedef network için paketleri, next-hop router'a göndermek için kullanırız. Default routing'i sadece, network'ten dışarı tek çıkış yolu olan stub network'lerde kullanmalısınız.

Önceki bölümde kullanılan ağlar arası haberleşme örneğinde, stub bir network'te olduğu düşünülecek router'lar, R1, R2 ve 871W'dir. R3 router'ına bir default route koymayı denerseniz, diğer router'lara yönlendirilmiş birden fazla interface olduğundan, paketler, doğru network'lere gönderilmeyecektir. Default routing ile kolayca kısır döngüler oluşturabilirsiniz. Bu yüzden çok dikkatli olun!

Bir default route oluşturmak için statik route'un network adres ve mask lokasyonlarında wildcard maskları kullanırsınız (871W konfigürasyonunda gösterdiğim gibi). Aslında, default route'u, network ve mask bilgisinin yerine wildcard'lar kullanan bir statik route olarak düşünebilirsiniz.

Bir default route kullanarak, sadece bir statik route kaydı oluşturabilirsiniz. Bu, tüm route'ları yazmaktan daha kolaydır!

```
871W(config)#ip route 0.0.0.0 0.0.0.0 10.1.11.1
871W(config)#ip classless
871W(config)#do show ip route
  Gateway of last resort is 10.1.11.1 to network 0.0.0.0
 10.0.0.0/24 is subnetted, 2 subnets
C       10.1.11.0 is directly connected, Vlan1
C       10.1.12.0 is directly connected, Dot11Radio0
S*   0.0.0.0/0 [1/0] via 10.1.11.1
871W(config)#
```

Routing tablosuna baktığınızda, sadece iki direkt bağlı network ve bu kaydın bir default route adayı olduğunu belirten bir S* göreceksiniz. Default route komutunu başka bir şekilde tamamlayabilirdim:

```
871W(config)#ip route 0.0.0.0 0.0.0.0 vlan1
```

Burada bize söylenen, "routing tablosunda bir network için kayıt yoksa onu VLAN1'e gönder" dir. (Onu FastEthernet0/0'dan gönderecektir.) Next-hop router'ın IP adresini ya da çıkış interface'ini seçebilirsiniz, her ikisinde de, aynı şekilde çalışacaktır. Bu çıkış interface konfigürasyonunu, R3 statik router yapılandırmasında kullandığımı hatırlayın.

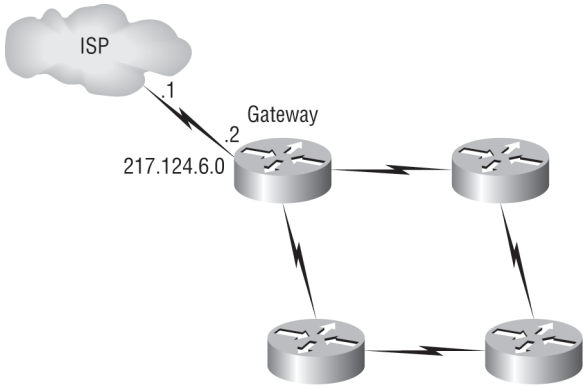
Ayrıca, routing tablosunda gateway of last resort'un ayarlandığına dikkat edin. Böyle olsa da, default route'lar kullanırken, bilmeniz gereken bir komut daha vardır: ***ip classless*** komutu.

Tüm Cisco router'ları, classful router'lardır. Yani, onlar router'ın her interface'inde varsayılan subnet maskı olmasını bekler. Router, routing tablosunda olmayan bir hedef subnet için paket alırsa varsayılan olarak bu paket atılacaktır. Şayet default routing kullanırsanız, routing tablosunda uzak subnet'lerin olmaması mümkün olacağından, ***ip classless*** komutu kullanılmalıdır.

Router'larımda, 12.4 IOS versiyonu olduğundan, `ip classless` komutu varsayılanda aktiftir. Şayet, default routing kullanıyorsanız ve bu komut konfigürasyonunuzda yoksa router'larınızdaki, network'leri subnet'lediyseniz, onu eklemeniz gerekecektir. Komut şöyle kullanılmaktadır:

```
871W(config)#ip classless
```

Onun bir global configuration mod komutu olduğuna dikkat edin. `İp classless` komutunun ilginç tarafı, onsuz, default routing'in bazen çalışması, bazen çalışmamasıdır. Güvenilir olması için, default routing kullandığınızda, daima `ip classless` komutunu çalıştırın.




***Şekil 6.10:*** *Bir gateway of last resort yapılandırmak.*

Bir gateway of last resort yapılandırmak için kullanabileceğiniz diğer bir komut, ***ip default-network*** komutudur. Şekil 6.10, yapılandırılmış bir gateway of last resort ibaresi olması gereken bir ağı göstermektedir.

ISP için gateway router'a bir gateway of last resort eklemek için kullanılan üç komut şöyledir (hepside aynı çözümü sağlar):

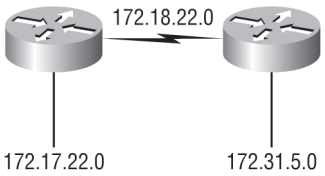
```
Gateway(config)#ip route 0.0.0.0 0.0.0.0 217.124.6.1

Gateway(config)#ip route 0.0.0.0 0.0.0.0 s0/0

Gateway(config)#ip default-network 217.124.6.0
```

Daha önce söylediğim gibi, bu komutların üçü de, gateway of last resort oluşturacaktır, fakat aralarında bazı farklılıklar vardır. İlk olarak, çıkış interface çözümü, 0 AD'ye sahip olduğundan, diğer ikisinden daha çok düşünülür. Ayrıca, `ip default-network` komutu, bir IGP (RIP gibi) yapılandırdığınızda, default network'ü yayınlar. Bundan dolayı, ağ topluluğunuzdaki diğer router'lar, bu route'ı varsayılan olarak otomatik alır.

Şayet bir default route yanlış yapılandırılırsa, ne olur? Bir `show ip route` komut çıktısına bakalım, Şekil 6.11'deki network ile mukayese edelim ve bir problem bulup bulamayacağınızı görelim:

```
Router#sh ip route
[output cut]
```

```
Gateway of last resort is 172.19.22.2 to network 0.0.0.0

C       172.17.22.0 is directly connected, FastEthernet0/0
C       172.18.22.0  is directly connected, Serial0/0
S*      0.0.0.0/0 [1/0] via 172.19.22.2
```




***Şekil 6.11:*** *Yanlış yapılandırılmış default route.*

Bir şey bulabildiniz mi? Şekle ve routing tablosundaki direk bağlı route'lara bakarak, WAN linkinin, 172.18.22.0 network'ünde olduğunu ve default route'un tüm paketleri 172.19.22.0 network'üne gönderdiğini görebilirsiniz. Bu oldukça kötü, o, asla çalışmayacaktır. Problem, yanlış yapılandırılmış statik (default) route'dur.

Dinamik routing'e geçmeden önce son bir şey daha var. Şayet aşağıdaki gibi bir routing tablosu çıktısı varsa, router, 10.1.6.100 den 10.1.8.5 host'una hedeflenmiş bir paket alırsa ne olur?

```
Corp#sh ip route
[output cut]
Gateway of last resort is 10.1.5.5 to network 0.0.0.0

R     10.1.3.0 [120/1] via 10.1.2.2, 00:00:00, Serial 0/0
C     10.1.2.0  is directly connected, Serial0/0
C     10.1.5.0  is directly connected, Serial0/1
C     10.1.6.0  is directly connected, Fastethernet0/0
R*    0.0.0.0/0 [120/0] via 10.1.5.5, 00:00:00 Serial 0/1
```

Bu, şimdiye kadar yukarda gösterdiklerimden farklıdır. Çünkü default route, onun RIP-enjekte route olduğunu belirten `R*` olarak listelenmiştir. Bundan dolayı uzak bir router'da hem `ip default-route` komutunu kullanan hem de RIP yapılandıran birisi, RIP'in bu route'ı, ağ topluluğuna bir default route gibi yaymasına neden olur. Hedef adresi, 10.1.8.5 olduğundan ve 10.1.8.0 network'üne route olmadığından router, default route'u kullanacaktır ve paketi, serial0/1'e gönderecektir.

# Dinamik Routing

Dinamik routing, protokollerin, network'leri bulması ve router'lardaki routing tablolarının güncellenmesi için kullanılmaktadır. Doğru, bu, statik ve default routing kullanmaktan daha kolaydır, fakat bu size, network linklerindeki router CPU'su ve bant genişliği açısından pahalıya malolacaktır. Bir routing protokolü, komşu router'lar arasında routing bilgisini ilettiği zaman, bir router tarafından kullanılan kurallar bütününü tanımlamaktadır.

Bu modülde bahsedeceğim roting protokolleri, Routing Information Protocol (RIP) versiyon 1 ve 2 ile biraz da Interior Gateway Routing Protocol'dür (IGRP).

Ağ topluluklarında, iki tip routing protokolü kullanılmaktadır: Interior gateway protokolleri (IGP) ve exterior gateway protokolleri (EGP). IGP'ler, aynı autonomous system'de (AS)bulunan router'ların, routing bilgilerinin değiş tokuş edilmesi için kullanılmaktadır. Bir AS, ortak yönetilen bir domain'deki ağların topluluğudur. Bunun basit olarak anlamı şudur: aynı routing tablosunu

paylaşan tüm router'lar, aynı AS'dedirler. EGP'ler, AS'leri arasında iletişim için kullanılmaktadır. EGP'ye örnek, Border Gateway Protoco'dür l (BGP). (BGP, bu kitabın kapsamında değildir.)

Routing protokolleri, dinamik routing için çok gerekli olduğundan, ilerde bilmeniz gereken temel bilgileri vereceğim. Bu modülün ilerleyen bölümlerinde konfigürasyona odaklanacağım.

## Routing Protokol Temelleri

RIP'e detaylı olarak geçmeden önce, bilmeniz gereken önemli noktalar vardır. Özellikle, administrative distance, üç farklı routing protokolünü ve routing döngülerini anlamanız gerekmektedir. Bunların her birine aşağıdaki bölümlerde detaylı bakacağız.

### Administrative Distance

Administrative Distance(AD), komşu bir router'dan bir router'a alınan routing bilgisinin güvenirliğini derecelendirmek için kullanılır. Bir administrative distance, 0 ve 255 arası bir sayıdır. 0, en güvenilendir ve 255, bu route yoluyla trafiğin geçirilmeyeceği anlamına gelmektedir.

Şayet bir router, aynı uzak ağı listeleyen iki güncelleme alırsa, router'ın kontrol ettiği ilk şey, AD'dir. Advertised (yayınlanan) route'lardan birisi, diğerinde daha düşük AD'ye sahipse, en düşük AD'li route, routing tablosuna konacaktır.

Şayet, aynı ağ için iki advertised route, aynı AD'ye sahipse, uzak ağa en iyi yolu bulmak için, routing protokol metrikleri (hop count ya da hattın bant genişliği) kullanılacaktır. En düşük metriğe sahip advertised route, routing tablosuna konacaktır. Şayet iki advertised route, hem aynı AD'ye hem de aynı metriklere sahipse, o zaman routing protokolü, uzak network'e load-balance yapacaktır. (paketler her iki linkten de gönderilecektir).

Tablo 6.2, uzak bir ağa ulaşmak için hangi route'u kullanacağına karar vermek için bir Cisco router'ın kullandığı varsayılan administrative distance'ları göstermektedir.

**Tablo 6.2:** Varsayılan Administrative Distance'lar

| Route Kaynağı | Varsayılan Ad |
|---|---|
| Bağlı interface | 0 |
| Statik route | 1 |
| EIGRP | 90 |
| IGRP | 100 |
| OSPF | 110 |
| RIP | 120 |
| External EIGRP | 170 |
| Bilinmeyen | 255 (bu route asla kullanılmayacaktır.) |

Şayet bir network direk bağlıysa router, daima network'e bağlı interface'i kullanacaktır. Şayet bir statik route yapılandırırsanız, router bu route'un diğer öğrenilen route'lardan öncelikli olacağına inanacaktır. Statik route'ın administrative distance'ını değiştirebilirsiniz. Fakat varsayılan olarak AD'si, 1'dir. Statik route konfigürasyonlarımızda, her router'ın AD'sini 150 veya 151 olarak ayarladık. Bu bizim, routing protokollerini, statik route'ları kaldırmak zorunda kalmadan yapılandırmamıza izin verir. Onlar, routing protokollerinde bazı arızaların olması durumunda yedek route olarak kullanılacaktır.

Örnek olarak, aynı network'ü belirten bir statik route, RIP tarafından yayınlanan route ve IGRP tarafından yayınlanan route varsa, varsayılan olarak, router, (statik route'un AD'sini değiştirmediğiniz müddetçe) daima statik route'u kullanacaktır.

### Routing Protokolleri

Üç routing protokol sınıfı vardır:

**Distance vector:** Distance vector protokolleri, uzaklığı muhakeme ederek, uzak bir network'e en iyi yolu bulur. Bir paketin bir router'a uğradığı her sefer, hop olarak belirtilir. Network'e, en az sayıda hop ile giden route, en iyi route olarak kabul edilir. Vektör, uzak network'e yönü işaret eder. RIP ve IGRP, distance-vector protokolüdür. Bunlar, routing tablolarının tamamını, direk bağlı komşularına gönderirler.

**Link State:** shortest-path-first protokolleri olarak da bilinen link state protokollerinde, router'ların hepsi, üç ayrı tablo oluştururlar. Bu tablolardan birisi, direkt bağlı komşularının kayıtlarını tutar, diğeri, tüm ağ topluluğunun topolojisini oluşturur ve sonuncusu da, routing tablosu olarak kullanılır. Link-state router'lar, distance-vector routing tablosundan daha fazla bilgiye sahiptirler. OSPF, tamamıyla link-state olan bir IP routing protokolüdür. Link-state protokolleri, ağdaki diğer tüm router'lara linklerinin durumunu içeren güncellemeleri gönderirler.

**Hybrid:** Hybrid protokolleri, distance vector ve link state'in özelliklerini kullanmaktadır (örneğin EIGRP).

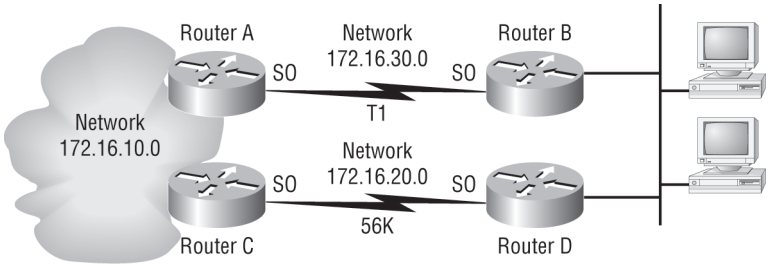
Her iş için, routing protokollerini yapılandırmanın belirlenmiş bir yolu yoktur. Bu, her bir durum için ayrı yapmanız gereken bir şeydir. Farklı routing protokollerinin nasıl çalıştığını bilirseniz, her işin kendi ihtiyaçlarına uygun iyi ve güvenilir kararlar verebilirsiniz.

## Distance-Vector Routing Protokolleri

Distance-vector routing algoritması, routing tablosunun içindekilerin tamamını komşu router'lara gönderir. Böylece, router'ların routing tablosunu tamamlamak için kendi routing tablolarıyla, alınan routing tablo kayıtları birleştirilir. Bu, kulaktan duyma routing olarak tanımlanır. Çünkü komşu router'dan güncelleme alan bir router, kendisi için olduğunu anlamaksızın uzak network'lerle ilgili bilgiye inanır.

Bir network'ün, aynı uzak ağa çok sayıda linke sahip olması mümkündür. Böyle olursa, ilk olarak, alınan her güncellemenin administrative distance'ı kontrol edilir. Şayet AD'ler aynı ise protokol, bu uzak ağa gitmek için kullanacağı en iyi yolu belirlemek için, diğer metrikleri kullanmak zorunda kalacaktır.

RIP, bir ağa giden en iyi yolu bulmak için sadece hop sayısını kullanır. Şayet RIP, aynı uzak ağa giden, aynı hop sayısında birden fazla link bulursa, otomatik olarak bir round-robin yük dengelemesi çalıştırır. RIP, altı eşit cost'a sahip linke kadar yük dengelemesi çalıştırabilir (varsayılan dört'tür).




***Şekil 6.12:*** *Pinhole tıkanıklığı.*

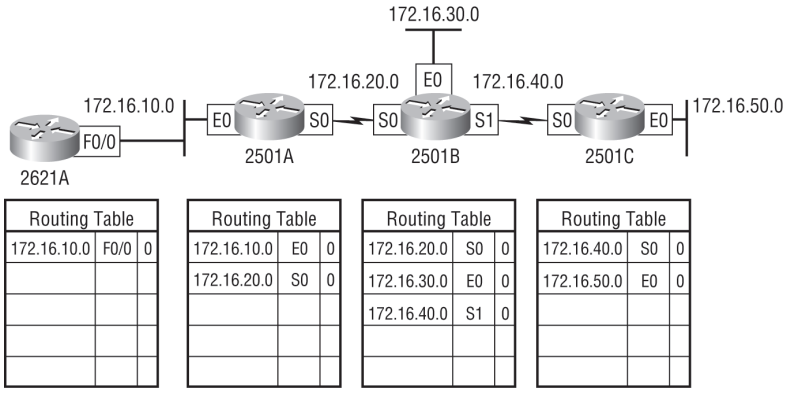
Bununla beraber, uzak network'e giden iki link, farklı bant genişliğine, fakat aynı hop sayısına sahipse, bu tip routing metrikleri problem oluşturur. Şekil 6.12, 172.16.10.0 uzak ağına, iki link göstermektedir.

172.16.30.0 network'ü, 1.544Mbps bant genişliğine sahip T1 linki ve 172.16.20.0, 56K bir link olduğundan router'ın 56K yerine T1 linkini seçmesini istersiniz, değil mi? Fakat hop sayısı, RIP

routing ile kullanılan tek metrik olduğundan, iki linkte, eşit cost değerli gibi görünür. Bu küçük problem, pinhole tıkanıklığı olarak tanımlanmaktadır.

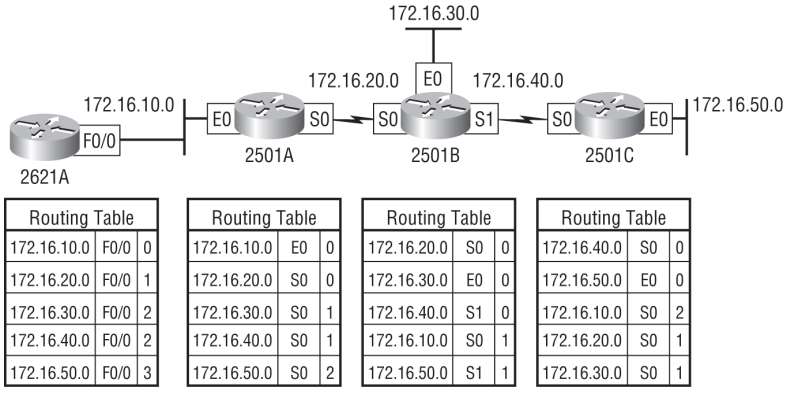
Çalışmaya başladığında bir distance-vector protokolünün ne yaptığını anlamak önemlidir. Şekil 6.13'te dört router, routing tablolarındaki direkt bağlı network'lerle çalışmaya başlar. Her router'da, distance-vector routing protokolü çalışmaya başladıktan sonra routing tabloları, komşu router'lardan toplanan route bilgileriyle güncellenir.

Şekil 6.13'te gösterildiği gibi her router, sadece direkt bağlı network'lere sahiptir. Her router, kendi routing tablosunun tamamını, aktif interface'lerine gönderir. Her router'ın routing tablosu, network numarasını, çıkış interface'ini ve network için hop sayısını içerir.




***Şekil 6.13:*** *Distance vector routing kullanan ağ topluluğu.*

Şekil 6.14'de, ağ topluluğundaki tüm network'ler hakkında bilgi içerdiği için routing tabloları tamamlanmıştır. Onlar, converged kabul edilirler. Router'lar, converge olurlarken, verilerin aktarılmaması mümkündür. Bu sebeple, hızlı convergence zamanı önemli bir artıdır. Aslında, RIP'le yaşanan problemlerden birisi, düşük convergence zamanı olmasıdır.



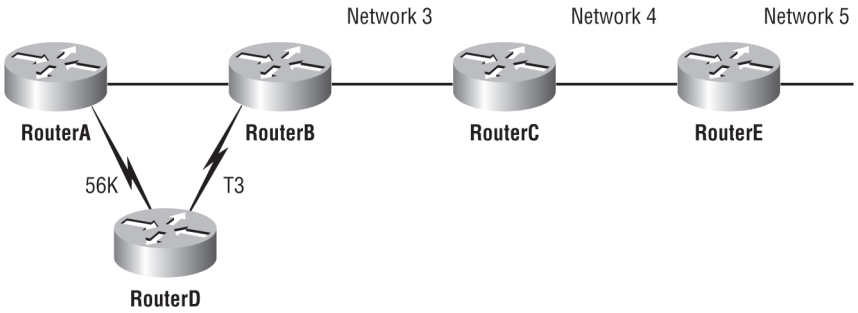

***Şekil 6.14:*** *Converged routing tabloları.*

Her router'daki routing tablosu, uzak network'ün adresi, router'ın network'e ulaşması için paketleri gönderdiği interface ve network için hop sayısı ya da metrik hakkındaki bilgileri tutar.

## Routing Kısır Döngüleri

Distance-vector routing protokolleri, aktif interface'lerinden periyodik routing güncellemelerini broadcast ederek, ağ topluluğundaki değişiklikleri takip eder. Bu broadcast, routing tablosunun tamamını içerir. Bu, oldukça iyi çalışır, fakat CPU prosesi ve linkin bant genişliği bakımından pahalıdır. Şayet network'te bir kesinti olursa, gerçekten sıkıntı verici problemler çıkabilir. Artı, distance-vector protokollerinin düşük convergence süresi, tutarsız routing tablolarına ve routing kısır döngülerine sebep olabilir.

Her router eşzamanlı, hatta yakın zamanda güncellenmezse routing kısır döngüleri olur. İşte bir örnek: Şekil 6.15'deki, Network 5'e bağlı interface'in arızalandığını düşünelim. Tüm router'lar Network 5 hakkındaki bilgiyi RouterE'den öğrenir. RouterA, tablosunda, Network 5'e RouterB üzerinden bir yola sahiptir.




***Şekil 6.15:*** *Routing kısır döngü örneği.*

Network 5 arızalandığında, RouterE, RouterC'ye söyler. Bu, RouterC'nin, RouterE üzerinden Network 5'e routing yapmasını durdurmasına sebep olur. Fakat RouterA, B ve D, henüz Network 5'in arızalandığını bilmezler. Bu nedenle, güncelleme bilgisi göndermeye devam ederler. RouterC sonunda güncellemesini gönderecektir ve RouterB, Network5'e routing yapmayı durduracaktır. Fakat RouterA ve D, hala güncel değildir. Onlara göre, Network 5, RouterB üzerinden 3 metriğiyle hala erişilebilir görünmektedir.

RouterA, kendi düzenli 30-saniye "Hello, ben hala buradayım" mesajını gönderir. Bunlar, hakkında bilgi sahibi oldukları linklerdir ve Network5'e ulaşabilmeyi de içermektedir. Şimdi, RouterB ve D, Network 5'e, RouterA'dan erişilebileceği haberini alırlar ve böylece, RouterB ve D, Network 5'in erişilebilir bilgisini gönderirler. Network 5 için hedeflenen her paket, RouterA'ya, RouterB'ye ve sonra tekrar RouterA'ya gidecektir. Bunun adı, routing kısır döngüsüdür. Bunu nasıl durdursunuz?

## Maximum Hop Count

Şimdi açıklanan routing kısır döngü problemi, counting to infinity (sonsuza saymak) olarak tanımlanır ve ona, dedikodu (broadcast) ile ağ topluluğu üzerinde iletilen ve yayılan yanlış bilgi sebep olur.

Bazı araya girme yöntemleri olmaksızın hop sayısı, bir paketin bir router'a uğradığı her sefer artarak devam eder.

Bu problemi çözmenin bir yolu, maximum hop count tanımlamaktır. RIP, 15'e kadar hop sayısına izin verir. Böylece, 16 hop, erişilemez kabul edilir. Diğer bir deyişle, 15 hop'lu bir döngüden sonra, Network5, arızalı kabul edilir. Böylece, maximum hop count, bir routing tablosu kaydının, geçersiz ya da şüpheli olması için ne kadar gideceğini kontrol eder.

## Split Horizon

Routing kısır döngü problemine diğer bir çözüm, split horizon olarak belirtilir. Bu, yanlış routing bilgisini ve (routing bilgisinin, alındığı yönden gönderilemeyeceği kuralını mecbur tutarak), distance vector network'te routing ek yükünü düşürür.

Diğer bir deyişle, routing protokolü bir network route'nun öğrenildiği interface'i tespit eder. Bu belirlenince route, aynı interface'den gönderilmeyecektir. Bu, RouterA'nın RouterB'den aldığı güncel bilgiyi RouterB'ye tekrar göndermesini engelleyecektir.

## Route Poisoning

Tutarsız güncellemelerin sebep olduğu problemlerden kaçınmanın ve network kısır döngülerini durdurmanın diğer yolu, route poisoning'tir. Örnek olarak, Network 5 gittiğinde, RouterE, Network

5'i 16.cı hop ya da erişilemez (bazen *sonsuz* olarak belirtilir) olarak yayınlayarak, route poisoning başlatır.

Network 5'e giden route'ın zehirlenmesi RouterC'nin, Network 5'e giden route hakkında yanlış güncellemelerden etkilenmesini sağlar. RouterC, RouterE'den bir route poisoning aldığında, *poison reverse* olarak belirtilen bir güncellemeyi tekrar RouterE'ye gönderir. Bu, segment'teki tüm router'ların zehirli route bilgisi almalarını kesinleştirir.

### Holddown

Bir holddown, düzenli güncelleme bilgilerinin, up ve down olan (flapping denir) bir route'da tekrarlanmasını engeller. Genellikle, bağlantısı kaybolup tekrar gelen seri linklerde olur. Bunu tespit etmek için bir yol yoksa network asla converge olmayacak ve bu gidip gelen interface, tüm network'ün down olmasına sebep olacaktır.

Holddown'lar, sonraki en iyi route ile değiştirmeden önce, down olan route'un tekrar up olması ya da network'ün dengelenmesi için bir süre geçmesine izin vererek, route'ların çok hızlı değişmesini engellerler. Bu, router'lara, belirli bir zaman dilimi için, yakınlarda kaldırılan route'ların etkilenebileceği değişiklikleri kısıtlamasını da söyler. Bu, çalışmayan route'ların, diğer router tablolarına vaktinden önce konmasını engeller.

## Routing Information Protocol (RIP)

Routing Information Protocol (RIP), gerçek bir distance-vector routing protokolüdür. RIP, routing tablosunun tamamını her 30 saniyede tüm aktif interface'lerine gönderir. RIP, uzak bir ağa en iyi yolu belirlemek için sadece hop sayısını kullanır. Fakat varsayılan olarak, maksimum 15 kabul edilebilir hop sayısına sahiptir. Yani, 16 erişilemez kabul edilmektedir. RIP, küçük network'lerde iyi çalışır, fakat düşük WAN linklerine sahip, geniş ağlarda ve çok sayıda router'ın kurulduğu network'lerde yetersizdir.

RIP versiyon1, sadece classfull routing kullanırlar. Yani, network'teki tüm cihazlar, aynı subnet maskını kullanmak zorundadır. Bundan dolayı, RIP versiyon1, beraberinde subnet mask içeren güncelleme göndermez. RIP versiyon2, prefix routing sağlar ve route güncellemeleriyle subnet masklarını gönderir. Bu, classless routing olarak belirtilir.

Aşağıdaki bölümlerde, RIP timer'ları ve sonra da RIP yapılandırmasını tartışacağız.

### RIP Timer'ları

RIP, performansını düzenlemek için, dört farklı timer kullanır:

**Route update timer:** Router'ın tüm komşularına, routing tablosunun tam kopyasını gönderdiği periyodik routing güncellemeleri arasındaki zaman aralığını ayarlar (genelde 30 saniyedir).

**Route invalid timer:** Bir router'ın, bir route'u geçersiz kabul etmesinden önce geçmesi gereken zamanın uzunluğunu belirler (180 saniye). Bu sonuca, bu periyotta belirli bir route hakkında herhangi bir güncelleme almayarak ulaşır. Bu olduğunda router tüm komşularına, route'un geçersiz olduğunu bilmelerini sağlayan güncellemeleri gönderecektir.

**Holddown timer:** Routing bilgilerinin kesildiği sıradaki zaman miktarını ayarlar. Route'un erişilemez olduğunu belirten bir güncelleme paketi alındığında, router'lar, holddown durumuna geçer. Bu, daha iyi metriğe sahip bir update paketi alınana ya da holddown timer süresi dolana kadar devam eder. Varsayılanda 180 saniyedir.

**Route flush timer:** Bir route'un geçersiz olması ile onun routing tablosundan çıkartılması arasındaki zamanı ayarlar (240 saniye). Tablodan çıkartılmadan önce, router, bu router'ın ölmesinin yakın olduğunu komşularına duyurur. Route invalid timer değerinin, route flush timer dan daha küçük olması gerekir. Bu router'a, lokal routing tablosu güncellenmeden önce, geçersiz route hakkında komşularının bilgilendirilmesi için yeterli zamanı sağlar.

## RIP Routing Konfigürasyonu

RIP routing konfigürasyonu için, `router rip` komutu ile protokolü açın ve RIP routing protokolüne, hangi network'lerin yayınlanacağını söyleyin. Hepsi bu kadar. Şimdi, bizim 5-router'lu ağ topluluğumuzu (Şekil 6.9), RIP routing ile yapılandıralım.

### Corp

RIP, 120 administartive distance'a sahiptir. Statik route'lar, varsayılanda 1 AD'ye sahiptir ve biz yapılandırılmış statik route'lara sahip olduğumuzdan, routing tabloları, RIP bilgileri ile dağıtılmayacaktır. Bununla beraber, her statik route'un sonuna 150/151'i eklediğimden, RIP'i kullanabiliriz.

RIP routing protokolünü, router rip ve network komutlarını kullanarak ekleyebilirsiniz. Network komutu, routing protokolüne, hangi classful network'ün yayınlanacağını söyler.

Corp router konfigürasyonuna bakın ve bunun ne kadar kolay olduğunu anlayın:

```
Corp#config t
Corp(config)#router rip
Corp(config-router)#network 10.0.0.0
```

Bu kadar. İki veya üç komut ve tamamladınız. Statik route kullanmaktan daha kolay olduğu kesin, değil mi? Yine de, fazladan router CPU prosesi ve bant genişliği kullandığınız aklınızda olsun.

Subnet'leri yazmadığıma dikkat edin. Sadece classful network adresleri vardır.(subnet ve host bit'lerinin hiçbiri kullanılmaz). Subnet'leri bulmak ve routing tablolarını yerleştirmek, routing protokolünün görevidir. RIP çalışan bir router'ımız olmadığından, routing tablosunda henüz bir RIP güncellemesi göremiyoruz.

NOT

*Network adresleri yapılandırıldığında, RIP'in, classful adres kullandığını hatırlayın. Bundan dolayı, network'teki tüm cihazlardaki subnet mask aynı olmalıdır (bu classful routing olarak tanımlanır). Bunu açıklamak için 172.16.10.0, 172.16.20.0 ve 172.16.30.0 subnet'leri ile 172.16.0.0/24 ClassB network adresi kullandığınızı farz edelim. Sadece 172.16.0.0 classful network adresini yazın, RIP'in subnet'leri bulmasını ve onları routing tablosuna yerleştirmesini sağlayabilirsiniz.*

### R1

R1 router'ımızı yapılandıralım:

```
R1#config t
R1(config)#router rip
R1(config-router)#network 10.0.0.0
R1(config-router)#do show ip route
     10.0.0.0/24 is subnetted, 12 subnets
S       10.1.11.0 [150/0] via 10.1.3.1
S       10.1.10.0 [150/0] via 10.1.3.1
S       10.1.9.0 [150/0] via 10.1.3.1
S       10.1.8.0 [150/0] via 10.1.3.1
S       10.1.12.0 [150/0] via 10.1.3.1
C       10.1.3.0 is directly connected, Serial0/0/1
C       10.1.2.0 is directly connected, Serial0/0/0
R       10.1.1.0 [120/1] via 10.1.3.1, 00:00:04, Serial0/0/1
                 [120/1] via 10.1.2.1, 00:00:04, Serial0/0/0
C       10.1.7.0 is directly connected, FastEthernet0/1
C       10.1.6.0 is directly connected, FastEthernet0/0
R       10.1.5.0 [120/1] via 10.1.3.1, 00:00:04, Serial0/0/1
```

```
                [120/1] via 10.1.2.1, 00:00:04, Serial0/0/0
R       10.1.4.0 [120/1] via 10.1.3.1, 00:00:09, Serial0/0/1
                [120/1] via 10.1.2.1, 00:00:09, Serial0/0/0
R1(config-router)#
```

Bu oldukça basittir. Bu routing tablosundan biraz bahsedelim. Routing tablomuzu değiş tokuş ettiğimiz bir tane RIP komşumuz olduğundan, Corp router'ından gelen RIP ağlarını görebiliriz. (Diğer tüm route'lar, hala statik görünmektedir.) RIP, Corp router'ına her iki bağlantıyı da bulacak ve onlar arasında yük-paylaşımı yapacaktır.

## R2

R2 router'ımızı RIP ile yapılandıralım:

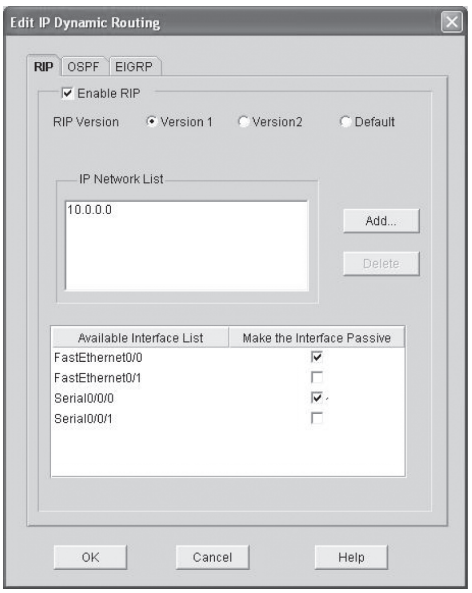
```
R2#config t
R2(config)#router rip
R2(config-router)#network 10.0.0.0
R2(config-router)#do show ip route
     10.0.0.0/24 is subnetted, 12 subnets
S       10.1.11.0 [150/0] via 10.1.4.1
S       10.1.10.0 [150/0] via 10.1.4.1
C       10.1.9.0 is directly connected, FastEthernet0/0
C       10.1.8.0 is directly connected, Dot11Radio0/3/0
S       10.1.12.0 [150/0] via 10.1.4.1
R       10.1.3.0 [120/1] via 10.1.4.1, 00:00:03, Serial0/2/0
R       10.1.2.0 [120/1] via 10.1.4.1, 00:00:03, Serial0/2/0
R       10.1.1.0 [120/1] via 10.1.4.1, 00:00:03, Serial0/2/0
R       10.1.7.0 [120/2] via 10.1.4.1, 00:00:03, Serial0/2/0
R       10.1.6.0 [120/2] via 10.1.4.1, 00:00:03, Serial0/2/0
R       10.1.5.0 [120/1] via 10.1.4.1, 00:00:03, Serial0/2/0
```

RIP komşularını ekledikçe, routing tablosu R'lerle büyümektedir. Routing tablosundaki route'lardan bazılarının hala statik olduğunu görebiliyoruz. Tamamlamak için iki router kaldı.

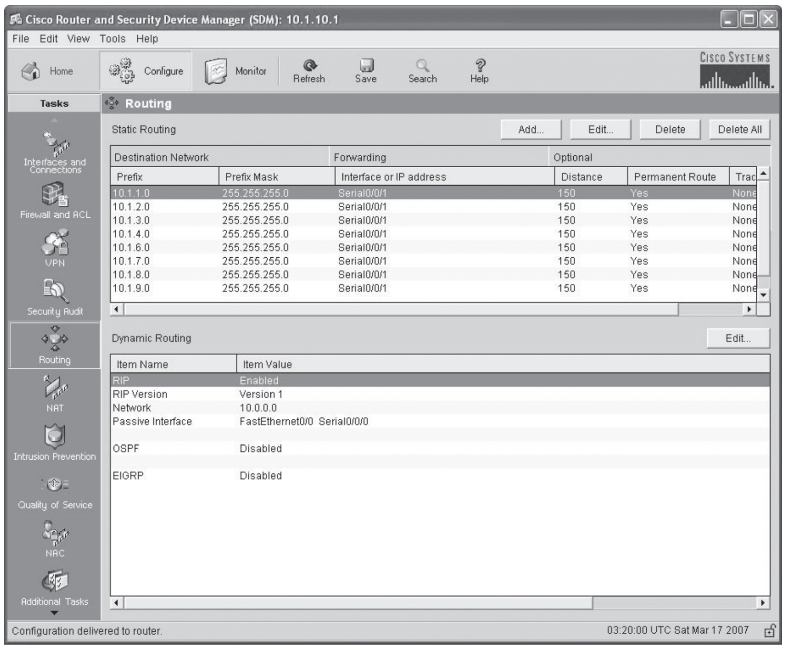
## R3

R3 router'ımızı, RIP ile yapılandıralım, alışıldığı gibi R3'te SDM kullanacağız.

Routing ekranından, Dynamic Routing'in sağındaki Edit butonuna tıkladım. Şimdi, RIP'i ve network adreslerini yapılandırabiliyorum. RIP'in broadcast edilmesini istemediğim interface'lere tıkladım. RIP'in broadcast edileceği interface'lere işaret konulmadı.



Bunlar, pasif interface olarak tanımlanır ve kısa bir süre sonra bahsedeceğim. Router'ların olmayacağı bir interface'den RIP broadcast'i göndermenin anlamı yoktur.




SDM ekranından, R3 konfigürasyonunu tamamladığımızı görebiliriz.

## 871W

Son router'ın RIP konfigürasyonu şöyledir:

```
871W#config t
871W(config)#no ip route 0.0.0.0 0.0.0.0 10.1.11.1
871W(config)#router rip
871W(config-router)#network 10.0.0.0
871W(config-router)#do sh ip route
     10.0.0.0/24 is subnetted, 12 subnets
C       10.1.11.0 is directly connected, Vlan1
R       10.1.10.0 [120/1] via 10.1.11.1, 00:00:23, Vlan1
R       10.1.9.0 [120/3] via 10.1.11.1, 00:00:23, Vlan1
R       10.1.8.0 [120/3] via 10.1.11.1, 00:00:23, Vlan1
C       10.1.12.0 is directly connected, Dot11Radio0
R       10.1.3.0 [120/2] via 10.1.11.1, 00:00:23, Vlan1
R       10.1.2.0 [120/2] via 10.1.11.1, 00:00:23, Vlan1
R       10.1.1.0 [120/2] via 10.1.11.1, 00:00:23, Vlan1
R       10.1.7.0 [120/3] via 10.1.11.1, 00:00:24, Vlan1
R       10.1.6.0 [120/3] via 10.1.11.1, 00:00:24, Vlan1
R       10.1.5.0 [120/1] via 10.1.11.1, 00:00:24, Vlan1
R       10.1.4.0 [120/2] via 10.1.11.1, 00:00:24, Vlan1
871W#
```

Son olarak, routing tablosunda görünen tüm route'lar, RIP-enjekte route'lardır.

Administrative distance'ları ve RIP routing'i eklemeden veya onları bizim yaptığımız gibi 120'den yüksek yapmadan önce, statik route'ları çıkartmamız gerektiğini hatırlamak önemlidir.

Varsayılan olarak, direk bağlı route'lar 0 AD'ye, statik route'lar, 1AD'ye ve RIP, 120 AD'ye sahiptir. Bir söylenti (advertised route) duyunca, ayırım yapmaksızın doğru olduğu kabul eden RIP'i, dedikodu protokolü olarak belirteceğim. Bu, RIP'in bir ağ topluluğunda nasıl davrandığını özetler. Dedikodu yayma prosesi gibi protokol!

## RIP Routing Tablolarının Doğruluğunu Kontrol Etmek

Her routing tablosu şimdi, hem komşu router'lardan alınan, RIP-enjekte route'lara hem de tüm direk bağlı route'lara sahip olmalıdır.

Bu çıktı, Corp routing tablosunun içindekileri göstermektedir:

```
     10.0.0.0/24 is subnetted, 12 subnets
R       10.1.11.0 [120/1] via 10.1.5.2, 00:00:28, Serial0/2/0
R       10.1.10.0 [120/1] via 10.1.5.2, 00:00:28, Serial0/2/0
R       10.1.9.0 [120/1] via 10.1.4.2, 00:00:26, Serial0/1/0
R       10.1.8.0 [120/1] via 10.1.4.2, 00:00:26, Serial0/1/0
R       10.1.12.0 [120/2] via 10.1.5.2, 00:00:28, Serial0/2/0
C       10.1.3.0 is directly connected, Serial0/0/1
C       10.1.2.0 is directly connected, Serial0/0/0
C       10.1.1.0 is directly connected, FastEthernet0/1
R       10.1.7.0 [120/1] via 10.1.3.2, 00:00:07, Serial0/0/1
                 [120/1] via 10.1.2.2, 00:00:10, Serial0/0/0
R       10.1.6.0 [120/1] via 10.1.3.2, 00:00:07, Serial0/0/1
                 [120/1] via 10.1.2.2, 00:00:10, Serial0/0/0
C       10.1.5.0 is directly connected, Serial0/2/0
C       10.1.4.0 is directly connected, Serial0/1/0
```

Bu çıktı bize, routing tablosunun **R** hariç, statik route'lar kullandığımızla aynı kayıtları içerdiğini gösterir. R, network'lerin RIP routing protokolü kullanarak dinamik eklendiklerini belirtmektedir. [120/1], uzak ağa hop sayısı (1) ile beraber, route'un administrative distance'ını (120) göstermektedir. Corp router'ından, iki hop uzakta olan 10.1.12.0 hariç, diğer network'ler bir hop uzaklıktadır.

RIP'ın küçük ağ topluluğumuzda çalıştığı doğru, fakat her kuruluş için çözüm değildir. Bu teknik, 15 maksimum hop sayısına sahiptir (16, ulaşılamaz kabul edilir). Artı, her 20 saniyede komple routing-tablosu güncellemesi çalıştırır ki bu geniş bir ağ topluluğunu kısa sürede ağırlaştırır.

RIP routing tabloları ve uzak network'lere yayınlamak için kullanılan parametreler hakkında göstermek istediğim bir şey daha var. Örnekteki gibi, aşağıdaki routing tablosu, 10.1.3.0 network metriğinde [120/15] gösterir. Bunun anlamı; administrative distance, 120 (RIP için varsayılandır), hop sayısı 15'tir. Router'ın, bir komşusuna güncelleme gönderdiği her sefer, her router için hop sayısını bir arttırdığını hatırlayın.

```
R3#sh ip route
     10.0.0.0/24 is subnetted, 12 subnets
C       10.1.11.0 is directly connected, FastEthernet0/1
C       10.1.10.0 is directly connected, FastEthernet0/0
R       10.1.9.0 [120/2] via 10.1.5.1, 00:00:15, Serial0/0/1
R       10.1.8.0 [120/2] via 10.1.5.1, 00:00:15, Serial0/0/1
R       10.1.12.0 [120/1] via 10.1.11.2, 00:00:00, FastEthernet0/1
R       10.1.3.0 [120/15] via 10.1.5.1, 00:00:15, Serial0/0/1
```

```
R       10.1.2.0 [120/1] via 10.1.5.1, 00:00:15, Serial0/0/1
R       10.1.1.0 [120/1] via 10.1.5.1, 00:00:15, Serial0/0/1
R       10.1.7.0 [120/2] via 10.1.5.1, 00:00:15, Serial0/0/1
R       10.1.6.0 [120/2] via 10.1.5.1, 00:00:15, Serial0/0/1
C       10.1.5.0 is directly connected, Serial0/0/1
R       10.1.4.0 [120/1] via 10.1.5.1, 00:00:15, Serial0/0/1
R3#
```

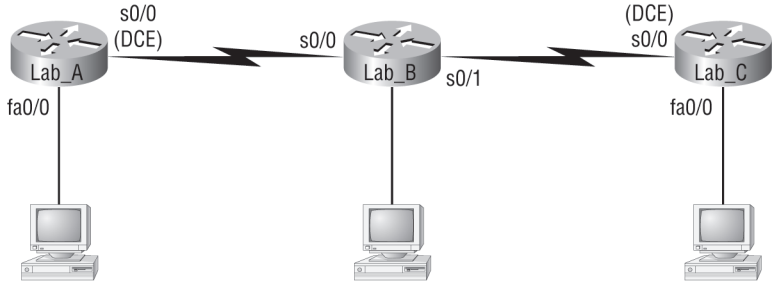
Bundan dolayı, bu [120/15] oldukça kötü sayılır, çünkü R3 router'ından tabloyu alan sonraki router, hop sayısı 16 olacağından, network 10.1.3.0'a olan route'u atacaktır.

*Şayet bir router, bir network'e zaten kendi routing tablosunda olandan daha yüksek bir cost değeri içeren bir routing güncellemesi alırsa, güncelleme göz ardı edilecektir.*

NOT

## RIP Routing Konfigürasyon Örneği 2

RIP konfigürasyonları hakkında daha fazla detay öğrenmeden önce, Şekil 6.16'ya bir bakalım. Bu örnekte, ilk olarak subnet'lerimizi bulup oluşturacağız ve sonra router'a RIP konfigürasyonunu ekleyeceğiz.




***Şekil 6.16:*** *RIP routing örneği 2.*

Bu konfigürasyon için, Lab_B ve Lab_A'nın zaten yapılandırıldığını düşüneceğiz ve sadece Lab_A router'ını yapılandıracağız. 192.168.164.0/28 network ID'sini kullanacağız. Lab_A'nın s0/0 interface'i, altıncı subnet'teki son geçerli IP adresini ve f0/0 interface'i, ikinci subnet'teki son IP adresini kullanacaktır. Zero subnet'ini, geçerli olarak düşünmeyin.

Başlamadan önce, /28'in, 255.255.255.240 olduğunu biliyorsunuz, değil mi? Ve dördüncü oktette,16 blok boyutuna sahibiz. Bunları bilmeniz çok önemlidir. Yoksa bölüm 2 ve 3'ü tekrar gözden geçirmelisiniz. Subnet'lemeyi gözden geçirmek size zarar vermez.

16 blok boyutuna sahip olduğumuz için, subnet'lerimiz, 16, 32, 48, 64, 80, 96, 112, 128, 144, vs.dir (bu örnek için, 0 subnet'ini kullanmıyoruz). Sekizinci subnet (s0/0 interface'i için kullanacağımız) 128'dir. 128 subnet'i için geçerli host aralığı 129-142 dir ve 128 subnet'inin broadcast adresi, 143'tür. İkinci subnet (f0/0 interface'i için kullanacağımız) 32 subnet'idir. Tanımlı host'lar, 33-46 arasıdır ve 47, 32 subnet'inin broadcast adresidir.

Aşağıda, Lab_A router'ındaki konfigürasyonumuzun nasıl göründüğünü bulabilirsiniz:

```
Lab_A(config)#interface s0/0
Lab_A(config-if)#ip address 192.168.164.142 255.255.255.240
Lab_A(config-if)#no shutdown
Lab_A(config-if)#interface fa0/0
Lab_A(config-if)#ip address 192.168.164.46 255.255.255.240
Lab_A(config-if)#no shutdown
Lab_A(config-if)#router rip
Lab_A(config-router)#network 192.168.164.0
```

```
Lab_A(config-router)#^Z
Lab_A#
```

Subnet'leri hesaplamak ve son geçerli host'u yapılandırmak oldukça basit olmalı. Şayet değilse, bölüm 3'e geri dönün. Bununla beraber, aslında dikkat etmenizi istediğim şey, Lab_A router'ına iki subnet eklediğimiz halde, RIP'te sadece tek ifade olmasıdır. Bazen, sadece classful (yani host bit'leri kullanılmaz) network komutunu yapılandırdığınızı hatırlamak zordur.

Bu, ikinci RIP konfigürasyon örneğinin gerçek amacıdır. Yani, classful network adreslemesini size hatırlatmak. Ayrıca, subnet'leme örneği yapmanın zararı olmaz, değil mi?

## RIP Yayınlarını Göndermek

Muhtemelen RIP network'lerinizin, LAN ve WAN'da her yere yayınlanmasını istemezsiniz. RIP network'ünüzü internete yayınlayarak kazanabileceğiniz bir şey yok, değil mi?

İstenmeyen RIP güncellemelerinin, LAN ve WAN'larınızdan yayılmasını durdurmanın birkaç yolu vardır. En kolay olanı, R3 konfigürasyonu sırasında gösterdiğim, passive-interface komutudur. Bu komut, RIP güncelleme broadcast'lerinin, belirli bir interface'den gönderilmesini engeller. Aynı interface hala RIP güncellemesi alacaktır.

Aşağıda, bir router'da CLI kullanarak passive-interface komutunun nasıl yapılandırılacağı gösterilmektedir:

```
Lab_A#config t
Lab_A(config)#router rip
Lab_A(config-router)#network 192.168.10.0
Lab_A(config-router)#passive-interface serial 0/0
```

Bu komut, serial 0/0 interface'inden RIP güncellemelerinin gönderilmesini durduracaktır. Fakat serial 0/0 interface'i, hala RIP güncellemesi alacaktır. Bu, R3 router'unda gösterdiğim gibi, SDM kullanılarak ta kolayca yapılabilir.

## RIP versiyon2 (RIPv2)

Cisco tescilli distance vector routing protokolü IGRP'ye geçmeden önce, RIPv2 için biraz zaman harcayalım.

RIP versiyon2, çoğunlukla RIP versiyon1 ile aynıdır. RIPv1 ve RIPv2, distance vector protokolüdür. Yani, RIP çalışan her router, periyodik zaman aralıklarında, routing tablolarının tamamını, tüm aktif interface'lerinden gönderecektir. Ayrıca, timer'lar ve kısır döngü-önleme yöntemleri, her iki RIP versiyonunda da aynıdır. (holddown timer ve split horizon kuralı). RIPv1 ve RIPv2, classful adresleme ile yapılandırılır (fakat RIPv2, subnet bilgisi her route güncellemesinde gönderildiğinden, classless olarak kabul edilir) ve her ikisinin administrative distance'ı aynıdır (120).

RIPv2'yi RIPv1'den daha ölçeklenebilir yapan, bazı önemli farklılıklar vardır. Devam etmeden önce, bir tavsiyede bulunacağım: Kesinlikle ağınızda RIP'in her iki versiyonunun da kullanılmasını savunmuyorum. Fakat RIP açık bir standarttır, RIP'i herhangi bir marka router ile kullanabilirsiniz. OSPF'te açık standart olduğundan, OSPF'de kullanabilirsiniz (Modül7'de bahsedilecek). RIP, çok fazla bant genişliği gerektirir ve network'ünüzün yoğun çalışmasına sebep olur. Daha üstün seçenekleriniz olduğunda, neden seçmeyesiniz?

## Gerçek Dünya Senaryosu

### *Bir Ağ Topluluğunda RIP Kullanmalı mıyız?*

*Büyüyen bir network'e birkaç tane Cisco router eklemek için danışman olarak işe alındınız. Network'te tutmak istedikleri birkaç tane Unix router'ları vardır. Bu router'lar, RIP haricindeki routing protokollerini desteklememektedir. Ve tüm network'te sadece RIP çalıştırmak zorundasınız.*

*Eski network'lere bağlı bir router'da RIP çalıştırabilirsiniz. Fakat tabi ki, tüm ağ topluluğunda RIP çalıştırmanıza gerek yoktur.*

*Basitçe bir tür routing protokolünü diğerine çeviren, redistribution'ı kullanabilirsiniz. Yani, eski router'ları, RIP kullanarak destekleyebilir, ağınızın geri kalanında, örneğin EIGRP kullanabilirsiniz.*

*Bu, RIP güncellemelerinin, tüm ağ topluluğuna gönderilmesini ve değerli bant genişliğinden harcamasını engelleyecektir.*

**Tablo 6.3:** RIPv1 ve RIPv2 Arasındaki Farklılıkları

| RIPv1 | RIPv2 |
|---|---|
| Distance vector | Distance vector |
| Maksimum hop sayısı 15 | Maximum hop sayısı 15 |
| Classful | Classless |
| Broadcast tabanlı | Multicast 224.0.0.9 kullanır. |
| VLSM'i desteklemez. | VLSM network'leri destekler. |
| Kimlik denetimi yok | MD5 authentication sağlar |
| Discontiguous network'leri desteklemez | Discontiguous network'leri destekler |

RIPv2, RIPv1'in aksine, classless bir routing protokolüdür (RIPv1 gibi classful olarak yapılandırılsa da). Yani, route güncellemeleri ile beraber, subnet mask bilgisi gönderilir. Güncellemelerle subnet mask bilgisini göndererek, RIPv2, hem Variable Length Subnet Mask'ları (VLSM) hem de network summarization'ı destekler. İlave olarak, RIPv2, discontiguous ağ kurulumunu destekler (Modül7'de daha çok bahsedeceğim).

RIPv2 konfigürasyonu, oldukça basittir. İşte bir örnek:

```
Lab_C(config)#router rip
Lab_C(config-router)#network 192.168.40.0
Lab_C(config-router)#network 192.168.50.0
Lab_C(config-router)#version 2
```

*RIPv2, classless'dır ve VLSM ile discontiguous network'lerde çalışır.*

NOT

Hepsi bu, (config-router)# istemcisinde `version 2` komutunu ekleyin, artık RIPv2 çalışıyorsunuz.

# Interior Gateway Routing Protocol (IGRP)

Interior Gateway Routing Protocol (IGRP), Cisco tescilli bir distance vector routing protokolüdür. Yani, ağınızda IGRP kullanmak için tüm router'larınızın Cisco router olması gerekir. Cisco, bu routing protokolünü RIP ile ilgili problemlerin üstesinden gelmek için geliştirdi.

IGRP, varsayılanda 100 (EIGRP ile aynı) olan 255 maksimum hop sayısına sahiptir. Bu, geniş ağlarda kullanışlıdır ve RIP network'lerindeki maksimum 15 hop problemini çözümler.

IGRP, RIP'ten farklı bir metrik kullanır. IGRP, bir ağ topluluğuna en iyi route'u belirlemek için varsayılan olarak hattın bant genişliğine ve gecikmesine (delay) bakar. Bu, composite metrik olarak belirtilir. Varsayılan olarak kullanılmasalar da, güvenilirlik (reliability), yük (load) ve maximum transmission unit'de (MTU) kullanılmaktadır.

NOT *RIP ve IGRP konfigürasyonları arasındaki ana farklılık, IGRP yapılandırdığınızda, autonomous system numarası kullanılmasıdır. Tüm router'lar, routing tablosu bilgisini paylaşmak için aynı numarayı kullanmak zorundadır.*

Tablo 6.4, RIP'te bulamayacağınız IGRP özelliklerini göstermektedir.

**Tablo 6.4:** IGRP ile RIP Karşılaştırması

| IGRP | RIP |
|---|---|
| Büyük ağ topluluklarında kullanılabilir. | En iyi küçük ağlarda çalışmasıdır. |
| Aktivasyon için bir autonomous sistem numarası kullanır. | Autonomous sistem numarası kullanmaz. |
| Her 90 saniyede komple route güncellemesi gönderir. | Her 30 saniyede komple route güncellemesi gönderir. |
| administrative distance, 100'dür. | administrative distance, 120'dir . |
| Metrik olarak, maksimum 255 hop sayısı ile hattın, bant genişliğini ve gecikmesini kullanır. | Uzak bir ağa en iyi yol için, maksimum 15 hop ile sadece hop sayısını kullanır. |

IGRP konusu neden burada sona ermektedir? Router'ımda IGRP yapılandırmaya çalıştığımda ne olduğunu izleyin:

```
R3#config t
Enter configuration commands, one per line.  End with CNTL/Z.
R3(config)#router igrp 10
                   ^
% Invalid input detected at '^' marker.
R3(config)#
```

İşte sebebi budur. Cisco, IGRP'yi artık desteklememektedir. Neden desteklesin ki? Tüm yapmanız gereken, IGRP'nin önüne bir ***E*** koymaktır. Böylece, çok daha iyi bir routing protokolü çalıştırmış oluyorsunuz. EIGRP'yi bir sonraki bölümde işleyeceğiz. Şimdi, RIP için bazı doğrulama komutlarına bakalım.

# Konfigürasyonlarınızın Doğruluğunu Kontrol Etmek

Tamamlayınca ya da tamamladığınızı düşündüğünüzde konfigürasyonlarınızın doğruluğunu kontrol etmek önemlidir. Aşağıdaki liste, Cisco router'larınızda yapılandırılan routed ve routing protokollerin doğrulanması için kullanabileceğiniz komutlar vardır:

- `show ip route`
- `show ip protocols`
- `debug ip rip`

İlk komutu önceki bölümde görmüştünüz. Aşağıdaki bölümlerde diğerlerinden bahsedeceğim.

## show ip protocols Komutu

`show ip protocols` komutu, router'ınızda yapılandırılan routing protokollerini göstermektedir. Aşağıdaki çıktıya bakarak, router'ınızda RIP çalıştığını ve RIP'in kullandığı timer'ları görebilirsiniz:

```
R3#sh ip protocols
Routing Protocol is "rip"
```

```
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Sending updates every 30 seconds, next due in 24 seconds
  Invalid after 180 seconds, hold down 180, flushed after 240
  Redistributing: rip
  Default version control: send version 1, receive version 1
    Interface             Send  Recv  Triggered RIP  Key-chain
    FastEthernet0/1       1     1

    Serial0/0/1           1     1

  Automatic network summarization is not in effect
  Maximum path: 4
  Routing for Networks:
    10.0.0.0
  Passive Interface(s):
    FastEthernet0/0
    Serial0/0/0
  Routing Information Sources:
    Gateway         Distance      Last Update
    10.1.11.2            120      00:00:10
    10.1.5.1             120      00:00:22
  Distance: (default is 120)
```

Bu çıktıdaki RIP'in varsayılan olarak her 30 saniyede güncelleme gönderdiğine dikkat edin. Distance vector'de kullanılan timer'lar da görünmektedir.

Daha aşağılarda RIP'in, direkt bağlı f0/1 ve s0/0/0 interface'leri için routing yaptığını görebilirsiniz. Versiyon, RIPv1 olarak, interface'lerin sağında listelenmektedir.

F0/0 ve s0/0/0, passive interface olarak listelenmiştir (RIP bilgisi göndermeyeceklerdir). Bulunan komşular, 10.1.11.2 ve 10.1.5.1'dir. Son kayıt, RIP'in varsayılan AD'sidir (120).

### show ip protocols ile Hata Tespiti

Şimdi, basit bir router kullanalım ve başka network'teki bir router'dan bu çıktıya bakarak routing hakkında neleri belirleyebileceğimizi görmek için `show ip protocols` komutunu kullanalım:

```
Router#sh ip protocols
Routing Protocol is "rip"
  Sending updates every 30 seconds, next due in 6 seconds
  Invalid after 180 seconds, hold down 180, flushed after240
  Outgoing update filter list for all interfaces is
  Incoming update filter list for all interfaces is
  Redistributing: rip
  Default version control: send version 1, receive any version
    Interface        Send  Recv   Key-chain
    Serial0/0         1     1 2
    Serial0/1         1     1 2
```

```
  Routing for Networks:
    10.0.0.0
  Routing Information Sources:
    Gateway         Distance      Last Update
    10.168.11.14         120      00:00:21
  Distance: (default is 120)
```

**Ayrıca, aynı router'dan** `show ip interface brief` **komutuna bakalım ve ne öğreneceğimizi anlayalım:**

```
Router#sh ip interface brief
Interface IP-Address      OK?    Method Status
FastEthernet0/0  192.168.18.1    YES    manual  up
Serial0/0        10.168.11.17    YES    manual  up
FastEthernet0/1  unassigned      YES    NRAM    Administatively
down
Serial0/1        192.168.11.21   YES    manual  up
```

`show ip protocols` çıktısında, 10.0.0.0 network'ü için RIP routing kullandığımızı görebilirsiniz. Yani, konfigürasyonumuz şöyle olacaktır:

```
Router(config)#router rip
Router(config-router)#network 10.0.0.0
```

Ayrıca, sadece serial0/0 ve serial0/1, RIP network'ünün bir parçasıdır. Son olarak, komşu router'umuz, 10.168.11.14'tür.

Show `ip interface brief` komutu çıktısından, sadece serial0/0'ın, 10.0.0.0 network'ünde olduğunu görebilirsiniz. Yani, router, sadece 10.0.0.0 network'ü ile routing güncellemeleri gönderip alacaktır ve 192.168.0.0 network'lerini herhangi bir interface'inden yayınlamayacaktır.

## debug ip rip Komutu

***debug ip rip*** komutu, routing güncellemelerini, router'da gönderilip alınıyorlarmış gibi, konsol oturumuna gönderirler. Şayet router'a telnet yaparsanız, `debug` komutundan çıktı alabilmesi için ***terminal monitor*** komutunu kullanmalısınız.

Bu çıktıda, RIP'in hem gönderdiğini hem de aldığını görebiliriz (metrik, hop sayısıdır):

```
R3#debug ip rip
RIP protocol debugging is on
R3#terminal monitor
*Mar 17 19:08:34.371: RIP: sending v1 update to 255.255.255.255
via
   Serial0/0/1 (10.1.5.2)
*Mar 17 19:08:34.371: RIP: build update entries
*Mar 17 19:08:34.371:   subnet 10.1.10.0 metric 1
*Mar 17 19:08:34.371:   subnet 10.1.11.0 metric 1
*Mar 17 19:08:34.371:   subnet 10.1.12.0 metric 2
*Mar 17 19:08:40.107: RIP: received v1 update from 10.1.5.1 on
   Serial0/0/1
*Mar 17 19:08:40.107:      10.1.1.0 in 1 hops
*Mar 17 19:08:40.107:      10.1.2.0 in 1 hops
```

```
*Mar 17 19:08:40.107:      10.1.3.0 in 1 hops
*Mar 17 19:08:40.107:      10.1.4.0 in 1 hops
*Mar 17 19:08:40.107:      10.1.6.0 in 2 hops
*Mar 17 19:08:40.107:      10.1.7.0 in 2 hops
*Mar 17 19:08:40.107:      10.1.8.0 in 2 hops
*Mar 17 19:08:40.107:      10.1.9.0 in 2 hops
*Mar 17 19:08:47.535: RIP: sending v1 update to 255.255.255.255
via
   FastEthernet0/1 (10.1.11.1)
*Mar 17 19:08:47.535: RIP: build update entries
*Mar 17 19:08:47.535:   subnet 10.1.1.0 metric 2
*Mar 17 19:08:47.535:   subnet 10.1.2.0 metric 2
*Mar 17 19:08:47.535:   subnet 10.1.3.0 metric 2
*Mar 17 19:08:47.535:   subnet 10.1.4.0 metric 2
*Mar 17 19:08:47.535:   subnet 10.1.5.0 metric 1
*Mar 17 19:08:47.535:   subnet 10.1.6.0 metric 3
*Mar 17 19:08:47.535:   subnet 10.1.7.0 metric 3
*Mar 17 19:08:47.535:   subnet 10.1.8.0 metric 3
*Mar 17 19:08:47.535:   subnet 10.1.9.0 metric 3
*Mar 17 19:08:47.535:   subnet 10.1.10.0 metric 1
*Mar 17 19:08:49.331: RIP: received v1 update from 10.1.11.2 on
   FastEthernet0/1
*Mar 17 19:08:49.331:      10.1.12.0 in 1 hops
R3#undeug all
*Mar 17 19:08:47.535:   subnet 10.1.10.0 metric 1
*Mar 17 19:08:49.331: RIP: received v1 update from 10.1.11.2 on
   FastEthernet0/1
```

Şimdi önemle vurguladığım bölümlere bakalım. İlk olarak RIP, 255.255.255.255'e, 10.1.5.2 vasıtasıyla, serial0/0/1 interface'inden v1 paketi gönderiyor. (herkese gönderilen broadcast). Burası, RIPv2'nin işe yarayacağı yerdir. Niçin? Çünkü RIPv2, broadcast göndermez. O, 224.0.0.9 multicast adresini kullanır. RIP paketleri, router olmayan bir network'e aktarılabilse de host'ların hepsi, onları göz ardı edecekti. Bu, RIPv2'yi RIPv1'e göre biraz daha gelişmiş kılar. R3'ümüzde, passive-interface kullanıyoruz. Böylece, router bağlı olmayan bir LAN'a broadcast göndermiyoruz.

Güzel, şimdi, şunu kontrol edelim; serial0/0/1'den gönderilen son yayın, sadece 10.1.10.0, 10.1.11.0 ve 10.1.12.0 network'lerine gönderildiği halde yayınları 10.1.11.0 ve 10.1.12.0 hariç, tüm network'ler için FastEthernet0/1'den göndermektedir. Niçin? Şayet split horizon kuralı diyorsanız, bildiniz! R3 router'ımız, Corp router'ından aldığı bu network'leri tekrar Corp router'ına göndermeyecektir.

*Şayet bir route'un metriği 16 görünüyorsa, bu bir route poison'dur ve route, erişilemez olarak yayınlanacaktır.*

NOT

## Debug ip rip Komutu ile Hata Tespiti

Şimdi, hem bir problemi bulmak hem de farklı örnek network'ten bir router'da RIP'in nasıl yapılandırıldığını anlamak için `debug ip rip` komutunu kullanalım:

```
07:12:58: RIP: sending v1 update to 255.255.255.255 via
  FastEthernet0/0 (172.16.1.1)
07:12:58:  network 10.0.0.0, metric 1
```

```
07:12:58:  network 192.168.1.0, metric 2
07:12:58: RIP: sending v1 update to 255.255.255.255 via
  Serial0/0 (10.0.8.1)
07:12:58:  network 172.16.0.0, metric 1
07:12:58: RIP: Received v1 update from 10.0.15.2 n Serial0/0
07:12:58: 192.168.1.0 in one hop
07:12:58: 192.168.168.0 in 16 hops (inaccessible)
```

Güncellemelerden, 10.0.0.0, 192.168.1.0 ve 172.16.0.0 network'leri hakkında bilgi gönderdiğimizi görebilirsiniz. Fakat 10.0.0.0 ve 172.16.0.0 network'leri, 1 hop sayısı (metrik) ile yayınlanıyor. Yani, bu network'ler direkt bağlıdırlar. 192.168.1.0, 2 metriğiyle yayınlanmaktadır (direk bağlı değildir).

Bu olduğundan, konfigürasyonumuz şöyle görünmelidir:

```
Router(config)#router rip
Router(config-router)#network 10.0.0.0
Router(config-router)#network 172.16.0.0
```

Buna bakarak başka bir şeyler olduğunu da fark edebilirsiniz: RIP network'ünün parçası olan en az iki router vardır. Çünkü iki interface'ten gönderiyoruz, fakat bir interface'ten RIP güncellemesi alıyoruz. Ayrıca, 192.168.168.0 network'ünün 16 hop'la yayınlandığına dikkat edin. RIP, maksimum 15 hop sayısına sahiptir, bu nedenle 16, erişilemez kabul edilmektedir. Öyleyse bu network, ulaşılamazdır.192.168.168.0 network'ünüzdeki bir host'a ping atmayı denerseniz, ne olur? Başarılı olamayacaksınız. Fakat 10.0.0.0 network'ünü pinglemeyi denerseniz, atabilmelisiniz.

Size göstermek istediğim bir çıktı daha var. Örnek router'ımızdan, `debug ip rip` ve `show ip route` çıktıları aşağıdaki gibidir:

```
07:12:56: RIP: received v1 update from 172.16.100.2 on Serial0/0
07:12:56:      172.16.10.0 in 1 hops
07:12:56:      172.16.20.0 in 1 hops
07:12:56:      172.16.30.0 in 1 hops

Router#sh ip route
[output cut]
Gateway of last resort is not set

   172.16.0.0/24 is subnetted, 8 subnets
C  172.16.150.0 is directly connected, FastEthernet0/0
C  172.16.220.0 is directly connected, Loopback2
R  172.16.210.0 is directly connected, Loopback1
R  172.16.200.0 is directly connected, Loopback0
R  172.16.30.0 [120/2] via 172.16.100.2, 00:00:04, Serial0/0
S  172.16.20.0 [120/2] via 172.16.150.15
R  172.16.10.0 [120/2] via 172.16.100.2, 00:00:04, Serial0/0
R  172.16.100.0 [120/2] is directly connected, Serial0/0
```

İki çıktıya bakarak kullanıcıların 172.16.20.0'a neden erişemediklerini söyleyebilir misiniz?

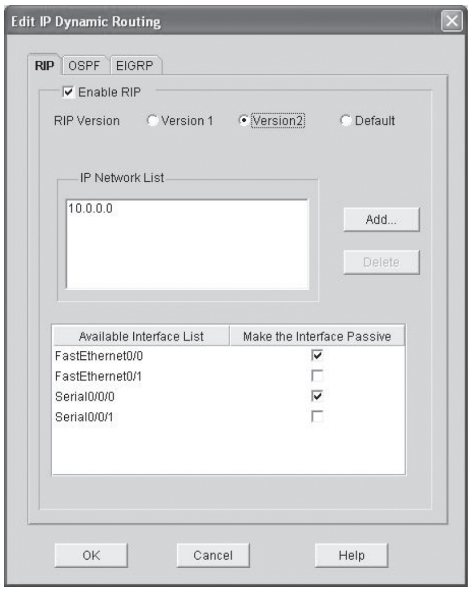
Debug çıktısı, 172.16.20.0 network'ünün bir hop uzakta olduğunu ve 172.16.100.2'den serial0/0 üzerinden alındığını gösterir. ***Show ip route*** çıktısını kontrol ederek, 172.16.20.0 hedef adresi olan bir paketin, statik route'tan dolayı 172.16.150.15'e gönderildiğini görebilirsiniz. Çıktı ayrıca, 172.16.150.0'ın, FastEthernet'e direkt bağlı olduğunu ve 172.16.20.0 network'ünün, serial0/0'dan çıktığını göstermektedir.

## Ağ Topluluğumuzda RIPv2'yi Etkinleştirmek

Bölüm 7'ye geçmeden ve EIGRP ve OSPF'i yapılandırmadan önce, router'larımızda RIPv2'yi etkinleştirmek istiyorum. Sadece birkaç saniye sürecek. Konfigürasyon şöyledir:

```
Corp#config t
Corp(config)#router rip
Corp(config-router)#version 2
Corp(config-router)#^Z

R1#config t
R1(config)#router rip
R1(config-router)#version 2
R1(config-router)#^Z

R2#config t
Enter configuration commands, one per line.  End with CNTL/Z.
R2(config)#router rip
R2(config-router)#version 2
R2(config-router)#^Z
```

R3 routerı için sadece Version 2 butonuna ve OK butonuna tıkladım. Bitti.




```
871W#config t
871W#(config)#router rip
871W#(config-router)#version 2
871W#(config-router)#^Z
```

Bu, şimdiye kadar kitapta yaptığımız en kolay konfigürasyondu. Routing tablolarımızda değişiklik olup olmadığına bakalım. Aşağıda, R3 router'ının, routing tablosunu görebilirsiniz:

```
     10.0.0.0/24 is subnetted, 12 subnets
C       10.1.11.0 is directly connected, FastEthernet0/1
C       10.1.10.0 is directly connected, FastEthernet0/0
R       10.1.9.0 [120/2] via 10.1.5.1, 00:00:23, Serial0/0/1
R       10.1.8.0 [120/2] via 10.1.5.1, 00:00:23, Serial0/0/1
R       10.1.12.0 [120/1] via 10.1.11.2, 00:00:18, FastEthernet0/1
R       10.1.3.0 [120/1] via 10.1.5.1, 00:00:23, Serial0/0/1
R       10.1.2.0 [120/1] via 10.1.5.1, 00:00:23, Serial0/0/1
R       10.1.1.0 [120/1] via 10.1.5.1, 00:00:23, Serial0/0/1
R       10.1.7.0 [120/2] via 10.1.5.1, 00:00:23, Serial0/0/1
R       10.1.6.0 [120/2] via 10.1.5.1, 00:00:23, Serial0/0/1
C       10.1.5.0 is directly connected, Serial0/0/1
R       10.1.4.0 [120/1] via 10.1.5.1, 00:00:23, Serial0/0/1
R3#
```

Bana aynı görünüyor. Debugging'i açacağım ve bize yeni bir şeyler gösterip göstermeyeceğine bakacağım:

```
*Mar 17 19:34:00.123: RIP: sending v2 update to 224.0.0.9 via
   Serial0/0/1 (10.1.5.2)
*Mar 17 19:34:00.123: RIP: build update entries
*Mar 17 19:34:00.123:   10.1.10.0/24 via 0.0.0.0, metric 1, tag 0
*Mar 17 19:34:00.123:   10.1.11.0/24 via 0.0.0.0, metric 1, tag 0
*Mar 17 19:34:00.123:   10.1.12.0/24 via 0.0.0.0, metric 2, tag
0col
*Mar 17 19:34:03.795: RIP: received v2 update from 10.1.5.1 on
   Serial0/0/1
[output cut]
```

Bingo! Şuna bakın! Network'ler hala, 30 saniyede bir yayınlanmakta. Fakat yayınlarını, v2 olarak ve 224.0.0.9 multicast adresi ile göndermektedirler. Şimdi, `show ip protocols` çıktısına bakalım:

```
R3#sh ip protocols
Routing Protocol is "rip"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Sending updates every 30 seconds, next due in 27 seconds
  Invalid after 180 seconds, hold down 180, flushed after 240
  Redistributing: rip
  Default version control: send version 2, receive version 2
    Interface             Send  Recv  Triggered RIP  Key-chain
    FastEthernet0/1       2     2

    Serial0/0/1           2     2
```

```
Automatic network summarization is not in effect
Maximum path: 4
Routing for Networks:
  10.0.0.0
Passive Interface(s):
  FastEthernet0/0
  Serial0/0/0
Routing Information Sources:
  Gateway         Distance      Last Update
  10.1.11.2            120      00:00:00
  10.1.5.1             120      00:00:02
Distance: (default is 120)
```

Şimdi, RIPv2 gönderip alabiliyoruz. Bir şeyleri tamamlamak güzel, değil mi? Şimdi, sonraki bölüme geçmeye hazırsınız!

# Özet

Bu bölümde IP routing detaylı bir şekilde işlendi. Bu bölümde işlenen temelleri anlamanız gerçekten çok önemlidir. Çünkü bir Cisco router'da yapılan her şey, tipik olarak yapılandırılan ve çalışan bir IP routing'e ihtiyaç duyacaktır.

Bu bölümde, IP routing'in paketleri router'lar arasında ve hedef host'a taşımak için frame'leri nasıl kullandığını öğrendiniz. Buradan, router'larımızda statik routing yapılandırdık ve bir hedef network'e, en iyi yolu belirlemek için IP tarafından kullanılan administrative distance'ı tartıştık. Şayet stub network'e sahipseniz, default routing yapılandırabilirsiniz. Default routing, bir router'da gateway of last resort ayarlamaktadır.

Sonra, detaylı olarak dinamik routingi, özellikle de RIP'i ve bir ağ topluluğunda nasıl çalıştığını tartıştık. RIP'i doğrulayarak tamamladık ve sonra küçük ağ topluluğumuza RIPv2'yi ekledik.

Sonraki bölümde, EIGRP ve OSPF'i tartışarak dinamik routing'e devam edeceğiz.

# Sınav Gereklilikleri

**Temel IP routing işleyişini anlamak:** Frame'in her hop'ta değiştiğini, fakat paketin asla değişmediğini ve herhangi bir yolla ayarlanmadığını hatırlamanız gerekmektedir.

**MAC adreslerinin daima lokal olduğunu anlamak:** Bir MAC (donanım) adresi, sadece lokal LAN'da kullanılabilmektedir. Bir router'ın interface'ine gönderilmez.

**Frame'in bir paketi sadece iki yere taşıdığını anlamak:** Bir frame, bir paketi, LAN'da göndermek için MAC (donanım) adresleri kullanır. Frame, paketi, LAN'daki bir host'a ya da (uzak bir network'e hedeflendiyse) bir router'ın interface'ine taşıyacaktır.

**RIP routingin nasıl yapılandırıldığını anlamak:** RIP routingi yapılandırmak için, ilk olarak global configuration modda olmalısınız ve sonra ***`router rip`*** komutunu girmelisiniz. Daha sonra, direk bağlı network'lerinizi ekleyin. (Classful adres kullandığınızdan emin olun.)

**RIP routingin nasıl doğrulandığını hatırlamak:** `Show ip route` komutu, routing tablosunun içeriğini sağlayacaktır. Tablonun sol tarafındaki *R*, RIP'le öğrenilenn bir route'u belirtir. ***`Debug ip rip`*** komutu size, router'ınızda gönderilip alınan RIP güncellemelerini göstermektedir. Şayet 16 metrikli bir route görürseniz, bu route, down kabul edilir.

**RIPv1 ve RIPv2 arasındaki farklılıkları hatırlamak:** RIPv1, her 30 saniyede broadcast gönderir ve AD'si 120 dir. RIPv2, her 30 saniyede multicast (224.0.0.9) gönderir ve 120 AD'ye sahiptir. RIPv2, route güncellemeleri ile subnet mask bilgini gönderir. Böylece, classless network'leri ve discontiguous ağları destekler. RIPv2, router'lar arasında kimlik denetimini de destekler, RIPv1 desteklemez.

# Yazılı Lab 6

Aşağıdaki soruların cevaplarını yazın:

1. 172.16.10.0/24 network'üne, 172.16.20.1 next-hop gateway ve 150 AD'si ile bir statik route oluşturun.
2. SDM'den, RIP'i etkinleştirdiniz ve seri interface'iniz için passive-interface kutusundaki işareti kaldırdınız. Bunun anlamı nedir?
3. 172.16.40.1'e statik route oluşturmak için hangi komutu yazarsınız?
4. Şayet default routing kullanıyorsanız, ayrıca hangi komut kullanılmalıdır?
5. Hangi network tipinde bir default route kullanırsınız?
6. Router'ınızdaki routing tablosunu görmek için hangi komutu kullanırsınız?
7. Statik ya da default route oluşturduğunuzda, next-hop adres kullanmak zorunda değilsiniz. _____ kullanabilirsiniz?
8. Doğru/Yanlış: bir hedef host'una erişmek için, uzak host'un MAC adresini bilmek zorundasınız.
9. Doğru/Yanlış: bir hedef host'una erişmek için, uzak host'un IP adresini bilmek zorundasınız.
10. Şayet bir DCE seri interface'iniz varsa, bu interface'in çalışması için hangi komutu girmelisiniz?
11. Bir router'da RIP routingi açmak ve 10.0.0.0 network'ünü yayınlamak için kullanılan komutları yazın.
12. Serial1 interface'inden RIP bilgisini göndermesini durdurmak için bir router'da kullanılan komutları yazın.
13. Distance-vector network'lerde routing kısır döngüsünü durdurmaya yardım etmesi için triggered güncellemelerle ne kullanılır?
14. Bir link arızalanır arızalanmaz bir maximum hop count göndererek, distance-vector ağlarında routing kısır döngülerini ne durdurur?
15. Distance vector ağlarda, bilgiyi aldığı interface'den göndermeyerek, routing kısır döngülerini ne durdurur?
16. RIP routing güncellemelerini, router'da gönderilip alınıyor gibi konsol oturumuna göndermek için hangi komut kullanılır?

(Yazılı lab'ın cevapları, bu bölümün gözden geçirme sorularının cevaplarından sonra bulunabilir.)

# Pratik Lab'lar

Aşağıdaki pratik lab'larda, üç router'lı bir network'ü yapılandıracaksınız.

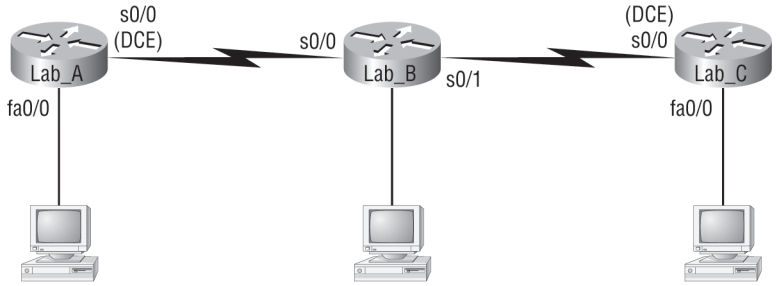
NOT *Bu bölümdeki pratik lab'lar, gerçek Cisco router'ları kullanmayı içermektedir. Şayet RouterSim.com ya da Sybex'ten yazılım kullanıyorsanız, lütfen, bu programlarda bulunan pratik lab'ları kullanın.*

Bu modül şunları içermektedir:

Lab 6.1: Statik Route'lar oluşturmak.

Lab 6.2: RIP Routing yapılandırmak.

Şekil 6.17, tüm router'ları yapılandırmak için kullanılacaktır.




***Şekil 6.17:*** *Pratik lab ağ topluluğu.*

Tablo 6.5, her router için IP adreslerimizi göstermektedir. (Her interface /24 mask kullanır.)

**Tablo 6.5:** IP Adreslerimiz

| Router | Interface | IP Adresi |
|---|---|---|
| Lab_A | F0/0 | 172.16.10.1 |
| Lab_A | S0/0 | 172.16.20.1 |
| Lab_B | S0/0 | 172.16.20.2 |
| Lab_B | S0/1 | 172.16.30.1 |
| Lab_C | S0/0 | 172.16.30.2 |
| Lab_C | Fa0/0 | 172.16.40.1 |

Bu lab'lar, Lab_B router'ındaki LAN interface'i kullanılmaksızın yazıldılar. Şayet gerekirse, bu LAN'ı ekleyebilirsiniz.

## Pratik Lab 6.1: Statik Route'lar Oluşturmak

Bu lab'da, üç router'da da bir statik route oluşturacaksınız. Böylece router'lar tüm network'leri görecekler. Tamamlandığında, ping atarak doğrulayın.

1. 1. Lab_A router'ı, 172.16.10.0 ve 172.16.20.0 network'lerine bağlıdır. 172.16.30.0 ve172.16.40.0 network'leri için route eklemeniz gerekmektedir.

```
Lab_A#config t
Lab_A(config)#ip route 172.16.30.0 255.255.255.0
  172.16.20.2
Lab_A(config)#ip route 172.16.40.0 255.255.255.0
  172.16.20.2
```

2. Privileged moda giderek ve `copy run start` yazıp, Enter tuşuna basarak Lab_A router'ı için o anki konfigürasyonu kaydedin.

3. Lab_B router'ında, 172.16.20.0 ve 172.16.30.0 network'lerine direkt bağlısınız. 172.16.10.0 ve 172.16.40.0 network'leri için route eklemeniz gerekir.

```
Lab_B#config t
Lab_B(config)#ip route 172.16.10.0 255.255.255.0
  172.16.20.1
Lab_B(config)#ip route 172.16.40.0 255.255.255.0
  172.16.30.2
```

4. User mod'a giderek ve `copy run start` yazıp Enter tuşuna basarak Lab_B router'ı için o anki konfigürasyonu kaydedin.

5. Lab_C router'ında direkt bağlı olmayan 172.16.10.0 ve 172.16.20.0 network'lerini görmek için bir statik route oluşturun. Lab_C'nin tüm network'leri görebileceği şekilde statik route oluşturun. Aşağıdaki gibi:

```
Lab_C#config t
Lab_C(config)#ip route 172.16.10.0 255.255.255.0
  172.16.30.1
Lab_C(config)#ip route 172.16.20.0 255.255.255.0
  172.16.30.1
```

6. User moda giderek ve `copy run start` yazıp, Enter tuşuna basarak Router 2501B için o anki konfigürasyonu kaydedin.
7. Dört network'ün de göründüğünden emin olmak için, routing tablonuzu kontrol edin.
8. Şimdi, her router'dan host'larınızı ve her router'dan, tüm router'ları ping'leyin. Kurulum doğru ise çalışacaktır.9.

## Pratik Lab 6.2: RIP Routingi Yapılandırmak

Bu lab'da, statik routing yerine RIP dinamik routing protokolünü kullanacağız.

1. `no ip route` komutunu kullanarak, router'ınızda, yapılandırılan statik veya default route'ları kaldırın. Örneğin, aşağıda, Lab_A router'ınızdaki, statik route'ları nasıl kaldıracağınızı görebilirsiniz:

```
Lab_A#config t
Lab_A(config)#no ip route 172.16.30.0 255.255.255.0
  172.16.20.2
Lab_A(config)#no ip route 172.16.40.0 255.255.255.0
  172.16.20.2
```

   Aynı şeyi, Lab_B ve Lab_C router'ları içinde yapın. Sadece direkt bağlı network'lerinizin routing tablosunda olduğunu kontrol edin.

2. Statik ve default route'ları temizledikten sonra, `config t` yazarak, Lab_A router'ında configuration moda gidin.
3. `Router rip` yazıp Enter tuşuna basarak router'ınıza RIP routing'i kullanmasını söyleyin:

```
config t
router rip
```

4. `Network 172.16.0.0` yazıp Enter tuşuna basarak, yayınlanmasını istediğiniz network adresini ekleyin.
5. Configuration mod'dan çıkmak için, Ctrl+Z tuş bileşimine basın.
6. Lab_B ve Lab_C router'larına gidin ve aynı komutları yazın:

```
Config t
Router rip
network 172.16.0.0
```

7. Her router'da aşağıdaki komutları yazarak router'larda RIP çalıştığının doğruluğunu kontrol edin:

```
show ip protocols
show ip route
show running-config veya show run
```

8. Her router'da `copy run star` veya `copy running-config startup-config` yazıp Enter'a basarak konfigürasyonlarınızı kaydedin.
9. Tüm network ve host'ları pingleyerek network'ün doğru çalıştığını kontrol edin.

# Gözden Geçirme Soruları

NOT *Aşağıdaki sorular, bu bölümün materyallerini anladığınızı test etmek için tasarlanmıştır. İlave sorulara nasıl sahip olacağınızla ilgili daha fazla bilgi için, lütfen bu kitabın Giriş bölümüne bakın.*

1. 206.143.5.0 network'ü, ISP'sine bağlanması için ACME firmasına tanımlanmıştır. Acme'nin yöneticisi, Internete erişmesi için bir router'ı komutlarla yapılandırmak istemektedir. Tüm network'e Internet erişimi sağlamak için Gateway isimli router'da hangi komutlar girilmelidir?

   A. `Gateway(config)#ip route 0.0.0.0 0.0.0.0 206.143.5.2`

   B. `Gateway(config)#router rip`

   C. `Gateway(config-router)#network 206.143.5.0`

   D. `Gateway(config)#router rip`

   E. `Gateway(config-router)#network 206.143.5.0 default`

   F. `Gateway(config)#ip route 206.143.5.0 255.255.255.0 default`

   G. `Gateway(config)#ip default-network 206.143.5.0`

2. RIP güncellemelerinin alınmasına izin verildiği halde bir interface'den gönderilmesini durdurmak için hangi komut kullanılmaktadır?

   A. `Router(config-if)#no routing`

   B. `Router(config-if)#passive-interface`

   C. `Router(config-router)#passive-interface s0`

   D. `Router(config-router)#no routing updates`

3. `ip route 172.16.4.0 255.255.255.0 192.168.4.2` komutu hakkında hangi ifadeler doğrudur? (İki şık seçin.)

   A. Komut bir statik route oluşturmak için kullanılmaktadır.

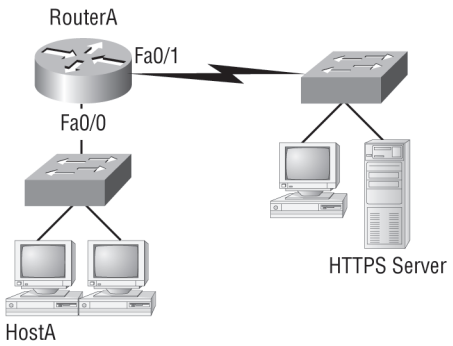
   B. Varsayılan administrative distance kullanılmaktadır.

   C. Komut bir default route oluşturmak için kullanılmaktadır.

   D. Kaynak adres için subnet mask, 255.255.255.0'dır.

   E. Komut bir stub network oluşturmak için kullanılmaktadır.

4. Şekilde gösterilen network'te, HTTPS sunucusuna veri göndermek için Host_A tarafından hangi hedef adresi kullanılacaktır? (İki şık seçin.)




   A. Switch'in IP adresi

   B. Uzak switch'in MAC adresi

   C. HTTPS sunucusunun IP adresi

D. HTTPS sunucusunun MAC adresi

E. RouterA'nın Fa0/0 interface'inin IP adresi

F. RouterA'nın Fa0/0 interface'inin MAC adresi

5. Aşağıdaki çıktıyla ilgili hangisi doğrudur? (İki şık seçin.)

```
04:06:16: RIP: received v1 update from 192.168.40.2 on Serial0/1
04:06:16:      192.168.50.0 in 16 hops (inaccessible)
04:06:40: RIP: sending v1 update to 255.255.255.255 via
   FastEthernet0/0 (192.168.30.1)
04:06:40: RIP: build update entries
04:06:40:       network 192.168.20.0 metric 1
04:06:40:       network 192.168.40.0 metric 1
04:06:40:       network 192.168.50.0 metric 16
04:06:40: RIP: sending v1 update to 255.255.255.255 via Serial0/1
   (192.168.40.1)
```

A. Bu güncellemeye katılan router'da üç interface vardır.

B. 192.168.50.1'e ping başarılı olacaktır.

C. Bilgileri değiş tokuş eden en az iki router vardır.

D. 192.168.40.2'ye ping başarılı olacaktır.

6. Split horizon nedir?

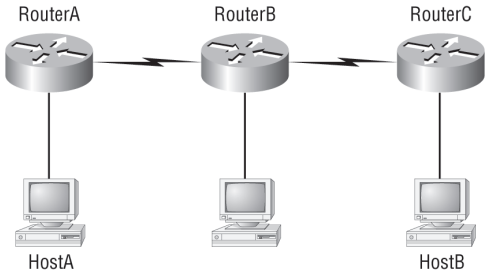
A. Bir route hakkındaki bilgi, orijinal güncellemenin geldiği yöne tekrar gönderilmemelidir.

B. Büyük bir bus (horizon) fiziksel network'ünüz olduğunda, trafiği böler.

C. Düzenli güncellemelerin arızalı linke broadcast edilmesini durdurur.

D. Düzenli güncelleme mesajlarının arızalan bir route'a tekrar gönderilmesini engeller.

7. Şayet Host_A, Host_B ile haberleşmeye çalışıyorsa ve RouterC'nin F0/0 interfece'i gittiyse, aşağıdakilerden hangisi doğrudur?




A. RouterC, HostB'ye erişilemeyeceği konusunda HostA'yı bilgilendirmek için bir ICMP kullanır.

B. RouterC, HostB'ye erişilemeyeceği konusunda RouterB'yi bilgilendirmek için bir ICMP kullanır.

C. RouterC, HostB'ye erişilemeyeceği konusunda RouterA ve RouterB'yi bilgilendirmek için bir ICMP kullanır.

D. RouterC, bir hedef erişilemez mesaj tipi gönderecektir.

E. RouterC, bir router seçme mesaj tipi gönderecektir.

F. RouterC, bir source quench mesaj tipi gönderecektir.

8. Classless routing protokolleri hakkında hangi ifade doğrudur? (iki şık seçin)
   A. Discontiguous network'lerin kullanılmasına izin verilmez.
   B. Değişken uzunlukta subnet mask'ların kullanılması kabul edilecektir.
   C. RIPv1, classless bir routing protokol'dür.
   D. IGRP, aynı autonomous system ile classless routing'i desteklemektedir.
   E. RIPv2, classless routing'i desteklemektedir.
9. Distance vector ve link state protokolleri hakkında aşağıdakilerden hangi ikisi doğrudur?
   A. Link state, periyodik zaman aralıklarında komple routing tablosunu, tüm aktif interface'lerinden gönderir.
   B. Distance vector, periyodik zaman aralıklarında komple routing tablosunu, tüm aktif interface'lerinden gönderir.
   C. Link state, ağ topluluğundaki tüm router'lara kendi linklerinin durumunu içeren güncellemeler gönderir.
   D. Distance vector, ağ topluluğundaki tüm router'lara kendi linklerinin durumunu içeren güncellemeler gönderir.
10. RIP routing güncellemelerini hangi komut görüntüler?
    A. `show ip route`
    B. `debug ip rip`
    C. `show protocols`
    D. `debug ip route`
11. Routing kısır döngülerini engellemek için hangisi kullanılır? (üç şık seçin)
    A. CIDR
    B. Split horizon
    C. Authentication
    D. Classless masking
    E. Holddown timers
12. Bir network yöneticisi, show ip route komutu çıktısına bakıyor. RIP ve IGRP ile yayınlanan bir network, routing tablosunda, IGRP route işaretli olarak görünmektedir. Routing tablosunda, bu network'e neden RIP route kullanılmamaktadır?
    A. IGRP, daha hızlı bir güncelleme timer sağlar.
    B. IGRP, daha düşük bir administrative distance'a sahiptir.
    C. RIP, bu route'dan daha yüksek bir metriğe sahiptir.
    D. IGRP route, daha az hop'a sahiptir.
    E. RIP hattında, routing döngüsü vardır.
13. Router konsolunuzda `debug ip rip` yazdınız ve 172.16.10.0'ın, size 16 metriğiyle yayınlandığını gördünüz. Bunun anlamı nedir?
    A. Route, 16 hop uzaktadır.
    B. Route'un gecikmesi, 16 mikro saniyedir.
    C. Route erişilemez..
    D. Route'da saniyede 16 mesaj kuyruğu oluşmaktadır.

14. Uzak bir network'e en iyi yolu bulmak için IGRP, aşağıdakilerden hangisini, varsayılan parametre olarak kullanacaktır?

    A. Hop count

    B. MTU

    C. Toplam interface gecikmesi

    D. STP

    E. Hattın bant genişliği değeri

15. Corporate router'ı, 192.168.214.20 kaynak adresi ve 192.168.22.3 hedef adresi ile bir IP paketi alır. Corporate router'ından alınan aşağıdaki çıktıya bakılarak, router'ın bu paketi ne yapacağı söylenebilir?

```
Corp#sh ip route
[output cut]
R    192.168.215.0 [120/2] via 192.168.20.2, 00:00:23, Serial0/0
R    192.168.115.0 [120/1] via 192.168.20.2, 00:00:23, Serial0/0
R    192.168.30.0 [120/1] via 192.168.20.2, 00:00:23, Serial0/0
C    192.168.20.0 is directly connected, Serial0/0
C    192.168.214.0 is directly connected, FastEthernet0/0
```

    A. Paket atılacaktır.

    B. Paket, S0/0 interface'inden route edilecektir.

    C. Router, hedefi bulmak için broadcast gönderecektir.

    D. Paket, Fa0/0 interface'inden route edilecektir.

16. Şayet routing tablonuzda, aynı network'e, statik RIP ve IGRP route varsa varsayılanda hangi route paketleri yollamak için kullanılacaktır?

    A. Kullanılabilir tüm route'lar

    B. RIP route

    C. Statik route

    D. IGRP route

    E. Hepsi, yük-dengelemesi yapacaktır.

17. Aşağıdaki routing tablosuna sahipsiniz. Komşu routing tablonuzda, aşağıdaki hangi network'ler yer almayacaktır?

```
R    192.168.30.0/24 [120/1] via 192.168.40.1, 00:00:12, Serial0
C    192.168.40.0/24 is directly connected, Serial0
     172.16.0.0/24 is subnetted, 1 subnets
C       172.16.30.0 is directly connected, Loopback0
R    192.168.20.0/24 [120/1] via 192.168.40.1, 00:00:12, Serial0
R    10.0.0.0/8 [120/15] via 192.168.40.1, 00:00:07, Serial0
C    192.168.50.0/24 is directly connected, Ethernet0
```

    A. 172.16.30.0

    B. 192.168.30.0

    C. 10.0.0.0

    D. Tamamı, komşu routing tablosuna yerleştirilecektir.

18. İki bağlı router, RIP routing ile yapılandırılmışlardır. Bir router, bir network'e routing tablosunda daha yüksek cost ile bulunan bir routing güncellemesi aldığında, sonuç ne olacaktır?
    A. Güncellenmiş bilgi, mevcut routing tablosuna eklenecektir.
    B. Güncelleme göz ardı edilecektir ve başka işlem olmaz.
    C. Güncelleme bilgisi, mevcut routing tablosu kaydı ile değiştirilir.
    D. Mevcut routing tablo kaydı, routing tablosundan silinecektir ve tüm router'lar converge olmak için routing güncellemelerini değiş tokuş edecektir.
19. Route poisoning nedir?
    A. Bir router'dan alınan protokolü, zehir hapı olarak geri gönderir. Bu, düzenli güncellemeleri durdurur.
    B. Bir router'dan alınan, üreten router'a geri gönderilemeyen bilgidir.
    C. Bu, düzenli güncelleme mesajlarının, yeni gelen bir route'u eski hale döndürmesini engeller.
    D. Bir router'ın, arızalı bir link için metriği sonsuza ayarlamasını açıklar.
20. RIPv2 hakkında aşağıdakilerden hangisi doğrudur?
    A. RIPv1'den daha düşük bir administartive distance'a sahiptir.
    B. RIPv1'den daha hızlı converge olur.
    C. RIPv1 ile aynı timer'lara sahiptir.
    D. RIPv1den daha zor yapılandırılır.

# Gözden Geçirme Sorularının Cevapları

1. A, G Aynı default route'u yapılandırmanın gerçekte üç farklı yolu vardır. Fakat sadece ikisi cevapta bulunmaktadır. İlki A şıkkındaki gibi bir default route'u, 0.0.0.0 0.0.0.0 mask ile ayarlayabilir ve sonra, next-hop belirtebilirsiniz. Son olarak, `ip default route` komutu ile G şıkkını seçebilirsiniz.
2. C `(config-router)#passive-interface` komutu, güncellemelerin bir interface'den gönderilmesini durdurur. Fakat route güncellemeleri hala alınacaktır.
3. A, B D şıkkı da doğru gibi görünse de, değildir. Mask, kaynak ağda değil de, uzak ağda kullanılan mask'tır. Statik route'un sonunda bir numara olmadığından, varsayılan 1 administartive distance'ını kullanıyordur.
4. C, FSwitch'ler, bir default gateway ya da diğer bir hedef olarak kullanılmazlar. Switchler, routing yapmazlar. MAC adresinin, daima bir router interface'i olduğunu hatırlamanız önemlidir. Bir frame'in hedef adresi, HostA'dan, RouterA'nın F0/0 interface'inin MAC adresi olacaktır. Bir paketin hedef adresi, HTTPS sunucusunun network interface card (NIC)'inin IP adresi olacaktır. Segment başlığındaki hedef port numarası, 443'dür (HTTPS).
5. C, D 192.168.40.2'ye route, erişilemezdir ve sadece s0/1 ve F0/0, RIP güncellemelerinde yer alır. Bir route güncellemesi alındığından, en az iki router RIP routing işleyişinin parçasıdır. 192.168.40.0 network'ü için bir route güncellemesi f0/0'dan gönderildiğinden ve 192.168.40.2 den bir route alındığından, bu adrese ping'in, başarılı olacağını tahmin edebiliriz.
6. A Bir split horizon, bir route'u, geri onu öğrendiği aynı router'a yayınlamayacaktır.
7. A, D RouterC, HostB'ye erişilemeyeceği konusunda HostA'yı bilgilendirmek için bir ICMP kullanır. Bunu, bir hedef erişilemez ICMP mesaj tipi göndererek çalıştıracaktır.
8. B, E Classful routing, ağ topluluğundaki tüm host'ların aynı maskı kullanmaları anlamına gelir. Classless routingin anlamı şudur; Variable Length Subnet Mask (VLSM) kullanabilirsiniz ve ayrıca discontiguous network kurulumunu destekleyebilirsiniz.
9. B, C Distance vector routing protokolü, periyodik zaman aralıklarında komple routing tablosunu, tüm aktif interface'lerinden gönderir. Link state routing protokolleri, ağ topluluğundaki tüm router'lara kendi linklerinin durumunu içeren güncellemeler gönderirler.
10. B *`debug ip rip`*, router'da gönderilip alınan, Routing Information Protocol (RIP)'i gösterir.
11. B, E RIPv2, RIPv1 ile aynı timer ve döngü engelleme yöntemlerini kullanır. Split Horizon, bir güncellemenin, alınan interface'den gönderilmesini durdurur. Holdown timer'ları, bir linkin gidip gelmesi durumunda, bir network'ün kararlı olması için bir zaman geçmesine izin verir.
12. B IGRP, 100 AD'ye sahipken, RIP, 120 AD'ye sahiptir. Bu nedenle router, 100'den daha büyük bir AD ile gelen bir route'ı kullanmayacaktır.
13. C Bir RIP network'ünde, 16 hop'lu bir AD'ye sahip olamazsınız. Şayet, 16 metrikli yayınlanan bir route alırsanız, onun erişilemez olduğu anlamına gelir.
14. C, E IGRP, uzak bir ağa en iyi yolu belirlemek için varsayılan olarak hattın bant genişliğini ve gecikmesini kullanır. Hattın gecikmesi, bazen toplam interface gecikmesi olarak belirtilmektedir.
15. A Routing tablosu, 192.168.22.0 network'üne bir route olmadığını gösterdiğinden, router, paketi atacaktır ve paketin üretildiği kaynak LAN olan FastEthernet0/0 interfece'ine bir ICMP hedef erişilemez mesajı gönderecektir.
16. C Statik route'lar, varsayılan 1 AD'sine sahiptir. Siz bunu değiştirmedikçe, bir statik route, daima diğer route'lardan öncelikli kullanılacaktır. Varsayılan olarak, IGRP, 100 AD ve RIP, 120 AD'ye sahiptir.

17. C Zaten 15 hop'ta olduğundan, 10.0.0.0 network'ü, routing tablosuna giremeyecektir. Bir hop daha, route'un metriğini 16 yapacaktır ve bu RIP network'lerinde geçersizdir.
18. B Bir router'dan, routing güncellemesi alındığında, router ilk olarak AD'yi kontrol eder ve daima en düşük AD'li route'u seçer. Bununla beraber, aynı AD'li iki route alınırsa, o zaman router, düşük AD'li olanı seçecektir (RIP'te düşük hop sayısını).
19. D Tutarsız güncellemelerin sebep olduğu problemlerden kaçınmanın ve network döngülerini durdurmanın diğer yolu, route poisoning'dir. Bir network gittiğinde, distance-vector protokolü, network'ü, 16 metrikli ya da erişilemez olarak (bazen bir *sonsuzluğa* işaret eder)yayınlayarak route poisoning başlatır.
20. C RIPv2, neredeyse RIPv1'e benzer. Aynı administrative distance ile timer'lara sahiptir ve aynı RIPv1 gibi yapılandırılır.21.

# Yazılı Lab 5 Cevapları

1. `ip route 172.16.10.0 255.255.255.0 172.16.20.1`
2. Şayet bir interface'in yanındaki kutu işaretli değilse, passive-interface kullanılmayacak ve RIP'in bu interface'den gönderilip alınacağı anlamına gelir.
3. `ip route 0.0.0.0 0.0.0.0 172.16.40.1`
4. `Router(config)#ip classless`
5. Stub network
6. `Router#show ip route`
7. Çıkış interface'i.
8. Yanlış. MAC adresi, uzak host olmaz, router interface'i olur.
9. Doğru
10. `Router(config-if)#clock rate speed`
11. `Router rip, network 10.0.0.0`
12. `Router rip, passive-interface s1`
13. Holddown timers
14. Route poisoning
15. Split horizon
16. `debug ip rip`

# 7 Enhanced IGRP (EIGRP) ve Open Shortest Path First (OSPF)

# 7 Enhanced IGRP (EIGRP) ve Open Shortest Path First (OSPF)

- EIGRP Özellikleri ve Operasyonu
- Komşu Tespiti
- Büyük Network'leri Desteklemesi İçin EIGRP'yi Kullanmak
- EIGRP ile Yük Dengelemesi
- EIGRP'nin Doğruluğunu Kontrol Etmek
- Open Shortest Path First (OSPF) Temelleri
- OSPF Konfigürasyonu
- OSPF Konfigürasyonunun Doğruluğunu Kontrol Etmek
- OSPF DR ve BDR Seçimleri
- OSPF ve Loopback Interface'leri
- OSPF Hata Tespiti
- EIGRP ve OSPF Summary Route'larını Yapılandırmak
- Özet
- Sınav Gereklilikleri
- Yazılı Lab 7
- Pratik Lab'lar
- Gözden Geçirme Soruları
- Gözden Geçirme Sorularının Cevapları
- Yazılı Lab 7 Cevapları

# Enhanced IGRP (EIGRP) ve Open Shortest Path First (OSPF)

Enhanced Interior Gateway Rsouting Protocol (EIGRP), Cisco router'larda çalışan, Cisco tescilli bir protokoldür. Bugün, muhtemelen kullanımda olan en popüler iki routing protokolünden biri olduğundan, EIGRP'yi anlamanız önemlidir. Bu bölümde, route'ları belirleyip ve seçip, yayınlamanın benzersiz yöntemine odaklanarak EIGRP'nin birçok özelliğini göstereceğim ve nasıl çalıştığını açıklayacağım.

Ayrıca size, günümüzdeki diğer popüler routing protokolü olan Open Shortest Path First'ü (OSPF) de tanıştıracağım. İlk olarak terminoloji ve iç operasyonuna aşina olarak ve sonra RIP'e karşı OSPF'in avantajlarını öğrenerek OSPF'i anlamak için sağlam bir temel oluşturacaksınız. Sonra, bazı broadcast ve non-broadcast network tiplerinde, OSPF uygulamalardaki sorunları araştıracağız. Farklı ve özel ağ kurulumu ortamlarında single-area OSPF'in nasıl oluşturulduğunu açıklayacakve her şeyin problemsiz çalıştığını kontrol etmeyi göstereceğim.

*Bu module ilgili son güncellemeler için www.lammle.com ve/veya www.sybex.com sitelerine bakın.*

NOT

## EIGRP Özellikleri ve Operasyonu

Enhanced IGRP (EIGRP), diğer Cisco tescilli protokol olan Interior Gateway Routing Protoco'den l (IGRP) daha üstün, classless, genişletilmiş bir distance-vector protokoldür. Bundan dolayı, genişletilmiş IGRP olarak tanımlanır. EIGRP, IGRP gibi aynı roting protokolü çalıştıran ve routing bilgisini paylaşan, ardışık router'ların grubunu açıklamak için bir autonomous system kavramı kullanır. Fakat IGRP'nin tersine, EIGRP, routing güncellemelerinde subnet maskını gönderir. Bildiğiniz gibi, subnet bilgisinin yayınlanması bize, network'lerimizi tasarladığımızda, Variable Length Subnet Mask'ları (VLSM) ve summarization kullanımı sağlar.

EIGRP bazen, hem distance-vector hem de link-state protokollerinin özelliklerine sahip olduğundan, bir hybrid routing protokol olarak belirtilir. Örnek olarak EIGRP, OSPF'in yaptığı gibi link-state paketleri göndermez. Onun yerine, network'ler ve yayınlayan router'ın perspektifinden onlara erişimin maliyeti (cost) hakkında bilgi içeren, klasik distance-vector güncellemeleri gönderir. EIGRP, link-state özellikleri de içerir. Başlangıçta, komşular arasındaki routing tablolarını senkronize eder ve sonra sadece topolojide değişiklik olduğunda, değişiklikleri içeren güncellemeler gönderir. Bu, EIGRP'nin büyük network'lerde kullanımını mümkün kılar. EIGRP, maksimum 255 hop sayısına sahiptir (varsayılan olarak 100'e ayarlıdır).

EIGRP'yi, IGRP ve diğer protokollerden üstün kılan bazı gelişmiş özellikler vardır. Başlıcaları şunlardır:

- Protokol-bağımsız modüller yardımıyla, IP ile IPv6'yı (ve bazı kullanışsız routed protokollerini) destekler.
- Classless olarak kabul edilir (RIPv2 ve OSPF gibi).
- VLSM/CIDR'ı destekler.
- Summarization ve discontiguous (ardışık olmayan) network'leri destekler.
- Etkili komşu tespiti.
- Reliable Transport Protocol (RTP) üzerinden iletişim.
- Diffusing Update Algorithm (DUAL) yardımıyla en iyi yol seçimi.

*Cisco, EIGRP'yi, distance-vector protokolü, bazen gelişmiş distance-vector ve hatta hybrid routing protokolü olarak belirtir.*

NOT

## Protokol-Bağımsız Modüller

EIGRP'nin en ilginç özelliklerinden birisi IP, IPX, AppleTalk ve IPv6 gibi birçok Network katmanı protokollerine routing desteği sağlamasıdır.(IPX ve AppleTalk kullanmayacağız, ama EIGRP onları desteklemektedir.) Ona yaklaşan ve birçok network katmanı protokolünü destekleyen tek routing protokolü, Intermediate System-to-Intermediate System'dır. (IS-IS)

EIGRP, protocol-dependent modüller (PDM) kullanarak, farklı Network katmanı protokollerini destekler. Her EIGRP PDM'i, belirli bir protokole uygulanan routing bilgisini içeren ayrı tablolar oluşturur. Yani, örneğin IP/EIGRP tabloları, IPv6/EIGRP tabloları vs. olacaktır.

# Komşu Tespiti

EIGRP router'ları, birbirlerinin route'larını değiş tokuş etmeyi başlatmalarından önce, komşu olmalıdırlar. Komşuluk kurulması için üç koşulun gerçekleşmesi gerekmektedir:

- Hello veya ACK alınması
- AS numaralarının eşleşmesi
- Aynı metrikler (K değerleri)

Link–state protokolleri, komşuluk kurmak için Hello mesajları kullanmaya meyillidir. Çünkü normalde, periyodik route güncellemeleri göndermezler ve yeni bir komşu geldiğinde ya da eskilerden biri gittiğinde, komşuları fark etmeye yardımcı olması için bazı mekanizmaların olması gerekir. Komşuluk ilişkilerini devam ettirmek için EIGRP router'ları, komşularından Hello paketleri almaya devam etmek zorundadır.

Farklı autonomous system'lere (AS) ait router'lar, routing bilgilerini otomatik olarak paylaşmaz ve komşu olmazlar. Bu davranış, büyük network'lerde kullanıldığında, belirli bir AS'e gönderilen route bilgisi miktarını düşürmek için gerçek bir fayda sağlayabilir. Dikkat etmeniz gereken tek şey, farklı AS'ler arasında manuel olarak redistribution yapmak zorunda kalmanızdır.

EIGRP'nin tüm routing tablosunu yayınlamak zorunda olacağı tek durum, yeni bir komşunun belirlenmesi ve Hello paketlerinin değiş tokuş edilmesiyle bir komşuluk kurulmasıdır. Bu olduğunda, her iki komşu da komple routing tablolarını birbirine yayınlar. Komşusunun route'larını öğrendikten sonra sadece routing tablosundaki değişiklikler gönderilir.

EIGRP router'ları, komşularının güncellemelerini aldıklarında onları lokal bir topoloji tablosunda tutar. Bu tablo, en iyi route'ların seçildiği ve routing tablosuna yerleştirildiği hammadde gibi, bilinen tüm komşu ve servislerden, bilinen tüm route'ları içerir.

Devam etmeden önce, bazı terimleri açıklayalım:

**Feasible distance:** Uzak bir ağa giden tüm yollar boyunca en iyi metriktir. Bu uzak network'ü yayınlayan komşuya olan metriği içerir. En iyi yol olarak kabul edildiğinden, routing tablosunda bulacağınız route'dır. Feasible distance'ın metriği, komşu (reported ya da advertised distance) tarafından rapor edilen metrik, artı route'u rapor eden komşuya olan metriktir.

**Reported/advertised distance:** Bu, bir komşu tarafından rapor edilen uzak bir ağın metriğidir. Ayrıca, komşunun routing tablosunun metriğidir ve topoloji tablosunda görünen parantez içindeki ikinci sayıdır.

**Neighbor table:** Her router, bitişik komşuları hakkında durum bilgilerini saklar. Yeni tespit edilen bir komşu öğrenildiğinde, komşunun interface ve adresi kaydedilir. Bu bilgi, RAM'de saklanan neighbor tablosunda tutulur. Her protokol-bağımsız modül için bir neighbor tablosu vardır. Sequence numaraları, güncelleme paketleri acknowledgment'larını eşleştirmek için kullanılır. Komşudan alınan son sequence numarası kaydedilir, böylece eski paketler belirlenebilir.

**Topoloji table:** Topoloji tablosu, protokol-bağımsız modüller tarafından yerleştirilir ve Diffusing Update Algorithm'den (DUAL) etkilenir. Her hedef adresini ve hedefi yayınlayan komşuların bir listesini tutarak, komşu router'lar tarafından yayınlanan tüm hedefleri içerir. Her komşu için sadece komşunun routing tablosundan gelen advertised metrik kaydedilir. Şayet komşu bu hedefi yayınlıyorsa route'un paketleri göndermesi için kullanılması gerekir.

*Neighbor ve topoloji tabloları RAM'de tutulur ve Hello ile güncelleme paketlerinin kullanılmasıyla devamlılığı sağlanır. Evet, routing tablosu RAM'de de tutulur, fakat bu bilgi sadece topoloji tablosundan toplanır.*

NOT

**Feasible successor:** Bir feasible successor, feasible distance'dan daha düşük reported distance'a sahip bir yoldur. EIGRP, topoloji tablosunda, altı feasible successor tutacaktır. Sadece en iyi metriği (successor) olanı routing tablosunda kopyalayıp yerleştirecektir. `Show ip route topology` komutu bir route'a, bilinen tüm EIGRP feasible route'ları gösterecektir.

*Bir feasible successor yedek bir route'dur ve topoloji tablosunda tutulur. Successor, topoloji tablosunda saklanır ve routing tablosuna kopyalanıp yerleştirilir.*

NOT

**Successor:** Successor route, uzak bir network'e en iyi route'dur. Bir successor route, trafiği bir hedefe göndermek için EIGRP tarafından kullanılır ve routing tablosunda tutulur. Topoloji tablosunda tutulan bir feasible successor ile yedeklenir (şayet varsa). Feasible distance kullanarak ve topoloji tablosunda yedek link olarak feasible succesorları kullanarak network anında converge edilir ve her komşusunu, sadece EIGRP tarafından gönderilen trafikle günceller.

## Reliable Transport Protocol (RTP)

EIGRP konuşan router'ler arasındaki mesajların iletişimini yönetmek için EIGRP, Reliable Transport Protocol (RTP) denilen tescilli bir protokol kullanır. İsminden de anlaşıldığı gibi, bu protokolün esas işlevi güvenilirliktir. Cisco, güncellemelerin çabuk iletilmesi ve veri alındılarının izlenmesi için multicast'leri ve unicast'leri kullanan bir mekanizma tasarlamıştır.

EIGRP multicast trafiği gönderdiğinde, 224.0.0.10 Klass D adresini kullanır. Daha önce söylediğim gibi her bir EIGRP router, komşularının kim olduğunun farkındadır ve her gönderdiği multicast için hangi komşularının yanıtladığına dair bir liste tutar. Eğer EIGRP bir komşusundan cevap almazsa aynı veriyi yeniden göndermek için unicast'leri kullanmaya başlar. Eğer 16 unicast denemesinden sonra da cevap alamazsa komşu ölmüş demektir. Genellikle bu proses reliable (güvenli) multicast olarak belirtilir.

Router'lar gönderdikleri bilgilerin kaydını her bir pakete bir sequence numarası vererek tutarlar. Bu teknikle, eski, gereksiz ya da sıralama dışı bilgilerin ulaştığını tespit etmeleri mümkündür.

Bu saydıklarımızı yapabiliyor olması sebebi ile EIGRP güzel bir protokoldür. Bu, başlangıçta routing veritabanlarının senkronize edilebilmesi ve sonra da, sadece değişiklikleri ileterek veritabanının tutarlılığının sürdürülmesi becerisine bağlıdır. Bu nedenle herhangi bir paketin kalıcı olarak kaybolması ya da paketlerin düzensiz iletilmesi, routing veritabanının bozulması ile sonuçlanabilir.

## Diffusing Update Algorithm (DUAL)

EIGRP, her bir uzak network için en iyi yolu seçip devamlılığını sağlamak için Diffusing Update Algorithm (DUAL) kullanır. Bu algoritma şunların yapılmasını sağlar:

- Mevcutta yoksa yedek route belirlenmesi
- VLSM desteği
- Dinamik route iyileştirmeleri
- Eğer route bulunamıyorsa, alternatif route sorgulaması

DUAL, EIGRP'nin tüm protokoller arasında en hızlı route convergence zamanına sahip olmasını sağlar. EIGRP'nin iki kat hızlı olmasının sebebi şunlardır: İlk olarak EIGRP router'ları, her uzak

network'e kendi cost'larını hesaplamakta kullanmak için komşularının route'larının bir kopyasını tutar. Şayet en iyi yol giderse yer değişecek en iyi route'u seçmek için topoloji tablosunun içindekileri mümkün olduğu kadar kolay inceleyebilir. İkincisi, şayet lokal topoloji tablosunda iyi bir alternatif yoksa EIGRP router'ları, komşularından çok hızlı bir şekilde yeni bir route bulmak için yardım ister. Yönleri sormaktan çekinmezler! Diğer router'lara güvenerek ve bilgileri kullanarak, DUAL'in diffusing özelliği için rapor sağlar.

Ve söylediğim gibi Hello protokolünün tüm amacı, yeni ya da ölü komşuların hızlı bir şekilde tespit edilmesini mümkün kılmaktır. RTP, naklederek ve sıraya koymak için güvenli bir mekanizma sağlayarak, bu çağrılara cevap verir. Bu güvenilir temel üzerine DUAL, en iyi yollar hakkında bilgileri seçmek ve devamlılığını sağlamaktan sorumludur.

# Büyük Network'leri Desteklemesi İçin EIGRP'yi Kullanmak

EIGRP, büyük network'lerde kullanılmasını mümkün kılan birçok güzel özelliğe sahiptir:

- Tek bir router'da çok sayıda AS'i destekler.
- VLSM ve summarization'ı destekler.
- Route belirleme ve onarımı.

Bu özelliklerin her biri, çok sayıda network ve çok büyük sayıda router desteğinin karmaşık bilmecesine bir parça ekler.

## Çoklu AS'ler

EIGRP, route bilgisini paylaşan router grubunu tespit etmek için autonomous system numaraları kullanır. Sadece aynı autonomous system numarasına sahip router'lar, route'ları paylaşırlar. Geniş network'lerde gerçekten karmaşık topolojiler ve routing tablolarıyla karşı karşıya kalabilirsiniz. Bu network'ler hesaplama işlemlerinin dağıtılması sırasında oldukça yavaş convergence olabilirler.

Bir yönetici, gerçekten büyük network'lerin yönetimini kolaylaştırmak için ne yapar? Network'ü, çok sayıda farklı AS'lere bölmek mümkündür. Her AS'i, ardışık router serilerini içerir ve route bilgileri, redistribution yardımıyla farklı AS'lere paylaştırılabilir.

EIGRP'de redistribution kullanımı diğer ilginç özellikler için bize yol gösterir. Normalde, EIGRP'nin administrative distance'ı (AD) 90'dır. Fakat bu, sadece internal EIGRP route'lar için geçerlidir. Diğer route tipi, external EIGRP route'dur ve AD'si 170'tir. Bu route'lar, manuel ya da otomatik redistribution'ın katkısıyla, EIGRP route tablolarında görünür ve network'lerin, EIGRP autonomous system'i dışından oluşturduğunu belirtir. Route'un, başka EIGRP autonomous system'den ya da OSPF gibi başka routing protokolünden oluşturulduğunun önemi yoktur. EIGRP'ye redistribute edildiğinde, hepsi extrenal (harici) route olarak kabul edilir.

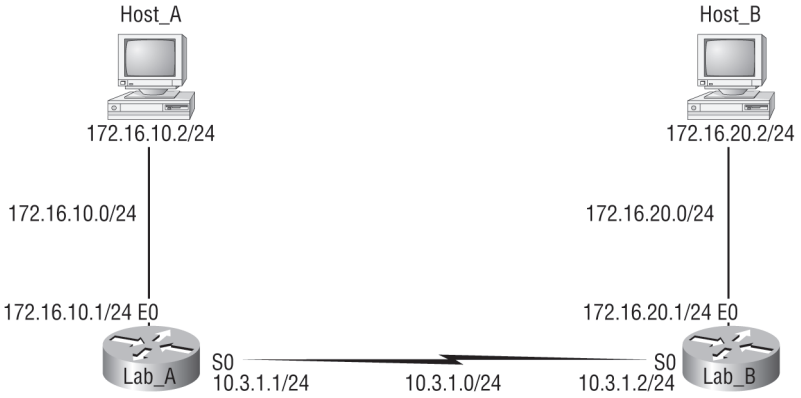
EIGRP ve redistribution ile ilgili bazı iyi haberler var. Diyelim ki, hepsinde IGRP çalışan router'lara sahip bir firmanız var. Network yöneticisi olarak henüz yeni işe alındınız, benim kitabımı okuduğunuz ve EIGRP'nin IGRP'ye karşı birçok üstünlüğü olduğunu bildiğinizden, network'ünüzde EIGRP çalışmasına karar verdiniz.

EIGRP'ye yavaşça geçmeniz gerektiğinden ve tüm router'ları, eşzamanlı değiştiremeyeceğiniz için redistribution yapılandırmanız gerekir, değil mi? EIGRP ile gerekmez! EIGRP için IGRP'de kullanılan aynı autonomous system numarasını kullandığınız müddetçe EIGRP, route'ları, IGRP'den EIGRP'ye otomatik olarak redistribute edecektir. Tabi ki, EIGRP bunları external route olarak görecektir (170 AD). Bu nedenle, bu sizin sonsuza kadar isteyeceğiniz bir şey değildir. Mümkün olduğu kadar çabuk geçmek istiyorsanız, bu redistribution özelliği sayesinde bu geçişi kolaylaştırabilirsiniz.

## VLSM Desteği ve Summarization

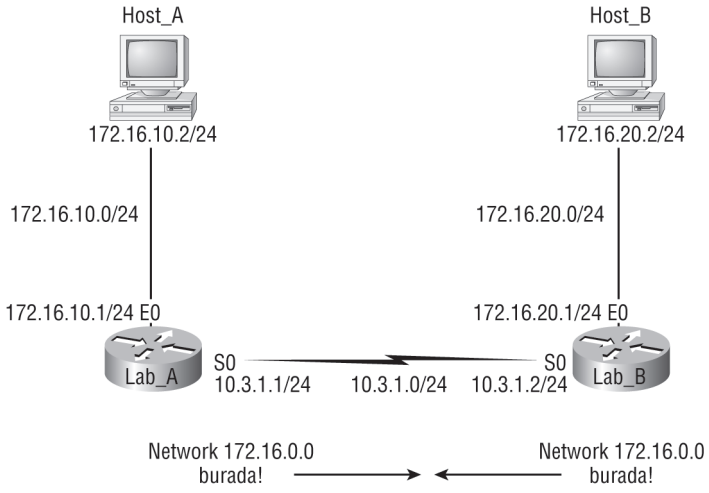
Çok karmaşık classless routing protokollerinden biri olarak EIGRP, Variable Length Subnet Mask'ların kullanımını destekler. Point-to-point network'ler için 30-bit subnet maskları kullanmak gibi, host ihtiyacına en yakın subnet masklarının kullanımıyla, adres uzayını korumaya izin verdiğinden, gerçekten çok önemlidir. Ve subnet maskın, her route güncellemesi ile gönderilmesinden dolayı, EIGRP, ardışık olmayan network'lerin kullanımını da destekler. Bu bize, network''ün IP adres planı tasarlandığında, oldukça esneklik sağlar.

Ardışık olmayan (discontiguous) network'ler nedir? Birbirlerine farklı classful network'lerle bağlı, classful bir network'ün iki yada daha çok sayıda alt network'üdür. Şekil 7.1, tipik bir discontigiuous network'ü göstermektedir. Varsayılan olarak, her router, sadece 172.16.0.0 classful network'ünü bekler.




***Şekil 7.1:*** *Bir discontigiuous network.*

EIGRP ayrıca tüm EIGRP router'larda, route tablolarının boyutunu büyük miktarda düşürebilen, manuel summary'lerin oluşturulmasını da destekler. Bununla beraber, EIGRP, classful sınırlardaki network'leri otomatik olarak summarize eder. Şekil 7.2, EIGRP çalışan bir router'ın, network'ü ve auto-summarize yapacağı sınırları nasıl anlayacağını göstermektedir.




***Şekil 7.2:*** *EIGRP auto-summarization.*

Açıkçası, bu varsayılan olarak asla çalışmaz! Şunu unutmayın ki RIPv1, RIPv2 ve IGRP de, varsayılan olarak aynı classful sınırları auto-summarize edecektir, fakat OSPF etmeyecektir.

*RIPv2 ve EIGRP, discontiguous ağ kurulumunu destekler, fakat varsayılanda değil. RIPv2 ve EIGRP'nin yaptığı gibi classful sınırları otomatik summarize etmediğinden OSPF, varsayılan olarak discontiguous ağ kurulumunu destekler.*

NOT

## Route Tespiti ve Onarımı

EIGRP'nin karma yapısı, route tespiti ve route onarımındaki yaklaşımlarında tamamıyla ortaya konur. Birçok link-state protokollünde olduğu gibi EIGRP, komşuların Hello prosesi ve durumlarının görüntülenmesi yardımıyla tespit edilmesi kavramını destekler. Birçok distance-vector routing protokolü gibi EIGRP, birçok router'ın, bir route güncellemesini ilk elden asla öğrenmediği, daha önce belirttiğim bir kulaktan duyma routing mekanizması kullanır. Onu, başka birinden öğrenmiş olan diğer bir router'dan öğrenirler.

EIGRP router'ların toplamak zorunda oldukları çok büyük miktarda bilginin, bir yere yerleştirilmesi gerekir, değil mi? EIGRP'nin, çevresindekilerle ilgili önemli bilgileri tuttuğu tablolar vardır:

**Neighborship table:** Neighborship tablosu (genellikle bir neighbor tablosu olarak bilinir), komşuluk ilişkisi kurulan router'lar hakkındaki bilgileri kaydeder.

**Topoloji table:** Topoloji tablosu tüm komşulardan alınan, ağ topluluğundaki her route hakkındaki route yayınlarını tutar.

**Route table:** Route tablosu, routing kararları vermek için halihazırda kullanılan route'ları tutar. EIGRP tarafından her protokol için bu tabloların ayrı tutulması ve , IPv6 kullanımı aktif olarak desteklenir.

Şimdi, EIGRP metriklerinden bahsedeceğim ve daha sonra basit EIGRP konfigürasyonuna geçeceğim.

### EIGRP Metrikleri

EIGRP ile ilgili diğer bir hoş özellik, en iyi yolu seçmek ve route'ları mukayese etmek için tek bir faktör kullanan diğer protokollerin aksine, EIGRP'nin dört faktörün kombinasyonunu kullanabilmesidir:

- Bant genişliği
- Gecikme
- Yük
- Güvenilirlik

IGRP gibi EIGRP'de varsayılan olarak, uzak ağlara en iyi yolu belirlemek için bir hattın bant genişliğini ve gecikmesini kullanır. Cisco bazen bunu, yol bant genişliği değeri ve toplam hat gecikmesi olarak belirtir.

Maximum transmission unit (MTU) boyutu olan beşinci unsurun bir kıymeti yoktur. Bu unsur, EIGRP hesaplamasında asla kullanılmaz, fakat redistribution'u içeren bazı EIGRP bağlantılı komutlarda gerekli bir parametredir. MTU unsurunun değeri, hedef network'e giden yol boyunca karşılaşılan, en küçük MTU değerini belirtir.

### Maksimum Yollar ve Hop Sayısı

Varsayılan olarak, EIGRP, dört linke kadar equal-cost yük paylaşımını destekler (aslında, tüm routing protokolleri bunu yaparlar). Ayrıca, aşağıdaki komutu kullanarak, EIGRP ile altı link ile yük-dengelemesi yapabilirsiniz:

```
Pod1R1(config)#router eigrp 10
Pod1R1(config-router)#maximum-paths ?
  <1-6>  Number of paths
```

İlave olarak EIGRP, 100 maksimum hop sayısına sahiptir, fakat 255'e kadar ayarlanabilir. Yol metrik hesaplamasında hop sayısı kullanmamasına rağmen, AS kapsamını sınırlandırmak için hala maksimum hop sayısını kullanır.

*Varsayılan olarak, tüm routing protokolleri, dört eşit cost değerine sahip linkte yük-dengelemesi yapabilir. Ayrıca, EIGRP size, altı linkte yük-dengelemesine izin verir ve variance komutu yardımıyla, altı eşit olmayan cost'a sahip linkte yük-dengelemesi yapabilir.*

NOT

# EIGRP Konfigürasyonu

EIGRP, IP, IPv6, IPX ve AppleTalk için yapılandırılabildiği halde, geleceğin CCNA'yi olarak, aslında şimdi sadece IP konfigürasyonuna odaklanmanız gerekmektedir. (IPv6 konfigürasyonları, Modül13'te gösterilecektir.)

EIGRP komutlarının girildiği iki mod vardır: router configuration mod ve interface configuration mod. Router configuration mod, protokollere izin verir, hangi network'lerin EIGRP çalışacağını belirler ve global özellikleri ayarlar. Interface konfigürasyon modu, summary, metrik, timer ve bant genişliğinin uyarlanmasını sağlar.

Bir router'da EIGRP oturumu başlatmak için, network'ünüzün autonomous system numarasını ekleyerek router eigrp komutunu kullanın. Sonra, network adreslerinizi eklemek için network komutunu kullanıp router'a bağlı network adreslerini girin.

10.3.1.0/24 ve 172.16.10.0/24 network adresine sahip iki ağa bağlı bir router'da, 20 no'lu autonomous system için, EIGRP'ye izin veren bir örneğe bakalım:

```
Router#config t
Router(config)#router eigrp 20
Router(config-router)#network 172.16.0.0
Router(config-router)#network 10.0.0.0
```

RIP'de olduğu gibi hiçbir subnet ve host bit'inin kullanılmadığı classful network adresi kullandığınızı hatırlayın.

*AS numarasının önemsiz olduğunu anlayın. Tüm router'lar aynı numarayı kullandığı müddetçe bu böyledir. 1 ile 65,535 arasında herhangi bir numara kullanabilirsiniz.*

NOT

BRI interface ya da seri interface'in internete çıkması gibi durumlarda belirli bir interface'de EIGRP çalışmasını durdurmanız gerekebilir. Bunu yapmak için bölüm 6'da RIP konusunda bahsettiğimiz gibi, `passive-interface interface` komutunu kullanarak interface'i pasif olarak etiketleyebilirsiniz. Aşağıdaki komut, size serial0/1 interface'inin nasıl pasif yapıldığını göstermektedir:

```
Router(config)#router eigrp 20
Router(config-router)#passive-interface serial 0/1
```

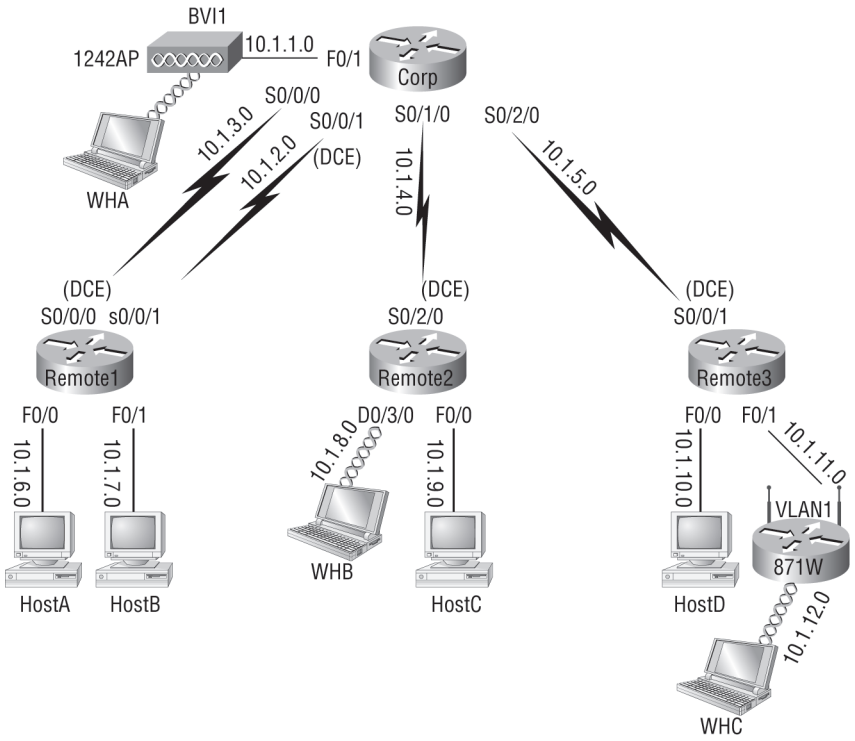
Bunu yapmak, interface'in Hello paketleri gönderip almasını engelleyecektir. Sonuç olarak, onun komşuluk kurmasını durduracaktır. Yani, router bu interface'inden route bilgisi alıp göndereme-yecektir.

*passive-interface komutunun etkisi, komutun çalıştırıldığı routing protokolüne bağlıdır. Örneğin, RIP çalışan bir interface'te, passive-interface komutu, route güncellemelerinin gönderilmesini durduracaktır, fakat alınmasına izin verecektir. Böylece, pasif interface'i olan bir RIP router, diğer router'lar tarafından yayınlanan network'ler hakkındaki bilgileri hala öğrenecektir. Bu, passive-interface komutu ile güncellemelerin gönderilip alınmayacağı, EIGRP davranışından farklıdır.*

NOT

Şimdi, bundan önceki modülde, RIP ve RIPv2 kullandığımız network'ü yapılandıralım. EIGRP'nin AD'si 90 olduğundan, bant genişliği kullanımı ve CPU dalgalanmalarını sorun etmediğiniz sürece, RIPv2'nin (statik route'larımızın da) hala çalışması önemli değildir. Statik route'larımızın AD'sinin 150/151 olarak değiştirildiğini, RIP'in AD'sinin 120 olduğunu hatırlayın, bu nedenle, RIP ve statik route etkin olsa dahi, sadece EIGRP route'ları routing tablosunda yer alacaktır.

Şekil 7.3, çalıştığımız network'ü göstermektedir. (EIGRP ile yapılandırmak için kullanacağımızla aynıdır.)




***Şekil 7.3:*** *Ağ topluluğumuz.*

Bir anımsatıcı olarak kullanabilmeniz için Tablo 7.1, her interface için kullanacağımız IP adreslerini içermektedir.

**Tablo 7.1:** IP Network'ü için Network Adreslemesi

| Router | Network Adresi | Interface | Adres |
|---|---|---|---|
| **Corp** | | | |
| Corp | 10.1.1.0 | F0/1 | 10.1.1.1 |
| Corp | 10.1.2.0 | S0/0/0 | 10.1.2.1 |
| Corp | 10.1.3.0 | S0/0/1(DCE) | 10.1.3.1 |
| Corp | 10.1.4.0 | s0/1/0 | 10.1.4.1 |
| Corp | 10.1.5.0 | s0/2/0 | 10.1.5.1 |
| **R1** | | | |
| R1 | 10.1.2.0 | S0/0/0 (DCE) | 10.1.2.2 |
| R1 | 10.1.3.0 | S0/0/1 | 10.1.3.2 |
| R1 | 10.1.6.0 | F0/0 | 10.1.6.1 |
| R1 | 10.1.7.0 | F0/1 | 10.1.7.1 |
| **R2** | | | |
| R2 | 10.1.4.0 | S0/2/0 (DCE) | 10.1.4.2 |
| R2 | 10.1.8.0 | D0/3/0 | 10.1.8.1 |
| R2 | 10.1.9.0 | F0/0 | 10.1.9.1 |
| **R3** | | | |
| R3 | 10.1.5.0 | S0/0/0/ (DCE) | 10.1.5.2 |
| R3 | 10.1.10.0 | F0/0 | 10.1.10.1 |
| R3 | 10.1.11.0 | F0/1 | 10.1.11.1 |
| **871W** | | | |
| 871W | 10.1.11.0 | Vlan 1 | 10.1.11.2 |
| 871W | 10.1.12.0 | Dot11radio0 | 10.1.12.1 |
| **1242 AP** | | | |
| 1242 AP | 10.1.1.0 | BVI 1 | 10.1.1.2 |

EIGRP'yi ağ topluluğumuza eklemek gerçekten kolaydır. Bu EIGRP'nin güzel bir yönüdür.

## Corp

Aşağıdaki router çıktısında göründüğü gibi AS numarası 1'den 65,535'e kadar herhangi bir numara olabilir. Bir router, olmasını istediğiniz sayıda AS'e dahil olabilir. Fakat biz bu kitabın amaçları doğrultusunda sadece tek bir AS yapılandıracağız:

```
Corp#config t
Corp(config)#router eigrp ?
  <1-65535>  Autonomous system number

Corp(config)#router eigrp 10
Corp(config-router)#network 10.0.0.0
```

`router eigrp [as]` komutu, router'daki EIGRP routing'i etkinleştirir. RIPv1'de olduğu gibi hala, yayınlamayı istediğiniz classful network adreslerini eklemeniz gerekir. Fakat RIP'in tersine EIGRP, classless routing kullanır. Fakat hala classful gibi yapılandırırsınız. Hatırladığınıza eminim ki, classless, subnet mask bilgisinin, routing protokol güncellemeleri ile beraber gönderilmesi anlamına gelmektedir. (RIPv2, classless'tır.)

## R1

R1 router'ını yapılandırmak için tüm yapmanız gereken, AS 10 kullanarak EIGRP routing'i etkinleştirmek ve sonra network adreslerini şu şekilde eklemektir:

```
R1#config t
R1(config)#router eigrp 10
R1(config-router)#network 10.0.0.0
*Mar 21 19:18:12.935: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 10: Neighbor
   10.1.2.2 (Serial0/0/0) is up: new adjacency
*Mar 21 19:18:12.935: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 10: Neighbor
   10.1.3.2 (Serial0/0/1) is up: new adjacency
```

R1 router'ı Corp komşusunu buldu. İki router, adjacent'dir. Router'lar arasındaki iki linki de bulduğuna dikkat edin. Bu güzel bir şeydir.

## R2

R2 router'ını yapılandırmak için tüm yapacağınız, yine AS 10'u kullanarak EIGRP'yi etkinleştirmektir:

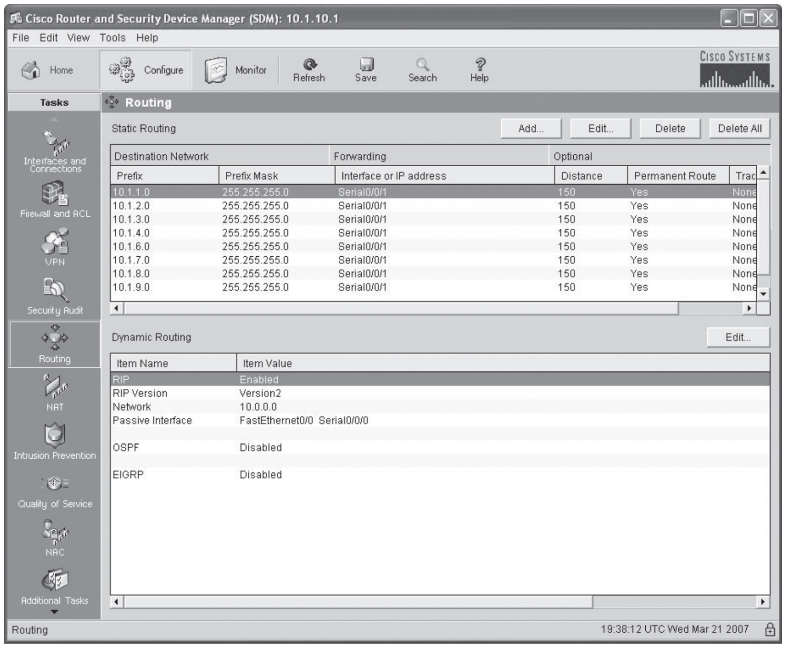
```
R2#config t
R2(config)#router eigrp 10
R2(config-router)#network 10.0.0.0
*Mar 21 19:20:29.023: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 10: Neighbor
   10.1.4.2 (Serial0/1/0) is up: new adjacency
```

Hepsi bu, gerçekten! Birçok routing protokolünün kurulumu oldukça basittir. EIGRP de bunlara dahildir. Tabii ki, bu sadece basit konfigürasyonlar için geçerlidir.

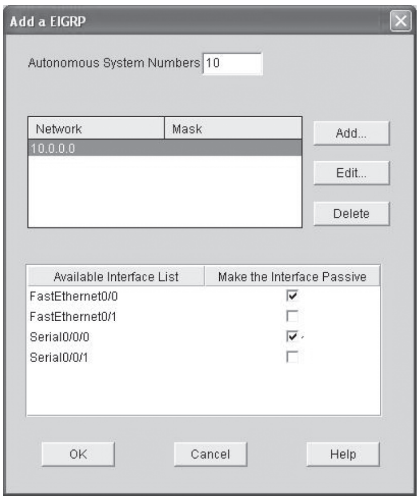
## R3

Gelin son birkaç modülde yaptığımız gibi, EIGRP konfigürasyonu için SDM kullanalım. Konfigürasyon prosesinin kendisi, fazla zaman almayacaktır. İlk olarak bağlanmak için yaptığım kısım, oldukça zaman almaktadır.

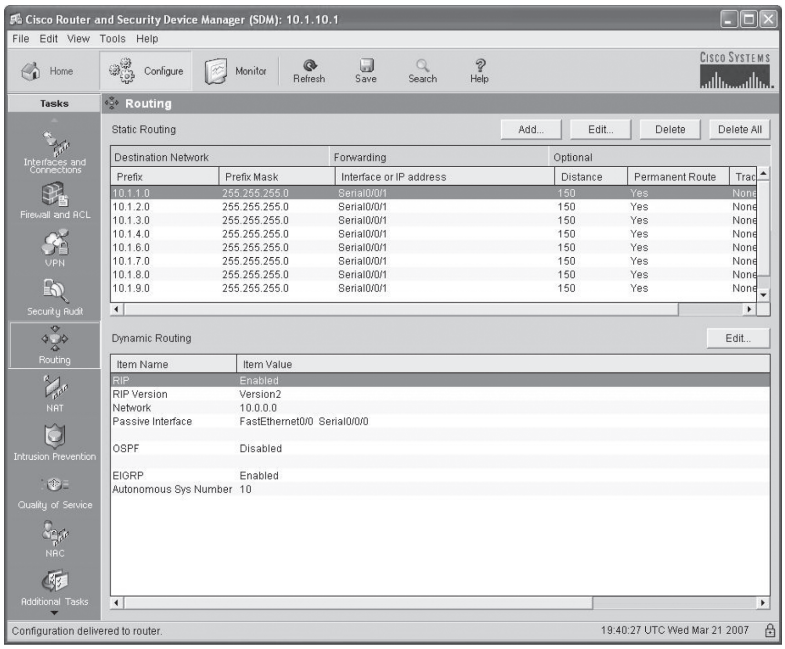
Bu ilk şekle bakarak, router'ımızda hala, hem RIPv2 hem de statik route'ların çalıştığını görebiliriz:




AS 10'u ekleyerek ve pasif interface'lerimizi ayarlamak için seçerek EIGRP'yi etkinleştirelim. (Sadece böyle yapılması çok kolay olduğundan!)




Son olarak, EIGRP'nin şimdi, AS 10 ile çalıştığını görebiliriz.




Hepsi bu, bitirdik bile.

Konfigürasyonumuz güvenilir görünüyor, fakat en düşük AD'ye sahip olduğundan, routing tablosunda sadece EIGRP'nin yer alacağını unutmayın. Arka planda RIP çalıştırarak, router'da sadece fazla hafıza kullanımı ve CPU dalgalanmalarına sahip olmadık aynı zamanda linklerimizin her birindeki değerli bant genişliğini de harcıyoruz. Bu gerçekten kötü ve unutmamanız gereken bir konudur.

Şimdi son router'ımızı yapılandıracağız. 871W router'ımın sahip olduğu IOS imajı, EIGRP'yi desteklemediğinden, RIPv2 çalışmasına devam edeceğiz. R3 router'ından, 871W router'ına redistribution (çeviri) prosesi yapılandıracağım.

## 871W Router'ını R3'ten Redistribute Etmek

Router R3'te, EIGRP ve RIP altında `redistribution` komutları eklememiz gerekti. SDM'in kısıtlamaları olduğundan, bunu CLI kullanarak yapmamız gerekir:

```
R3#config t
R3(config)#router eigrp 10
R3(config-router)#redistribute rip ?
  metric     Metric for redistributed routes
  route-map  Route map reference
  <cr>
R3(config-router)#redistribute rip metric ?
  <1-4294967295>  Bandwidth metric in Kbits per second
R3(config-router)#redistribute rip metric 10000000 ?
  <0-4294967295>  EIGRP delay metric, in 10 microsecond units
R3(config-router)#redistribute rip metric 10000000 20000 ?
  <0-255>  EIGRP reliability metric where 255 is 100% reliable
R3(config-router)#redistribute rip metric 10000000 20000 255 ?
  <1-255>  EIGRP Effective bandwidth metric (Loading) where 255
is 100% loaded
R3(config-router)#redistribute rip metric 10000000 20000 255 1 ?
```

```
  <1-65535>  EIGRP MTU of the path
R3(config-router)#redistribute rip metric 10000000 20000 255 1
1500
R3(config-router)#do show run | begin router eigrp 10
router eigrp 10
 redistribute rip metric 10000000 20000 255 1 1500
 passive-interface FastEthernet0/0
 passive-interface Serial0/0/0
 network 10.0.0.0
 no auto-summary
!
```

RIP'in hop sayısı metriğini, EIGRP'nin bant genişliği, gecikme, güvenilirlik, yük ve MTU metriği ile eşleşmesi için değiştirmemiz gerekti. Varsayılan olarak, EIGRP sadece hattın gecikmesi ve bant genişliğini kullansa da, redistribution kullandığınızda tüm metrik değerlerini kullanmak zorundasınız.

R3 router'ımız şimdi çift lisanlıdır; hem RIP hem de EIGRP konuşmaktadır. Diğer router'larımız için bir çeşit tercümanlık dahi yapabilecektir. Fakat henüz tamamlamadık. EIGRP'den RIP'e redistribution konfigürasyonuna da gerek vardır (karşılıklı redistribution). Böylece 871W router, sadece RIP konuştuğundan EIGRP route'larını RIP route'ları gibi alacaktır:

```
R3(config)#router rip
R3(config-router)#redistribute eigrp 10 ?
  metric     Metric for redistributed routes
  route-map  Route map reference
  <cr>
R3(config-router)#redistribute eigrp 10 metric ?
  <0-16>       Default metric
  transparent  Transparently redistribute metric

R3(config-router)#redistribute eigrp 10 metric 1
```

Yukarıdaki çıktı EIGRP'yi RIP'e redistribute ettiğimizi ve metriği hop sayısıyla değiştirdiğimizi gösterir.

Bunun gerçekten çalıştığını anlamanın tek yolu, R3 haricinde diğer router'larda RIP'in devre dışı bırakılmasıdır. Bu, 871W router'ına ve 871W router'ından dönüşüm sağlar. Nasıl yaptığımı aşağıda görebilirsiniz:

```
Corp#config t
Corp(config)#no router rip
R1#config t
R1(config)#no router rip
R2#config t
R2(config)#no router rip
```

Corp'un routing tablosunu kontrol edelim:

```
Corp#sh ip route
     10.0.0.0/24 is subnetted, 12 subnets
```

```
D       10.1.11.0 [90/2172416] via 10.1.5.2, 00:04:57, Serial0/2/0
D       10.1.10.0 [90/2172416] via 10.1.5.2, 00:04:57, Serial0/2/0
D       10.1.9.0 [90/2195456] via 10.1.4.2, 00:04:57, Serial0/1/0
D       10.1.8.0 [90/2195456] via 10.1.4.2, 00:04:57, Serial0/1/0
D       10.1.12.0 [90/2172416] via 10.1.5.2, 00:03:00, Serial0/2/0
C       10.1.3.0 is directly connected, Serial0/0/1
C       10.1.2.0 is directly connected, Serial0/0/0
C       10.1.1.0 is directly connected, FastEthernet0/1
D       10.1.7.0 [90/2195456] via 10.1.3.2, 00:04:58, Serial0/0/1
                 [90/2195456] via 10.1.2.2, 00:04:58, Serial0/0/0
D       10.1.6.0 [90/2195456] via 10.1.3.2, 00:04:58, Serial0/0/1
                 [90/2195456] via 10.1.2.2, 00:04:58, Serial0/0/0
C       10.1.5.0 is directly connected, Serial0/2/0
C       10.1.4.0 is directly connected, Serial0/1/0
```

Güzel, 871W router'a bağlı kablosuz 10.1.12.0 LAN'ı da dahil, tüm route'lar görünmektedir. Sadece o değil, R3, bizim için RIP'ten EIGRP'ye tercüme ettiğinden, bir EIGRP network'ü gibi görünmektedir. Aşağıda, tüm route'ları R3 router'ından RIP route gibi alan, 871W router'ının routing tablosunu görebilirsiniz:

```
871W#sh ip route
     10.0.0.0/24 is subnetted, 12 subnets
C       10.1.11.0 is directly connected, Vlan1
R       10.1.10.0 [120/1] via 10.1.11.1, 00:00:19, Vlan1
R       10.1.9.0 [120/2] via 10.1.11.1, 00:00:19, Vlan1
R       10.1.8.0 [120/2] via 10.1.11.1, 00:00:19, Vlan1
C       10.1.12.0 is directly connected, Dot11Radio0
R       10.1.3.0 [120/2] via 10.1.11.1, 00:00:19, Vlan1
R       10.1.2.0 [120/2] via 10.1.11.1, 00:00:19, Vlan1
R       10.1.1.0 [120/2] via 10.1.11.1, 00:00:19, Vlan1
R       10.1.7.0 [120/2] via 10.1.11.1, 00:00:19, Vlan1
R       10.1.6.0 [120/2] via 10.1.11.1, 00:00:19, Vlan1
R       10.1.5.0 [120/1] via 10.1.11.1, 00:00:19, Vlan1
R       10.1.4.0 [120/2] via 10.1.11.1, 00:00:19, Vlan1
```

Routing tablosunda, tüm network'leri görebiliriz. 871W router'ı tüm network'ler için RIP olarak bilgileri ediniyor. Oldukça güzel! Şayet RIP çalışan eski bir router'ınız varsa ve diğer router'larınızda RIP kurmak istemiyorsanız, bu bir network'ü yapılandırma yolu olarak oldukça iyi bir örnektir.

## Discontgiuous Network'ler

Auto-summarization hakkında bilmeniz gereken bir konfigürasyon daha var. Şekil 7.1'i ve EIGRP'nin bitişik olmayan bir ağda sınırları nasıl otomatik summarize ettiğini gösterdiğini hatırlıyor musunuz? Şekle tekrar bakın. İki router'ı da EIGRP ile yapılandıracağım.

Şekilde, Lab_A, 172.16.10.0/24 ağına ve 10.3.1.0/24 omurgasına bağlıdır. Lab_B, 172.16.20.0/24 ağına ve 10.3.1.0/24 omurgasına bağlıdır. Varsayılan olarak, iki router'da classful sınırları summarize edecektir ve routing çalışmayacaktır. Aşağıda, bu network'ü çalıştıran konfigürasyonu bulabilirsiniz:

```
Lab_A#config t
Lab_A(config)#router eigrp 100
Lab_A(config-router)#network 172.16.0.0
Lab_A(config-router)#network 10.0.0.0
Lab_A(config-router)#no auto-summary

Lab_B#config t
Lab_B(config)#router eigrp 100
Lab_B(config-router)#network 172.16.0.0
Lab_B(config-router)#network 10.0.0.0
Lab_B(config-router)#no auto-summary
```

No `auto-summary` komutunu kullandığımdan, EIGRP, iki router arasında, tüm subnetleri yayınlayacaktır. Network'ler daha geniş olsaydı, aynı sınırlarda manuel summarization yapabilirdiniz.

NOT

*R3 router'ını, SDM kullanarak yapılandırdığımda, EIGRP altında no auto-summary'yi otomatik olarak ekledi. İsteyip istemediğimi sormadı bile ve ben sadece CLI'dan doğrulayabilir yada etkisiz hale getirebilirim.*

# EIGRP ile Yük Dengelemesi

Varsayılan olarak, EIGRP'nin dört eşit cost değerli linke kadar yük-dengelemesi yapabileceğini biliyorsunuzdur. Fakat EIGRP'yi uzak bir network için altı eşit/eşit olmayan cost değerli linklerde yük dengelemesi yapmak için yapılandırabileceğimizi hatırladınız mı? Evet, yapabiliyoruz, yük dengelemesi yapmak için Corp ve R1 router'larıyla uğraşalım. İlk olarak, R1'in routing tablosuna bakalım ve EIGRP'nin, router'lar arasında her iki linki de bulduğundan emin olalım:

```
R1#sh ip route
     10.0.0.0/24 is subnetted, 12 subnets
D       10.1.11.0 [90/2684416] via 10.1.3.1, 00:50:37, Serial0/0/1
                  [90/2684416] via 10.1.2.1, 00:50:37, Serial0/0/0
D       10.1.10.0 [90/2707456] via 10.1.3.1, 01:04:40, Serial0/0/1
                  [90/2707456] via 10.1.2.1, 01:04:40, Serial0/0/0
D       10.1.9.0 [90/2707456] via 10.1.3.1, 01:24:09, Serial0/0/1
                 [90/2707456] via 10.1.2.1, 01:24:09, Serial0/0/0
D       10.1.8.0 [90/2707456] via 10.1.3.1, 01:24:09, Serial0/0/1
                 [90/2707456] via 10.1.2.1, 01:24:09, Serial0/0/0
D       10.1.12.0 [90/2684416] via 10.1.3.1, 00:10:10, Serial0/0/1
                  [90/2684416] via 10.1.2.1, 00:10:10, Serial0/0/0
C       10.1.3.0 is directly connected, Serial0/0/1
C       10.1.2.0 is directly connected, Serial0/0/0
D       10.1.1.0 [90/2172416] via 10.1.3.1, 01:24:11, Serial0/0/1
                 [90/2172416] via 10.1.2.1, 01:24:11, Serial0/0/0
C       10.1.7.0 is directly connected, FastEthernet0/1
C       10.1.6.0 is directly connected, FastEthernet0/0
D       10.1.5.0 [90/2681856] via 10.1.3.1, 01:24:11, Serial0/0/1
                 [90/2681856] via 10.1.2.1, 01:24:11, Serial0/0/0
D       10.1.4.0 [90/2681856] via 10.1.3.1, 01:24:11, Serial0/0/1
                 [90/2681856] via 10.1.2.1, 01:24:11, Serial0/0/0
```

Bu yeni ve farklı, gerçekten çok ilginç bir routing tablosudur. Ağ topluluğumuzdaki her route için iki linkimiz olduğunu görebilirsiniz ve aynı metriklere sahip olduklarından, EIGRP'nin s0/0/0 ve s0/0/1 linklerinde varsayılan olarak yük dengelemesi yapacaktır.

EIGRP gerçekten bazı güzel özellikler önerir ve bunlardan biri, otomatik yük dengelemesidir. Fakat linklerin demetlenmesi (bundling) için durum nasıldır? EIGRP, ilave konfigürasyon olmadan bunu yapmamızı da sağlar. Bunun nasıl çalıştığını size göstereceğim. Corp ve R1 router'larımız arasındaki linkleri, aynı subnette yapılandıracağım. Yani, iki linkin tüm interface'leri, aynı subnette olacaklar. Konfigürasyonumu kontrol edin:

```
Corp#config t
Corp(config)#int s0/0/1
Corp(config-if)#ip address 10.1.2.4 255.255.255.0

R1#config t
R1(config)#int s0/0/1
R1(config-if)#ip address 10.1.2.3 255.255.255.0
R1(config-if)#do show run | begin interface
interface Serial0/0/0
 description 1st Connection to Corp Router
 ip address 10.1.2.2 255.255.255.0
!
interface Serial0/0/1
 description 2nd connection to Corp Router
 ip address 10.1.2.3 255.255.255.0
```

Şimdi, her iki linkin dört interface'i de aynı subnettedir.

```
R1(config-if)#do show ip route
     10.0.0.0/24 is subnetted, 12 subnets
D       10.1.11.0 [90/2684416] via 10.1.2.4, 00:04:44, Serial0/0/1
                  [90/2684416] via 10.1.2.1, 00:04:44, Serial0/0/0
D       10.1.10.0 [90/2707456] via 10.1.2.4, 00:04:44, Serial0/0/1
                  [90/2707456] via 10.1.2.1, 00:04:44, Serial0/0/0
D       10.1.9.0 [90/2707456] via 10.1.2.4, 00:04:44, Serial0/0/1
                 [90/2707456] via 10.1.2.1, 00:04:44, Serial0/0/0
D       10.1.8.0 [90/2707456] via 10.1.2.4, 00:04:44, Serial0/0/1
                 [90/2707456] via 10.1.2.1, 00:04:44, Serial0/0/0
D       10.1.12.0 [90/2684416] via 10.1.2.4, 00:04:44, Serial0/0/1
                  [90/2684416] via 10.1.2.1, 00:04:44, Serial0/0/0
D       10.1.3.0 [90/3193856] via 10.1.2.4, 00:04:44, Serial0/0/1
                 [90/3193856] via 10.1.2.1, 00:04:44, Serial0/0/0
C       10.1.2.0 is directly connected, Serial0/0/0
                 is directly connected, Serial0/0/1
D       10.1.1.0 [90/2172416] via 10.1.2.4, 00:03:56, Serial0/0/1
                 [90/2172416] via 10.1.2.1, 00:03:56, Serial0/0/0
C       10.1.7.0 is directly connected, FastEthernet0/1
C       10.1.6.0 is directly connected, FastEthernet0/0
```

```
D       10.1.5.0 [90/2681856] via 10.1.2.4, 00:04:46, Serial0/0/1
                 [90/2681856] via 10.1.2.1, 00:04:46, Serial0/0/0
D       10.1.4.0 [90/2681856] via 10.1.2.4, 00:04:46, Serial0/0/1
                 [90/2681856] via 10.1.2.1, 00:04:46, Serial0/0/0
```

Routing tablosunda, bir veya iki kurnazca yapılan değişikliği fark ettiniz mi? Ayrı ayrı görünen 10.1.2.0 ve 10.1.3.0 network'leri, direkt bağlı interface'lerdi, fakat şimdi değiller. Şimdi, sadece 10.1.2.0 network'ü, iki direkt bağlı interface gibi görünmektedir ve router şimdi, iki 1.5Mbps T1 linki yerine 3 Mbit bir kanala sahiptir. Bu değişikliklerin kurnazca yapılması, onların daha iyi olmasını engellemez!

NOT *Bu mükemmel konfigürasyonu çalışır hale getirmek için, EIGRP'nin ilk olarak etkinleştirilmesi gerekir. Yoksa router'ınızda adreslerin çakıştığı ile ilgili bir uyarı alırsınız.*

Fakat burada biraz duralım. 10.1.3.0, routing tablosunda hala neden duruyor ve 10.1.3.0 subneti, var olmamasına rağmen, neden, EIGRP için DUAL anlamına gelen ***D*** olarak görünmektedir? Cevap oldukça basit, R3 router'ında, SDM yardımıyla statik route'lar yapılandırdığımızda, Permanent route seçeneğine tıkladım. Bu komutun etkisi, "şayet herhangi bir statik network giderse, bu route'u, R3'ün routing tablosunda saklamaya devam et"'dir. 10.1.3.0 network'ü, Corp ve R1 router'ları arasında yapılandırılmamıştı, bu nedenle, gerçekte olmamasına rağmen, R3 router'ı, 10.1.3.0'ı kullanılabilir gibi yayınlamaktadır. Bunu, `redistribution` komutumuzu kullandığımız için yapmaktadır.

10.1.3.0 subnetini tekrar network'lere ekleyeceğim. Böylece, bu ikili linklerle biraz eğlenebiliriz. Corp ve R1 s0/0/1 interface'lerine gideceğim ve 10.1.3.1/24 ve 10.1.3.2/24'ü yapılandıracağım. Şimdi, 10.1.3.0 yeniden yayınlanıyor olacaktır, fakat bu sefer, network, gerçekten mevcuttur. Şimdi, bazı parametreleri karıştıralım, 10.1.3.0 linkinin metriğini değiştirelim ve ne olduğuna bakalım:

NOT *Bu, statik route'larla permanent seçeneğini neden kullanmamamız gerektiğinin gerçek sebebidir. Çünkü bunu yaparsanız, routing tablonuz, mevcut olmayan bir subneti bile gösterebilecektir!*

```
R1#config t
R1(config)#int s0/0/1
R1(config-if)#bandwidth 256000
R1(config-if)#delay 300000
Corp#config t
Corp(config)#int s0/0/1
Corp(config-if)#bandwidth 256000
Corp(config-if)#delay 300000
```

Varsayılan olarak, her network'e giden en iyi yolu belirlemek için, hattın bant genişliği ve gecikmesini kullandığından, R1 ve Corp router'larının s0/0/1 interface'lerinin gecikmesini artırdım ve bant genişliğini düşürdüm. Şimdi, network'ümüzdeki EIGRP'yi kontrol edelim, artı, R1 ve Corp router'ları arasındaki çiftli linklerin ne olduğuna bakalım.

# EIGRP'nin Doğruluğunu Kontrol Etmek

EIGRP konfigürasyonlarının doğruluğunu kontrol etmesi ve hata tespitinde size yardımcı olması için bir router'da kullanılabilecek bazı komutlar vardır. Tablo 7.2, EIGRP operasyonlarını doğrulamak ve her komutun yaptığının özet açıklaması ile birlikte, kullanılan komutların en önemlilerini içermektedir.

**Tablo 7.2:** EIGRP Hata Tespiti Komutları

| Komut | Açiklama/Fonksiyon |
|---|---|
| `show ip route` | Routing table'ın tamamını gösterir |
| `show ip route eigrp` | Sadece, routing tablosundaki EIGRP kayıtlarını gösterir. |
| `show ip eigrp neighbors` | EIGRP komşularının hepsini gösterir. |
| `show ip eigrp topology` | EIGRP topoloji tablosundaki kayıtları gösterirler. |
| `debug eigrp packet` | Komşu router'lar arasında alınıp/gönderilen Hello paketlerini gösterir. |
| `Debug ip eigrp notification` | EIGRP değişikliklerini ve network'ünüzde olan güncellemeleri gösterir. |

Yapılandırdığımız ağ topluluğumuzu da kullanarak, Tablo 7.2'deki komutları nasıl kullanacağınızı göstereceğim (ardışık olmayan network örneğini içermeyecektir).

Aşağıdaki router çıktısı, örneğimizdeki Corp router'ından alınmıştır:

```
Corp#sh ip route
     10.0.0.0/24 is subnetted, 12 subnets
D       10.1.11.0 [90/2172416] via 10.1.5.2, 00:01:05, Serial0/2/0
D       10.1.10.0 [90/2195456] via 10.1.5.2, 00:01:05, Serial0/2/0
D       10.1.9.0 [90/2195456] via 10.1.4.2, 00:01:05, Serial0/1/0
D       10.1.8.0 [90/2195456] via 10.1.4.2, 00:01:05, Serial0/1/0
D       10.1.12.0 [90/2172416] via 10.1.5.2, 00:01:05, Serial0/2/0
C       10.1.3.0 is directly connected, Serial0/0/1
C       10.1.2.0 is directly connected, Serial0/0/0
C       10.1.1.0 is directly connected, FastEthernet0/1
D       10.1.7.0 [90/2195456] via 10.1.2.2, 00:01:06, Serial0/0/0
D       10.1.6.0 [90/2195456] via 10.1.2.2, 00:01:06, Serial0/0/0
C       10.1.5.0 is directly connected, Serial0/2/0
C       10.1.4.0 is directly connected, Serial0/1/0
```

Tüm route'ların, routing tablosunda olduğunu (10.1.3.0, tekrar direk bağlı görünmektedir) ve 10.1.6.0 ile 10.1.7.0 network'lerine sadece bir link olduğunu görebilirsiniz. EIGRP route'larının, basitçe ***D*** (DUAL) ile belirtilmekte olduğuna ve bu route'un varsayılan AD'sinin 90 olduğuna dikkat edin. Bu, internal EIGRP route'ları olduğunu gösterir. Şimdi, metriklerini değiştirdiğimiz R1 routing tablosuna bir bakalım:

```
R1#sh ip route
     10.0.0.0/24 is subnetted, 12 subnets
D       10.1.11.0 [90/2684416] via 10.1.2.1, 00:00:09, Serial0/0/0
D       10.1.10.0 [90/2707456] via 10.1.2.1, 00:00:09, Serial0/0/0
D       10.1.9.0 [90/2707456] via 10.1.2.1, 00:00:09, Serial0/0/0
D       10.1.8.0 [90/2707456] via 10.1.2.1, 00:00:09, Serial0/0/0
D       10.1.12.0 [90/2684416] via 10.1.2.1, 00:00:09, Serial0/0/0
C       10.1.3.0 is directly connected, Serial0/0/1
C       10.1.2.0 is directly connected, Serial0/0/0
D       10.1.1.0 [90/2172416] via 10.1.2.1, 00:00:09, Serial0/0/0
C       10.1.7.0 is directly connected, FastEthernet0/1
```

```
C       10.1.6.0 is directly connected, FastEthernet0/0
D       10.1.5.0 [90/2681856] via 10.1.2.1, 00:00:09, Serial0/0/0
D       10.1.4.0 [90/2681856] via 10.1.2.1, 00:00:09, Serial0/0/0
```

Şimdi, her uzak network'e sadece bir route'a sahibiz ve 10.1.3.0 network'ü bizim yedek linkimizdir. Açıkçası, aynı anda her iki linkin kullanılması daha iyidir, fakat örneğimde 10.1.3.0 network'ünü yedek bir link olarak ayarladım.

Corp router'ına geri dönelim ve routing tablosunda bize ne göstereceğine bakalım:

```
Corp#sh ip eigrp neighbors
IP-EIGRP neighbors for process 10
H   Address     Interface  Hold Uptime   SRTT  RTO Q  Seq
                           (sec)         (ms)    Cnt Num
1   10.1.3.2    Se0/0/1    14 00:35:10    1   200  0  81
3   10.1.5.2    Se0/2/0    10 02:51:22    1   200  0  31
2   10.1.4.2    Se0/1/0    13 03:17:20    1   200  0  20
0   10.1.2.2    Se0/0/0    10 03:19:37    1   200  0  80
```

Bu çıktıdaki bilgiyi şöyle okuyoruz:

- **H** alanı, komşunun bulunma sırasını belirtir.
- Hold time ise, bu router'ın, belirli bir komşudan bir Hello paketi ulaşmasını bekleyeceği süredir.
- Uptime, komşuluğun, kurulu olduğu süreyi belirtir.
- SRTT(smooth round-trip timer) alanı, bu router'dan, komşusuna gidiş dönüş için geçen zamanı belirtir. Bu değer, bu komşusundan bir cevap multicast'inden sonra ne kadar bekleneceğini belirlemek için kullanılır. Şayet cevap zamanında alınmazsa router, haberleşmeyi tamamlamak için unicast'leri kullanmaya kalkışacaktır. Multicast girişimleri arasındaki zaman ile belirlenir.
- Retransmission Time Out (RTO) alanı, bir paketi, yeniden aktarım kuyruğundan bir komşuya tekrar aktarmadan önce, EIGRP'nin beklediği zaman miktarıdır.
- Q değeri, kuyrukta önemli mesajların olup olmadığını belirtir.(sürekli büyük değerler, bir problem olduğunu belirtir.)
- Seq alanı, komşudan alınan son güncellemenin sıra numarasını belirtir. Senkronizasyonun devamı ve mesajların yanlış sıralanmasının belirlenmesi prosesi veya tekrarlanmalarından kaçınmak için kullanılmaktadır.

NOT *show ip eigrp neighbors komutu, hem IP adreslerini kontrol etmenizi hem de adjacency kuran komşular için zaman aralığını ve kuyruk sayılarını tekrar aktarmanızı sağlar.*

`Show ip eigrp topology` komutunu kullanarak, Corp topoloji tablosunda ne göreceğimize bakalım:

```
Corp#sh ip eigrp topology
IP-EIGRP Topology Table for AS(10)/ID(10.1.5.1)
Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
       r - reply Status, s - sia Status
P 10.1.11.0/24, 1 successors, FD is 2172416
        via 10.1.5.2 (2172416/28160), Serial0/2/0
P 10.1.10.0/24, 1 successors, FD is 2172416
        via 10.1.5.2 (2195456/281600), Serial0/2/0
```

```
P 10.1.9.0/24, 1 successors, FD is 2195456
        via 10.1.4.2 (2195456/281600), Serial0/1/0
P 10.1.8.0/24, 1 successors, FD is 2195456
        via 10.1.4.2 (2195456/72960), Serial0/1/0
P 10.1.12.0/24, 1 successors, FD is 2172416
        via 10.1.5.2 (2172416/28160), Serial0/2/0
P 10.1.3.0/24, 1 successors, FD is 76839936
        via Connected, Serial0/0/1
        via 10.1.2.2 (9849856/7719936), Serial0/0/0
P 10.1.2.0/24, 1 successors, FD is 2169856
        via Connected, Serial0/0/0
        via 10.1.2.2 (2681856/551936), Serial0/0/0
P 10.1.1.0/24, 1 successors, FD is 28160
        via Connected, FastEthernet0/1
P 10.1.7.0/24, 1 successors, FD is 793600
        via 10.1.2.2 (2195456/281600), Serial0/0/0
        via 10.1.3.2 (77081600/281600), Serial0/0/1
P 10.1.6.0/24, 1 successors, FD is 793600
        via 10.1.2.2 (2195456/281600), Serial0/0/0
        via 10.1.3.2 (77081600/281600), Serial0/0/1
P 10.1.5.0/24, 1 successors, FD is 2169856
        via Connected, Serial0/2/0
P 10.1.4.0/24, 1 successors, FD is 2169856
        via Connected, Serial0/1/0
```

Tüm route'ların önünde P olduğuna dikkat edin. Yani route, passive state'tedir. Bu iyi bir şeydir, çünkü active state(A)'deki route'lar, router'ın bu network'e olan yolunu kaybettiğini ve yenisini bulmaya çalıştığını belirtir. Her kayıt ayrıca, tüm uzak network'lere feasible distance (FD), artı paketlerin, hedefleri boyunca gidecekleri next-hop komşularını belirtmektedir. Ayrıca, her kayıt, parantez içinde iki sayıya sahiptir. İlki feasible distance'ı, ikincisi uzak bir ağ için advertised distance'ı belirtir.

Şimdi burada, bazı şeyler ilginç olmaktadır. 10.1.7.0 ve 10.1.6.0 çıktılarında, her network'e iki link olduğunu ve feasible distance ile advertised distance'ın farklı olduğuna dikkat edin. Bunun anlamı, network'lere bir succesor ve yedek bir route olarak bir feasible successor olmasıdır. Böylesi çok güzel! 10.1.6.0 ve 10.1.7.0 network'üne iki route'da, topoloji tablosunda olduğu halde, sadece successor route (en düşük metriğe sahip olan), routing tablosuna kopyalanıp yerleştirilecektir.

*Route'un feasible successor olması için, advertised distance'ının, successor route'un feasible distance'ından daha küçük olması gerekmektedir.* **NOT**

Eşit variance'a (eşit cost değerine ) sahip olduklarında, EIGRP, iki linkte, otomatik olarak yük dengelemesi yapacaktır. EIGRP ayrıca, variance komutunu kullanırsak, eşit olmayan cost değerli linklerde de yük dengelemesi yapacaktır. Variance metriği, varsayılanda, 1'e ayarlıdır. Yani sadece eşit-cost değerli linklerde, yük dengelemesi olacaktır. Metriği, herhangi bir yerde, 128'e kadar değiştirebilirsiniz. Variance değerini değiştirmek, EIGRP'nin, lokal routing tablosunda, eşit olmayan cost değeriyle, çok sayıda, döngü oluşturmayan route'lar kurmasını sağlar.

Böylece variance, 1'e ayarlanırsa sadece aynı metriğe sahip route'lar successor olarak lokal routing tablosunda oluşturulacaklardır. Örneğin, variance 2'ye ayarlandıysa, successor metriğinin yarısından küçük metrikli EIGRP route, lokal routing tablosunda oluşturulacaktır (şayet, zaten bir feasible successor ise).

Şimdi debugging çıktılarını kontrol etmek için oldukça iyi bir zaman. İlk olarak, komşu router'lar arasında gönderilen Hello paketlerimizi gösterecek olan `debug eigrp packet` komutunu kullanalım:

```
Corp#debug eigrp packet
EIGRP Packets debugging is on
    (UPDATE, REQUEST, QUERY, REPLY, HELLO, IPXSAP, PROBE, ACK,
STUB,
    SIAQUERY, SIAREPLY)
Corp#
*Mar 21 23:17:35.050: EIGRP: Sending HELLO on FastEthernet0/1
*Mar 21 23:17:35.050:   AS 10, Flags 0x0, Seq 0/0 idbQ 0/0 iidbQ
un/rely 0/0
*Mar 21 23:17:35.270: EIGRP: Received HELLO on Serial0/1/0 nbr
10.1.4.2
*Mar 21 23:17:35.270:   AS 10, Flags 0x0, Seq 0/0 idbQ 0/0 iidbQ
   un/rely 0/0 peerQ un/rely 0/0
*Mar 21 23:17:35.294: EIGRP: Received HELLO on Serial0/0/0 nbr
10.1.2.2
*Mar 21 23:17:35.294:   AS 10, Flags 0x0, Seq 0/0 idbQ 0/0 iidbQ
   un/rely 0/0 peerQ un/rely 0/0
*Mar 21 23:17:38.014: EIGRP: Received HELLO on Serial0/2/0 nbr
10.1.5.2
*Mar 21 23:17:38.014:   AS 10, Flags 0x0, Seq 0/0 idbQ 0/0 iidbQ
   un/rely 0/0 peerQ un/rely 0/0
```

Corp router'ımın, üç EIGRP komşusuna bağlı olması ve 224.0.0.10 multicast'inin her 5 saniyede bir gönderilmesinden dolayı, bu güncellemeleri görmede herhangi bir problemim olmadı. AS'ninin, güncellemede yer aldığını fark ettiniz mi? Bundan dolayı bir komşu, aynı AS'e sahip değilse Hello güncellemesi atılacaktır.

Size önemli bir debugging komutu daha göstermek istiyorum: `debug ip eigrp notification` komutu (12.4 öncesi router'larda, `debug ip eigrp events` olarak belirtilirdi.) Bu komut çıktısının bir şey göstermemesi, sizin için sürpriz olmuştur. Haklısınız! Bu komuttan çıktı alacağınız tek zaman, ağınızda problem olduğu ya da ağ topluluğunuzda router'ınızdan bir network eklediğiniz ya da sildiğiniz zamandır. Çok iyi, problemsiz bir network'üm olduğunu bildiğimden bazı çıktıları görmek için Corp router'ımda bir interface'i kapatacağım:

```
Corp(config)#int f0/1
Corp(config-if)#shut
*Mar 21 23:25:43.506: IP-EIGRP(Default-IP-Routing-Table:10):
Callback:
   route_adjust FastEthernet0/1
*Mar 21 23:25:43.506: IP-EIGRP: Callback: ignored connected AS 0
10.1.1.0/24
*Mar 21 23:25:43.506:           into: eigrp AS 10
*Mar 21 23:25:43.506: IP-EIGRP(Default-IP-Routing-Table:10):
Callback:
   callbackup_routes 10.1.1.0/24
Corp(config-if)#n
```

```
*Mar 21 23:25:45.506: %LINK-5-CHANGED: Interface FastEthernet0/1,
   changed state to administratively down
*Mar 21 23:25:46.506: %LINEPROTO-5-UPDOWN: Line protocol on
Interface
   FastEthernet0/1, changed state to down
Corp(config-if)#no shut
Corp(config-if)#^Z
*Mar 21 23:25:49.570: %LINK-3-UPDOWN: Interface FastEthernet0/1,
   changed state to up
*Mar 21 23:25:49.570: IP-EIGRP(Default-IP-Routing-Table:10):
Callback:
   lostroute 10.1.1.0/24
*Mar 21 23:25:49.570: IP-EIGRP(Default-IP-Routing-Table:0):
Callback:
   redist connected (config change) FastEthernet0/1
*Mar 21 23:25:49.570: IP-EIGRP(Default-IP-Routing-Table:0):
Callback:
   redist connected (config change) Serial0/0/0
*Mar 21 23:25:49.570: IP-EIGRP(Default-IP-Routing-Table:0):
Callback:
   redist connected (config change) Serial0/0/1
*Mar 21 23:25:49.570: IP-EIGRP(Default-IP-Routing-Table:0):
Callback:
   redist connected (config change) Serial0/1/0
*Mar 21 23:25:49.570: IP-EIGRP(Default-IP-Routing-Table:0):
Callback:
   redist connected (config change) Serial0/2/0
*Mar 21 23:25:49.570: IP-EIGRP(Default-IP-Routing-Table:10):
Callback:
   route_adjust FastEthernet0/1
*Mar 21 23:25:50.570: %LINEPROTO-5-UPDOWN: Line protocol on
Interface
   FastEthernet0/1, changed state to up
```

Şimdiye kadar EIGRP hakkında çok şey öğrendiğinizi biliyorum, fakat bir yere ayrılmayın, çünkü bu bölüm henüz tamamlanmadı. Şimdi OSPF konusunu hafifletme zamanıdır.

*Önemli olduğundan bunu tekrarlayacağım; gerçek ağınızda bu komuttan bir çıktı gelmesini görmek istemezsiniz! Şayet görürseniz, bulup çözmeniz gereken en az bir probleminiz olma ihtimali vardır.*

NOT

# Open Shortest Path First (OSPF) Temelleri

*Open Shortest Path First (OSPF),* Cisco da dahil olmak üzere geniş bir yelpazedeki network üreticilerinin uyguladığı açık standart bir routing protokolüdür. Eğer birden fazla router'ınız varsa ve hepsi Cisco değilse, EIGRP'yi kullanamazsınız, değil mi? Böylece, kalan CCNA seçenekleriniz temel olarak RIP, RIPv2 ve OSPF'dir. Eğer büyük bir network ise, o zaman gerçekte tek seçeneğiniz OSPF ve route distribution'dır. Redistribution, bu bölümde daha önce bahsettiğimiz, routing protokolleri arasındaki bir çevrim servisidir.

OSPF, Dijkstra algoritması kullanarak çalışır. İlk olarak, bir shortest path tree (SPF) yapılandırılır ve sonuçta oluşan en iyi yollar routing tablosuna yerleştirilir. EIGRP kadar olmasa bile OSPF, çabuk converge olur ve aynı hedef içini çoklu, eşit cost değerli route'ları destekler. EIGRP gibi hem IP hem de Ipv6 routed protokollerini destekler.

OSPF şu özellikleri sağlar:

- Area'lar ve autonomous system'lardan oluşur.
- Routing güncelleme trafiğini minimize eder.
- Ölçeklenebilirlik sağlar.
- VLSM/CIDR'ı destekler.
- Sınırsız hop sayısına sahiptir.
- Birçok üretici konuşlandırmasına izin verir (open standart).

OSPF, çoğu insanın bildiği bir link-state routing protokolüdür. Bu nedenle, RIPv2 ve RIPv1 gibi geleneksel distance-vector routing protokolleri ile kıyaslanması faydalı olacaktır. Tablo 7.3, bu üç protokolün karşılaştırmasını vermektedir.

**Tablo 7.3:** OSPF ve RIP Karşılaştırması

| Karakteristik | OSPF | RIPv2 | RIPv1 |
|---|---|---|---|
| Protokol tipi | Link state | Distance vector | Distance vector |
| Classless desteği | Evet | Evet | Hayır |
| VLSM desteği | Evet | Evet | Hayır |
| Auto-summarization | Hayır | Evet | Evet |
| Manuel summarization | Evet | Hayır | Hayır |
| Discontiguous desteği | Evet | Evet | Hayır |
| Route yayma | Değişiklikte multicast | Periyodik multicast | Periyodik broadcast |
| Path metriği | Bant genişliği | Hoplar | Hoplar |
| Hop count limiti | Hiç bir şey | 15 | 15 |
| Convergence | Hızlı | Yavaş | Yavaş |
| Peer authentication | Evet | Evet | Hayır |
| Hiyerarşik network | Evet (area'lar kullanarak) | Hayır, (sadece flat) | Hayır, (sadece flat) |
| Güncellemeler | Değişiklikte olan | Route tablosu güncellemeleri | Route tablosu güncellemeleri |
| Route hesaplama | Dijkstra | Bellman-Ford | Bellman-Ford |

OSPF, Şekil 7.3'te listelediğim az sayıdakilerin dışında çok fazla özelliğe sahiptir ve tamamı binlerce network'e yayılabilen hızlı, ölçeklenebilir ve sağlam bir protokol olmasına katkıda bulunur.

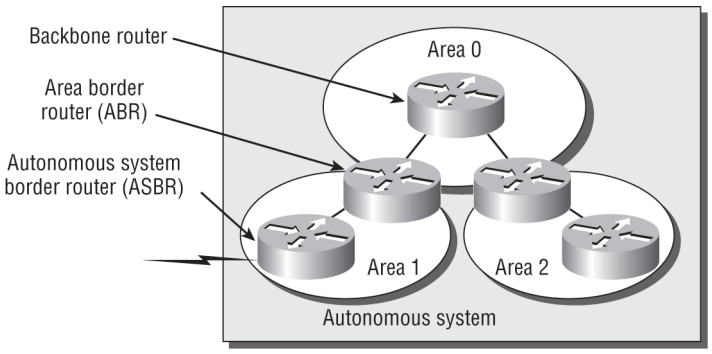
OSPF'in, hiyerarşik şekilde tasarlanması gerekir. Yani, büyük ağ topluluklarını, area denilen daha küçük ağ topluluklarına ayırabilirsiniz. Bu, OSPF için en iyi tasarımdır.

Aşağıdakiler, hiyerarşik tasarımda OSPF oluşturulması için sebeplerdir:

- Routing yükünü düşürmek.
- Convergence'i düşürmek.
- Network tutarsızlığını, tek area network'lerle sınırlı tutmak.

Bu, OSPF konfigürasyonunu basitleştirir, fakat daha çok dikkat ve ayrıntı gerektirir.

Şekil 7.4, tipik bir OSPF basit tasarımını gösterir. Her router'ın, area 0 ya da backbone area denilen omurgaya nasıl bağlandığına dikkat edin. OSPF, bir area 0'a sahip olmalıdır ve diğer area'lar bu area'ya bağlanmalıdır. (Area'ların, sanal linkler kullanarak bu area 0'a bağlanması bu kitabın kapsamında değildir.) Aynı AS'deki diğer area'ları, backbone area'ya bağlayan router'lar, Area Border Router (ABR) olarak adlandırılırlar. ABR'ın en az bir interface'inin area 0'da olması gerekmektedir.




***Şekil 7.4:*** *OSPF tasarım örneği.*

OSPF, içerde bir autonomous system çalıştırır, fakat ayrıca çok sayıda autonomous system'ları birbirine bağlar. Bu AS'leri bağlayan router, Autonomous System Boundary Router (ASBR) olarak tanımlanır.

İdeal olarak, route güncellemelerini minimumda tutmaya yardımcı olması ve problemlerin network'e yayılmasını engellemek için diğer router area'larını oluşturursunuz. Fakat bu, modülün kapsamı dışındadır. Onu sadece not alın.

EIGRP bölümünde olduğu gibi, ilk olarak, OSPF'i anlamanız için gerekli temel terminolojiyi anlatacağım.

## OSPF Terminolojisi

Bir harita ve pusula verildiğini, fakat doğu, batı, güney, kuzey nehir ve dağ, göl ya da çölle ilgili hiçbir bilginizin olmamasının ne kadar zor bir durum olduğunu hayal edebiliyor musunuz? Bu kavramlar hakkında bilgi sahibi olmadan yeni araçları iyi kullanamayacağınızı anlamak zor değildir. Bundan dolayı, OSPF'i keşfetmeye, sonraki bölümleri kaçırmanızı engelleyecek uzun bir terim listesiyle başlayacaksınız. Aşağıdakiler, daha önce kullandıklarınızla aşina gelecek OSPF terimleridir:

**Link:** Link, belirli bir network'e atanan network ya da router interface'idir. Bir interface, OSPF prosesine eklendiğinde, OSPF tarafında bir link olarak kabul edilir. Bu link ya da interface, hem up/down olması hem de bir ya da daha fazla IP adresi ile ilgili durum bilgisine sahip olacaktır.

**Router ID:** Router ID (RID)'si, router'ı belirlemek için kullanılan bir IP adresidir. Cisco, tüm yapılandırılmış loopback adreslerinin en yüksek IP adresli olanını kullanarak, Router ID'yi seçer. Şayet, yapılandırılmış bir loopback adresi yoksa OSPF, tüm aktif fiziksel interface'lerden en yüksek IP adresli olanını seçecektir.

**Neighbor:** Neighbor'lar, point-to-point seri linkle bağlı iki router gibi, genel ağda bir interface'e sahip iki ya da daha fazla router'dır.

**Adjacency:** Bir adjacency, route güncellemelerini direkt olarak değiş tokuş etmeye izin veren iki OSPF router arasındaki ilişkidir. OSPF, tüm komşularıyla route'ları direk olarak paylaşan EIGRP'nin tersine, routing bilgilerinin paylaşılmasında oldukça seçicidir. OSPF, sadece adjacen-

cy kurduğu komşularıyla route'ları paylaşırlar. Tüm komşularda, adjacent olmayacaktır, bu hem network tipine hem de router'ların konfigürasyonuna bağlıdır.

**Hello protokolü:** OSPF Hello protokolü, dinamik komşu tespiti sağlar ve komşu ilişkilerini devam ettirir. Hello paketleri ve Link State Advertisement'ları (LSA), topolojik veritabanı oluşturur ve devamını sağlar. Hello paketleri, 224.0.0.5'e gönderilir.

**Neighborship database:** Neighborship database, Hello paketlerinin görüldüğü, tüm OSPF router'larının bir listesidir. Router ID'sini ve durumunu içeren detaylar, neighborship database'indeki router'larda korunmaktadır.

NOT *LSA paketleri, topoloji veritabanının güncellenmesi ve devamlılığının sağlanması için kullanılmaktadır.*

**Topological database:** Topological database, bir area'dan alınan Link State Advertisement paketlerinin hepsiyle ilgili bilgileri içerir. Router'lar, topoloji database'indeki bilgileri, her network'e en kısa yolu hesaplayan Dijkstra algoritmasına girdi olarak hesaplarlar.

**Link State Advertisement:** Bir Link State Advertisement (LSA), OSPF router'lar arasında paylaşılan link-state ve routing bilgisini içeren bir OSPF veri paketidir. Farklı LSA paket türleri vardır ve bunlara kısaca gireceğim. Bir OSPF router, sadece adjacency kurduğu router'larla LSA paketlerini değiş tokuş edecektir.

**Designated router:** Bir desiganted router (DR), OSPF router'ların, aynı multi-access network'lere bağlandığı zaman seçilirler. Cisco, bu broadcast network'lerden bahsetmeyi sever. Fakat aslında onlar, çok sayıda alıcıya sahip network'lerdir. Multi-access'i, multipoint ile karıştırmamaya çalışın. Bazen kolayca karıştırılabilir.

Başlıca örnek, bir Ethernet LAN'ıdır. Kurulu komşulukların sayısını minimuma indirmek için, bir DR, broadcast network ya da linkindeki diğer router'lara yada router'lardan routing bilgilerini yaymak/almak için seçilmektedir(elenmektedir). Bu, topoloji tablolarının senkronize olmasını garantiler. Paylaşılan network'lerdeki tüm router'lar, DR ve backup designated router (BDR) ile komşuluk kuracaklardır. Seçim, en yüksek priority'e sahip router tarafından kazanılacaktır ve priority'nin birden fazla router'da aynı olması durumunda, DR seçimi için Router ID kullanılmaktadır.

**Backup designated router:** Bir backup designated router (BDR), multi-access linklerinde (Cisco'nun bunu bazen, broadcast network'ü olarak belirttiğini hatırlayın) DR için öncelikli bir yedektir. BDR, OSPF komşu router'larından tüm routing güncellemelerini alır, fakat LSA güncellemelerini göndermez.

**OSPF area'ları:** Bir OSPF area, bitişik network ve router'larının bir grubudur. Aynı area'daki tüm router'lar genel bir Area ID'sini kullanır. Bir router'ın, aynı anda birden fazla area'ya üye olabilmesinden dolayı AreaID, router'daki belirli bir interface ile ilgilidir. Bu, bazı interface'ler area 0'a ait olurken, bazı interface'lerin area 1'e ait olmalarını sağlar. Aynı area'daki tüm router'lar, aynı topoloji tablosuna sahiptirler. OSPF yapılandırdığınızda, bir area 0'ın olmasını ve bunun tipik olarak, network'ün omurgasına bağlı router'larda yapılandırıldığını hatırlamalısınız. Area'lar ayrıca, hiyerarşik bir network organizasyonu kurulmasında da rol oynarlar. (OSPF'in ölçeklenirliğini artırır.)

**Broadcast (multi-access):** Broadcast (multi-access) network'leri, Ethernet gibi hem aynı network'lere bağlanması (ya da erişmesi) için birçok cihaza izin veren hem de bir paketin, ağdaki tüm network'lere taşındığı bir broadcast kabiliyeti sağlayan network'lerdir. OSPF'te, her broadcast multi-access network için bir DR ve bir BDR seçilmelidir.

**Non-broadcast multi-access:** Non-broadcast multi-access (NBMA) network'leri, Frame-Relay, X.25 ve Asynchronous Transfer Mode (ATM) türü network'lerdir. Bu network'ler, multi-access sağlar, fakat Ethernet gibi broadcast yeteneği yoktur. Bu nedenle, NBMA network'leri, düzgün çalışması için, belirli OSPF konfigürasyonu gerektirir ve komşu ilişkisi tanımlanmalıdır.

NOT *DR ve BDR, broadcast ve non-broadcast multi-access network'lerde seçilirler. Seçimler, bu bölümde detaylı bir şekilde işlenecektir.*

**Point-to-point:** Point-to-point, tek bir iletişim yolu sağlayan iki router arasında direkt bir bağlantı içeren network topoloji tipini belirtir. Point-to-point bağlantı, iki router'ı direkt bağlayan seri kablo gibi fiziksel ya da Frame Relay network'ündeki bir devre ile bağlı, binlerce kilometre uzaklıktaki iki router gibi mantıksal olabilir. Her iki durumda da, bu konfigürasyon tipi, DR ve BDR ihtiyacını ortadan kaldırır. Komşular, otomatik olarak belirlenir.

**Point-to-multipoint:** Point-to-multipoint, bir router'daki tek bir interface ile birçok hedef router arasındaki bağlantıların serisinden oluşan network topoloji tipine işaret eder. Point-to-multipoint bağlantıyı paylaşan tüm router'ların interface'lerinin hepsi, aynı network'e dahildir. Point-to-point'teki gibi, DR ve BDR'a gerek yoktur.

Bu terimlerin tamamı, OSPF operasyonunu anlamada önemli bir rol oynarlar. Bu nedenle, bunlarla aşina olduğunuza emin olun. Bu bölümün kalanını okumak, terimlerin anlamlarının oturmasında size yardımcı olacaktır.

## SPF Tree Hesaplaması

Bir area'daki her router, bu area'daki her network için en iyi/kısa yolu hesaplar. Bu hesaplama, topoloji veritabanında toplanan bilgiye dayanır ve shortest path first (SPF) olarak bilinen bir algoritmadır. Router'ın kök olduğu ve diğer network'lerin, dallar ve yapraklarla düzenlendiği bir ağacı (daha çok aile soyağacı gibi) oluşturan, area'daki her router'ı resmeder. Bu, route'ları routing tablosuna koymak için route'lar tarafından kullanılan shortest path ağacıdır.

Şunu anlamak önemlidir ki, bu ağaç, router'ın kendisi gibi, sadece, aynı area'da olan network'leri içerir. Şayet bir router, birçok area'ya bağlı interface'e sahipse, her area için ayrı ağaç oluşturulacaktır. SPF algoritmasının route seçme prosesi sırasında dikkat edilen önemli kriterlerden birisi, bir network için olası her yolun metriği ya da cost değeridir. Fakat bu SPF hesaplaması, diğer area'lardan route'lara uygulanmaz.

OSPF, cost olarak belirtilen bir metrik kullanır. Bir cost, bir SPF ağacında bulunan her dış interface'le ilgilidir. Cost'un, RFC 2338'de tanımlanan gelişigüzel bir değer olmasından dolayı, Cisco, her OSPF çalışan interface'i için kendi cost hesaplama yöntemini geliştirmek zorunda kaldı. Cisco, $10^8$/bandwidth formülünü kullanır. Bandwidth, interface'in yapılandırılmış bant genişliğidir. Bu formülü kullanırsak, 100Mbps Ethernet interface'in, 1 cost değerine ve 10Mbps Ethernet interface'in de 10 cost değerine sahip olduğunu buluruz.

*64,000 bandwidht ayarlanmış bir interface, 1,563 cost değerine sahip olacaktır.*

NOT

*Cisco, link cost'larını bant genişliğine dayandırır. Diğer üreticiler, verilen linkin cost'unu hesaplamak için diğer metrikleri kullanabilir. Farklı üreticilerin router'ları arasındaki linkler bağlandığında, diğer üreticinin router'ı ile eşleşmesi için cost'u ayarlamak zorunda kalabilirsiniz. OSPF'in çalışması için iki router'a da aynı cost değeri atanması gerekir.*

NOT

Bu değer, `ip ospf cost` komutu kullanılarak, geçersiz kılınabilir. Cost, 1'le 65,535 arasında bir sayı ile değiştirilerek manipüle edilebilir. Cost, her interface'e atandığından değer, cost'unu değiştirmek istediğiniz interface'de değiştirilmelidir.

# OSPF Konfigürasyonu

Basit OSPF konfigürasyonu, RIP, IGRP ve EIGRP gibi basit değildir ve OSPF'in sağladığı birçok seçenek kullanılınca, gerçekten karmaşık olabilmektedir. Çalışmalarınızda, basit single-area OSPF konfigürasyonu ile ilgilenmelisiniz. Aşağıdaki bölüm, tek area'lı bir OSPF'in nasıl yapılandırılacağını açıklamaktadır.

Aşağıdakiler, OSPF konfigürasyonunun temel unsurlarıdır:

- OSPF'i etkinleştirmek.
- OSPF area'ları yapılandırmak.

## OSPF'i Etkinleştirmek

OSPF'i yapılandırmanın en kolay ve ayrıca en az ölçeklenebilir yolu, tek bir area kullanmaktır. Bunu yapmak, minimum iki komut kullanmayı gerektirir.

OSPF routing prosesini aktif hale getirmek için kullandığınız komut şöyledir:

```
Lab_A(config)#router ospf ?
<1-65535>
```

1-65,535 aralığındaki bir değer, OSPF Process ID'sini tanımlar. Bu router'daki benzersiz bir numaradır ve belirli bir proses altındaki OSPF konfigürasyon komutlar serisini gruplar. Farklı OSPF router'lar, haberleşmek için aynı Process IP'sini kullanmak zorunda değildir. Sadece lokal bir değerdir, 0'la başlayamaz, en az 1 ile başlamak zorundadır.

Şayet isterseniz, aynı router'da, birden fazla OSPF prosesi eşzamanlı olarak çalışabilir. İkinci proses, tamamen farklı bir topoloji tablosu tutacak ve bağlantılarını, ilk prosesten bağımsız olarak yönetecektir. CCNA konularının, tek OSPF prosesi ile çalışan single-area OSPF'i kapsamasından dolayı, bu kitapta ona odaklanacağım.

NOT *OSPF Process ID'sinin, OSPF veritabanının eşsiz bir örneğini teşhis etmesi gerekir ve lokal olarak önemlidir.*

## OSPF Area'ların Konfigürasyonu

OSPF prosesini tanımladıktan sonra hem OSPF haberleşmelerini aktifleştirmek istediğiniz interface'leri hem de her oturumdaki area'ları tanımlamanız gerekir. Bu ayrıca, diğerlerine yayınlayacağınız network'leri yapılandıracaktır. OSPF, konfigürasyonda wildcard'lar kullanır. (Ayrıca, access-list'lerde de kullanılmaktadır, bölüm 10'da işlenecektir.)

Basit OSPF konfigürasyon örneği aşağıdaki gibidir:

```
Lab_A#config t
Lab_A(config)#router ospf 1
Lab_A(config-router)#network 10.0.0.0 0.255.255.255
 area ?
  <0-4294967295>  OSPF area ID as a decimal value
  A.B.C.D         OSPF area ID in IP address format
Lab_A(config-router)#network 10.0.0.0 0.255.255.255
 area 0
```

OSPF Process ID numarasının ne olduğunun önemli olmadığını hatırlayın. Network'teki her router'da aynı ya da farklı olabilir, bunun önemi yoktur. O, lokal olarak önemlidir ve sadece router'daki OSPF routing'ini etkinleştirir.

Network komutunun bağımsız değişkenleri, network adresi (10.0.0.0) ve wildcard maskı (0.255.255.255)'dir. Bu iki numaranın kombinasyonu, OSPF'in çalışacağı interface'leri tanımlar ve OSPF LSA yayınlarında yer alır. OSPF bu komutu, 10.0.0.0 network'ünde yapılandırılan router'daki bir interface'i bulmak için kullanacaktır ve bulduğu herhangi bir interface'i, area 0'ya yerleştirecektir. 4.2 milyar area oluşturabileceğinize dikkat edin (bir router'ın, aslında bu kadar çok area'yı oluşturacağından kuşkuluyum, fakat onlara 4.2 milyara kadar numara verebilirsiniz.). Ayrıca bir alanı IP adresi formatı kullanarak ta etiketleyebilirsiniz.

NOT *Area'lar, 0'dan 4.2 milyara kadar herhangi bir sayı olabilir. Bu sayıları, 1'den 65,535'e kadar olan Process ID'si ile karıştırmayın.*

Wildcard'lara hızlı bir göz atalım: wildcard mask'taki 0 okteti, network'te, tamamıyla eşleşmesi gereken ilgili okteti işaret eder. Diğer taraftan, 255, network adresindeki ilgili oktetin ne olduğuyla ilgilenmediğinizi belirtir. 1.1.1.1 0.0.0.0 network ve wildcard kombinasyonu, sadece 1.1.1.1'i eşleştirecektir. Bu, belirli bir interface'te OSPF'i, basit yoldan aktifleştirmek isterseniz, çok kulla-

nışlıdır. Şayet bir network aralığının eşleşmesinde ısrar ediyorsanız, 1.1.0.0 0.0.255.255 network ve mask kombinasyonu, 1.1.0.0-1.1.255.255 aralığındaki her şeyi eşleştirecektir. Bundan dolayı, 0.0.0.0 wildcard masklarını kullanmaya devam etmek ve OSPF interface'lerini tek tek tanımlamak, daha basit ve güvenli olacaktır.

Son bağımsız değişken, area numarasıdır. Bu, network'te tanımlanan interface'lerin ve wildcard mask bölümünün ait olduğu area'yı belirtir. OSPF router'larının sadece, interface'leri aynı area numarasına ait yapılandırılan bir network'ü paylaşırlarsa, komşu olacaklarını hatırlayın. Area numarası formatı, 1–4,294,967,295 aralığında bir ondalık sayı yada standart noktalı-ondalık sayı gösteriminde olabilir. Örnek olarak, area 0.0.0.0, olası bir area'dır ve area 0 ile özdeştir.

## Wildcard Örneği

Network'ümüzü yapılandırmaya geçmeden önce subnetleri ve wildcard'ları kullanıyorsak, OSPF network tanımlarımızın ne olacağını anlamak için daha zor OSPF network konfigürasyonlarına bir göz atalım.

Dört farklı interface'ine bağlı, şu dört subnete sahip bir router'ınız vardır:

- 192.168.10.64/28
- 192.168.10.80/28
- 192.168.10.96/28
- 192.168.10.8 /30

Tüm interface'lerin area 0'da olması gerekir. Bana, en kolay konfigürasyon bu geliyor:

```
Test#config t
Test(config)#router ospf 1
Test(config-router)#network 192.168.10.0 0.0.0.255 area 0
```

Fakat kolay her zaman en iyisi demek değildir, bu nedenle kolay olmasına rağmen, eğlence bunun neresindedir? Ve daha kötüsü, muhtemelen CCNA konularını kapsamamaktadır. Gelin, subnet adreslerini ve wildcard'ları kullanarak her interface için ayrı bir network ifadesi oluşturalım. Bu şöyle görünecektir:

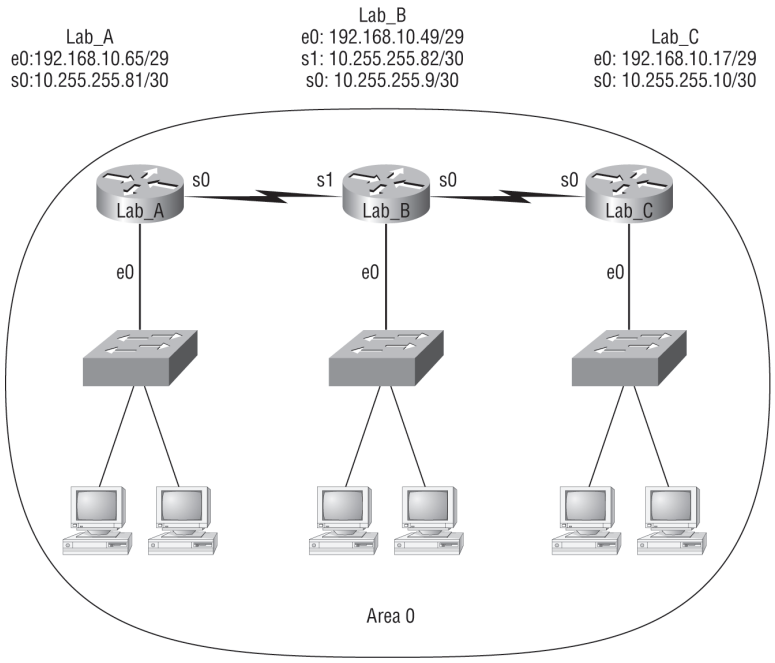
```
Test#config t
Test(config)#router ospf 1
Test(config-router)#network 192.168.10.64 0.0.0.15 area 0
Test(config-router)#network 192.168.10.80 0.0.0.15 area 0
Test(config-router)#network 192.168.10.96 0.0.0.15 area 0
Test(config-router)#network 192.168.10.8 0.0.0.3 area 0
```

Gerçekten OSPF, ilk olarak gösterdiğim kolay konfigürasyonla tamamen aynı şekilde çalışır. Kolay konfigürasyonun tersine, bu CCNA konularını içermektedir!

Wildcard'ları yapılandırdığımızda, onların daima blok boyutundan bir küçük olduğunu hatırlayın. /28, 16 blok boyutudur, bu nedenle, subnet adresini kullanarak network komutunu yazdık ve sonra ilgili oktette 15 wildcard'ı ekledik. /30 için blok boyutu 4'tür, 3 wildcard'ını kullanırız.

Örnek olarak Şekil 7.5'i kullanalım ve bu konuyu iyi kavradığınızdan emin olmak için wildcard'ları kullanarak bu network'ü OSPF ile yapılandıralım. Şekil 7.5, her interface'in IP adresi ile birlikte üç-router'lı bir network'ü göstermektedir.

Bunu başarabilmeniz için yapmanız gereken, her interface'e bakmak ve adreslerin olduğu subnetleri belirlemektir. "Neden interface'lerin IP adreslerini, 0.0.0.0 wildcard ile kullanmıyoruz? diye düşündüğünüzü biliyorum. Evet, yapabilirsiniz, burada CCNA konularına dikkat ediyoruz, en kolayını yapmaya değil.




***Şekil 7.5:*** *Örnek OSPF wildcard konfigürasyonu.*

Her interface için IP adresi, şekilde görülmektedir. Lab_A router'ı, iki doğrudan bağlı subnete sahiptir: 192.168.10.64/29 ve 10.255.255.80/30. Wildcard'ları kullanarak OSPF konfigürasyonu şöyledir:

```
Lab_A#config t
Lab_A(config)#router ospf 1
Lab_A(config-router)#network 192.168.10.64 0.0.0.7 area 0
Lab_A(config-router)#network 10.255.255.80 0.0.0.3 area 0
```

Lab_A router, ethernet 0 interface'inde, /29 yada 255.255.255.248 maskı kullanmaktadır. Bu 8 blok boyutudur ve wildcard, 7'dir. S0 interface'i, 255.255.255.252 maskına sahiptir, blok boyutu 4'tür ve wildcard, 3'tür. IP adres ve / gösterimine bakmaz ve sonra subnet maskı ile wildcard'ını anlamazsanız, OSPF'i bu yolla yapılandıramazsınız, değil mi?

Diğer iki konfigürasyonumuz şöyledir:

```
Lab_B#config t
Lab_B(config)#router ospf 1
Lab_B(config-router)#network 192.168.10.48 0.0.0.7 area 0
Lab_B(config-router)#network 10.255.255.80 0.0.0.3 area 0
Lab_B(config-router)#network 10.255.255.8 0.0.0.3 area 0

Lab_C#config t
Lab_C(config)#router ospf 1
Lab_C(config-router)#network 192.168.10.16 0.0.0.7 area 0
Lab_C(config-router)#network 10.255.255.8 0.0.0.3 area 0
```

Lab_A konfigürasyonunda belirttiğim gibi, bir interface'in IP adresine bakarak subnet maskını ve wildcard'ını belirleyebilmelisiniz. Şayet bunu yapamazsanız, gösterdiğim şekilde wildcard'ları kullanarak OSPF'i yapılandıramazsınız. Bu nedenle, kendinizi rahat hissedene kadar bunu çalışalım.

## Network'ümüzü OSPF ile Yapılandırmak

Şimdi biraz eğlenelim. Ağ topluluğumuzu, sadece area 0 kullanarak, OSPF ile yapılandıralım. Bunu yapmadan önce, OSPF'in 110 administrator distance'a sahip olmasından dolayı (EIGRP'nin AD'si 90'dır), EIGRP'yi silmek zorundayız. Fakat şimdi 871W, OSPF routing protokolünü desteklediğinden, R3 ve 871W router'larından RIP'i de sileceğim.

OSPF konfigürasyonun birçok yolu vardır ve dediğim gibi, kullanmak için en kolayı ve basiti, 0.0.0.0 wildcard maskıdır. Fakat her router'ı OSPF'le, farklı şekilde yapılandırmak ve tamamıyla aynı sonuca ulaşacağımızı göstermek istiyorum. (Bu bize, bir şeyleri berbat etmenin birçok yöntemi olduğunu gösterir!)

### Corp

Corp router'ın konfigürasyonu şöyledir:

```
Corp#config t
Corp(config)#no router eigrp 10
Corp(config)#router ospf 132
Corp(config-router)#network 10.1.1.1 0.0.0.0 area 0
Corp(config-router)#network 10.1.2.1 0.0.0.0 area 0
Corp(config-router)#network 10.1.3.1 0.0.0.0 area 0
Corp(config-router)#network 10.1.4.1 0.0.0.0 area 0
Corp(config-router)#network 10.1.5.1 0.0.0.0 area 0
```

Burada, üzerinde konuşmamız gereken bazı hususlar görülmektedir. İlk olarak, EIGRP'yi sildim, sonra OSPF'i ekledim. Peki, neden OSPF 132'yi kullandım? Onun gerçekten önemi yoktur. Sayı önemsizdir.

Network komutları oldukça basittir. Her interface'in IP adresini yazdım ve IP adresinin, her okteti tamamen eşleştireceği, 0.0.0.0 wildcard maskını kullandım. Fakat kolay olan daima iyidir diyorsanız, şunu yapın:

```
Corp(config)#router ospf 132
Corp(config-router)#network 10.1.0.0 0.0.255.255 area 0
```

Beş yerine tek satır! Anlamanızı istediğim, network komutunu ne şekilde yapılandıracağınızın gerçekten önemli olmamasıdır. OSPF burada aynı şekilde çalışacaktır. Şimdi, R1'e geçelim. Bazı şeyleri kolaylaştırmak için basit konfigürasyonumuzu kullanacağım

### R1

R1 router'ı, dört doğrudan bağlı network'e sahiptir. Her interface'de yazmak yerine, tek bir network komut örneği kullanabilir ve tamamıyla aynı çalışmasını sağlayabiliriz:

```
R1#config t
R1(config)#no router eigrp 10
R1(config)#router ospf 1
R1(config-router)#network 10.1.0.0 0.0.255.255 area0
                                                   ^
% Invalid input detected at '^' marker.

R1(config-router)#network 10.1.0.0 0.0.255.255 area 0
```

Ufak yazı hatalarından biri daha, area komutu ile area numarası arasına boşluk bırakmayı unuttum. Bu gerçekten hızlı ve etkili bir konfigürasyondur.

Tüm yaptığım, ilk olarak EIGRP'yi pasif yapmam ve sonra, OSPF routing process 1'i etkinleştirmek ve 0.0.255.255 wildcard mask ile network 10.1.0.0 komutunu eklemek oldu. Bu yapılan basitçe "10.1 ile başlayan herhangi bir interface'i bul ve bu interface'leri area 0'a yerleştir." demektir. Çabuk, kolay ve düz!

## R2

R2 router'ını, üç network'e direkt bağlı olduğu konusunda uyaralım:

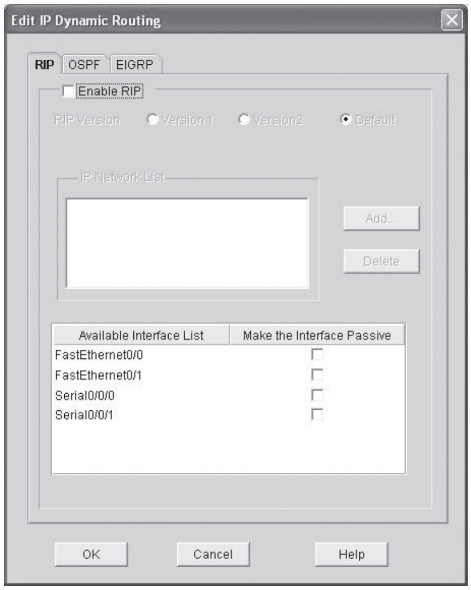
```
R2#config t
R2(config)#no router eigrp 10
R2(config)#router ospf 45678
R2(config-router)#network 10.0.0.0 0.255.255.255 area 0
```

1'le 65,535 arasında istediğim herhangi bir sayıyı Process ID olarak kullanabilirim. 10.0.0.0'ı, 0.255.255.255 wildcard ile kullandığıma dikkat edin. Bu da sorunsuz çalışır.

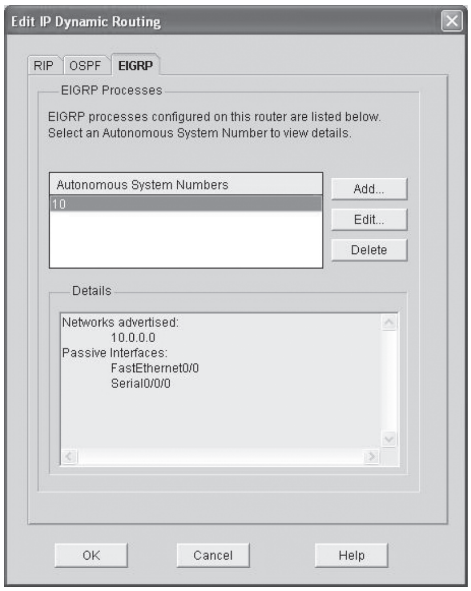
## R3

R3 router'ı için, OSPF daha düşük AD'a sahip olduğundan, RIP'i rahatsız etmese de, RIP ve EIGRP'yi devreden çıkarmamız gerekir. Genelde yaptığımız gibi SDM'i kullanacağız.

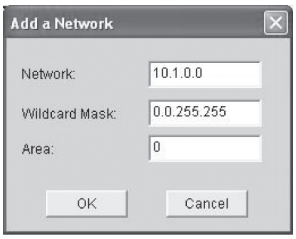
İlk ekran çıktımız, RIP'in devre dışı olduğunu gösterir.



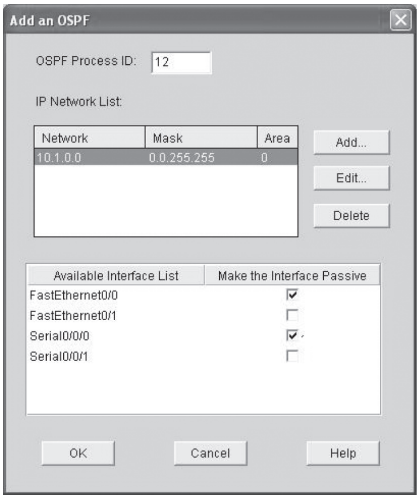

Sonraki grafiğimiz, EIGRP'nin devre dışı olduğunu göstermektedir. (sadece Delete butonuna tıkladım.)



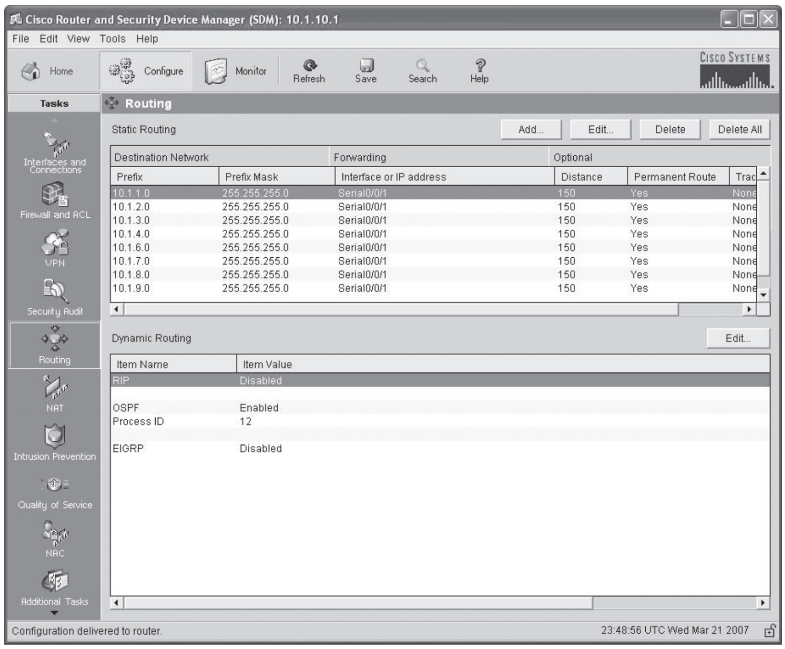

OSPF sekmesine tıkladıktan sonra, Add Network'e tıkladım ve OSPF bilgimi ekledim.




Sonra, OK butonuna tıkladım, pasif interface'lerimi seçtim ve tekrar OK butonuna tıkladım.




İyi gidiyoruz. R3 router'ında sadece OSPF routing çalıştığını görebilirsiniz.




### 871W

Sonunda, son router'ımız! RIP'i devreden çıkaralım ve OSPF'i ekleyelim:

```
871W#config t
871W(config)#no router rip
871W(config)#router ospf 1
871W(config-router)#network 10.1.11.0 0.0.0.255 area 0
871W(config-router)#network 10.1.12.0 0.0.0.255 area 0
```

Router'larımızın hepsini OSPF ile yapılandırdık, sırada ne var? Sırada, doğruluklarını kontrol etmek var. Hala, OSPF'in gerçekten çalıştığından emin olmamız gerekmektedir. Şimdi bununla ilgileneceğiz.

## OSPF Konfigürasyonunun Doğruluğunu Kontrol Etmek

Uygun OSPF konfigürasyonunu ve çalışmasını doğrulamak için bazı yollar vardır ve aşağıdaki bölümlerde, bunu yapmanız için bilmeniz gereken OSPF `show` komutlarını göstereceğim. C o r p router'ının routing tablosuna hızlıca göz atarak başlayacağız:

Corp router'ında `show ip route` komutunu çalıştıralım:

```
     10.0.0.0/24 is subnetted, 12 subnets
O       10.1.11.0 [110/65] via 10.1.5.2, 00:01:31, Serial0/2/0
O       10.1.10.0 [110/65] via 10.1.5.2, 00:01:31, Serial0/2/0
O       10.1.9.0 [110/74] via 10.1.4.2, 00:01:31, Serial0/1/0
O       10.1.8.0 [110/65] via 10.1.4.2, 00:01:31, Serial0/1/0
O       10.1.12.0 [110/66] via 10.1.5.2, 00:01:31, Serial0/2/0
C       10.1.3.0 is directly connected, Serial0/0/1
C       10.1.2.0 is directly connected, Serial0/0/0
C       10.1.1.0 is directly connected, FastEthernet0/1
O       10.1.7.0 [110/74] via 10.1.3.2, 00:01:32, Serial0/0/1
                 [110/74] via 10.1.2.2, 00:01:32, Serial0/0/0
O       10.1.6.0 [110/74] via 10.1.3.2, 00:01:32, Serial0/0/1
```

```
                [110/74] via 10.1.2.2, 00:01:32, Serial0/0/0
C       10.1.5.0 is directly connected, Serial0/2/0
C       10.1.4.0 is directly connected, Serial0/1/0
```

Corp router'ı, tüm 12 network'ümüz için "O" ile başlayan OSPF internal route'larının bulunduğunu göstermektedir. (C'ler bizim direk bağlı network'lerimizdir.) Router, ayrıca 10.1.6.0 ve 10.1.7.0 network'lerine çift route olduğunu da gösterir. İnterface'lerden bandwidth ve delay komutlarını sildim, böylece metriği belirlemek için varsayılanlar kullanılmaktadır. Fakat OSPF'in, bir network'e ulaşmak için en iyiyi bulmada, sadece bant genişliğini kullandığını hatırlayın.

*OSPF, sadece eşit cost değerli linklerde yük dengelemesi yapabilmektedir. EIGRP' gibi eşit olmayan cost değerli linklerde yük dengelemesi yapamaz.*

NOT

Bilmeniz gereken OSPF doğrulama komutlarını gösterme zamanı geldi.

## Show ip ospf Komutu

`Show ip ospf komutu`, router'da çalışan bir ya da tüm OSPF proseslerini görüntülemek için kullanılır. Bilgi, RouterID'si, area bilgisi, SPF istatistikleri ve LSA timer bilgisini de içerir. Corp router çıktısını kontrol edelim:

```
Corp#sh ip ospf
 Routing Process "ospf 132" with ID 10.1.5.1
 Start time: 04:32:04.116, Time elapsed: 01:27:10.156
 Supports only single TOS(TOS0) routes
 Supports opaque LSA
 Supports Link-local Signaling (LLS)
 Supports area transit capability
 Router is not originating router-LSAs with maximum metric
 Initial SPF schedule delay 5000 msecs
 Minimum hold time between two consecutive SPFs 10000 msecs
 Maximum wait time between two consecutive SPFs 10000 msecs
 Incremental-SPF disabled
 Minimum LSA interval 5 secs
 Minimum LSA arrival 1000 msecs
 LSA group pacing timer 240 secs
 Interface flood pacing timer 33 msecs
 Retransmission pacing timer 66 msecs
 Number of external LSA 0. Checksum Sum 0x000000
 Number of opaque AS LSA 0. Checksum Sum 0x000000
 Number of DCbitless external and opaque AS LSA 0
 Number of DoNotAge external and opaque AS LSA 0
 Number of areas in this router is 1. 1 normal 0 stub 0 nssa
 Number of areas transit capable is 0
 External flood list length 0
    Area BACKBONE(0)
        Number of interfaces in this area is 5
        Area has no authentication
        SPF algorithm last executed 00:14:52.220 ago
        SPF algorithm executed 14 times
```

```
        Area ranges are
        Number of LSA 6. Checksum Sum 0x03C06F
        Number of opaque link LSA 0. Checksum Sum 0x000000
        Number of DCbitless LSA 0
        Number of indication LSA 0
        Number of DoNotAge LSA 0
        Flood list length 0
```

10.1.5.1 Router ID (RID)'in, router'da yapılandırılan en yüksek IP adresi olduğuna dikkat edin.

## Show ip ospf database Komutu

`Show ip ospf database` komutunu kullanmak bize, ağ topluluğundaki (AD) router'ların sayıları ve komşu router'ın ID'si (bu, daha önce açıkladığım topoloji database'idir) hakkındaki bilgileri verir. Show ip eigrp topology komutunun aksine, bu komut, "OSPF router'larını" gösterir. EIGRP'ni yaptığı gibi, AS'deki tüm linkleri göstermez.

Çıktı, area 0'ı incelemektedir. Yine Corp router'dan örnek bir çıktı bulabilirsiniz:

```
Corp#sh ip ospf database

       OSPF Router with ID (10.1.5.1) (Process ID 132)

                Router Link States (Area 0)

Link ID         ADV Router      Age         Seq#       Checksum
Link count
10.1.5.1        10.1.5.1        72          0x80000002 0x00F2CA 9
10.1.7.1        10.1.7.1        83          0x80000004 0x009197 6
10.1.9.1        10.1.9.1        73          0x80000001 0x00DA1C 4
10.1.11.1       10.1.11.1       67          0x80000005 0x00666A 4
10.1.12.1       10.1.12.1       67          0x80000004 0x007631 2

                Net Link States (Area 0)

Link ID         ADV Router      Age         Seq#       Checksum
10.1.11.2       10.1.12.1       68          0x80000001 0x00A337
```

Beş router'ın hepsini ve her birinin RID'lerini (her router'daki en yüksek IP adresini) görebilirsiniz. Router çıktısı, link ID'sini (bir interface'in aynı zamanda bir link olduğunu hatırlayın) ve ADV router yada yayınlayan router'da, bu linkteki router'un RID'ini gösterir.

## Show ip ospf interface Komutu

`Show ip ospf interface` komutu, tüm interface bağlantılı OSPF bilgisini gösterir. Veri, tüm interface'ler ya da belirli interface'ler için OSPF bilgisini görüntülemektedir. (bazı önemli yerleri koyu renkli göstereceğim)

```
Corp#sh ip ospf interface f0/1
FastEthernet0/1 is up, line protocol is up
  Internet Address 10.1.1.1/24, Area 0
```

```
  Process ID 132, Router ID 10.1.5.1, Network Type BROADCAST,
Cost: 1
  Transmit Delay is 1 sec, State DR, Priority 1
  Designated Router (ID) 10.1.5.1, Interface address 10.1.1.1
  No backup designated router on this network
  Timer intervals configured, Hello 10, Dead 40, Wait 40,
Retransmit 5
    oob-resync timeout 40
    Hello due in 00:00:01
  Supports Link-local Signaling (LLS)
  Index 1/1, flood queue length 0
  Next 0x0(0)/0x0(0)
  Last flood scan length is 0, maximum is 0
  Last flood scan time is 0 msec, maximum is 0 msec
  Neighbor Count is 0, Adjacent neighbor count is 0
  Suppress hello for 0 neighbor(s)
```

Bu komutla şu bilgiler görüntülenmiştir:

- Interface IP adresi
- Area ataması
- Process ID
- Router ID
- Network tipi
- Cost
- Priority
- DR/BDR seçim bilgisi (şayet uygulanabilirse)
- Hello ve Dead timer süreleri
- Bitişik komşu bilgisi

Show `ip ospf interface f0/1` komutunu kullanmamın sebebi, FastEthernet broadcast multi-access network'te bir designated router seçiminin olacağını bilmemdir. DR ve BDR seçiminin detayına birazdan geçeceğiz.

## Show ip ospf neighbor Komutu

Show `ip ospf neighbor komutu`, çok kullanışlıdır. Çünkü o, komşuları ve adjacency durumu hakkında uygun OSPF bilgilerini özetler. Şayet bir DR ve BDR mevcutsa, bu bilgi de gösterilecektir. İşte bir örnek:

```
Corp#sh ip ospf neighbor
Neighbor ID  Pri  State    Dead Time    Address     Interface
10.1.11.1     0   FULL/   - 00:00:37    10.1.5.2     Serial0/2/0
10.1.9.1      0   FULL/   - 00:00:34    10.1.4.2     Serial0/1/0
10.1.7.1      0   FULL/   - 00:00:38    10.1.3.2     Serial0/0/1
10.1.7.1      0   FULL/   - 00:00:34    10.1.2.2     Serial0/0/0
```

Gerçek network'lerde çok kullanışlı olduğundan, bu, çok önemli bir komuttur. Şimdi, R3 ve 871W router çıktılarına bakalım:

```
R3#sh ip ospf neighbor
Neighbor ID  Pri  State   Dead Time    Address      Interface
10.1.5.1      0   FULL/  -  00:00:39    10.1.5.1    Serial0/0/1
10.1.11.2     1   FULL/BDR  00:00:31    10.1.11.2
FastEthernet0/1
871W#sh ip ospf nei
Neihbor ID   Pri  State   Dead Time    Address      Interface
10.1.11.1     1   FULL/DR   00:00:30    10.1.11.1     Vlan1
```

Corp router'ında bir Ethernet linki (broadcast multi-acceess) olduğundan, kimin DR kimin BDR olacağıyla ilgili bir seçim olacaktır. 871W'nun DR olduğunu ve network'teki en yüksek IP adresine sahip olduğundan, kazandığını görebiliriz. Bunu değiştirebilirsiniz, fakat varsayılanı budur.

Çıktıda Corp'un R1, R2 ve R3 bağlantılarının DR ve BDR'a sahip olmamasının sebebi, varsayılan olarak, point-to-point bağlantılarda seçimin olmamasıdır. Çıktıdan, Corp router'ının üç router'a da tamamıyla adjacent olduğunu görebiliriz.

## Show ip protocols Komutu

`Show ip protocols` komutu da, router'ınızda ister OSPF, EIGRP, RIP, BGP, IS-IS ister yapılandırabileceğiniz diğer routing protokolleri çalışsın, oldukça kullanışlı bir komuttur. Çalışan tüm protokollerin gerçek işleyişiyle ilgili mükemmel bir bakış sağlar.

Corp router'ının çıktısını kontrol edelim:

```
Corp#sh ip protocols
Routing Protocol is "ospf 132"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Router ID 10.1.5.1
  Number of areas in this router is 1. 1 normal 0 stub 0 nssa
  Maximum path: 4
  Routing for Networks:
    10.1.1.1 0.0.0.0 area 0
    10.1.2.1 0.0.0.0 area 0
    10.1.3.1 0.0.0.0 area 0
    10.1.4.1 0.0.0.0 area 0
    10.1.5.1 0.0.0.0 area 0
 Reference bandwidth unit is 100 mbps
  Routing Information Sources:
    Gateway         Distance      Last Update
    10.1.11.1            110      00:28:53
    10.1.11.2            110      00:28:53
    10.1.9.1             110      00:28:53
    10.1.7.1             110      00:28:53
         Distance: (default is 110)
```

Bu çıktıya bakarak, OSPF ProcessID, OSPF Router ID, OSPF area tipi, OSPF için yapılandırılmış network ve area'lar ile komşuların Router ID'lerini belirleyebilirsiniz. Daha önce bu komuttan RIP çıktılarında görünen timer'ların yokluğuna dikkat ettiniz mi? Link-state routing protokolleri, distance-vector routing algoritmalarının yaptığı gibi, network'ün kararlı çalışmasını sağlamak için timer'lar kullanmaz.

## Debugging OSPF

Debugging, tüm protokoller için mükemmel bir araçtır, bu nedenle OSPF hata tespiti için Tablo 7.4'teki bazı `debugging` komutlarına bakalım.

**Tablo 7.4:** OSPF Hata Tespiti İçin Debugging Komutları

| Komut | Açıklama/Fonksiyon |
|---|---|
| `debug ip ospf packet` | Router'ınızdaki gönderilip alınan Hello paketlerini gösterir |
| `debug ip ospf hello` | Router'ınızdaki gönderilip alınan Hello paketlerini gösterir. debug ip ospf packet komutundan daha fazla detay gösterir. |
| `debug ip ospf adj` | Broadcast ve non-broadcast multi-access network'lerindeki DR ve BDR seçimlerini gösterir. |

`Debug ip ospf packet` komutunu kullanarak, Corp router'dan alınan çıktıyı göstererek başlayacağım:

```
Corp#debug ip ospf packet
OSPF packet debugging is on
*Mar 23 01:20:42.199: OSPF: rcv. v:2 t:1 l:48 rid:172.16.10.3
      aid:0.0.0.0 chk:8075 aut:0 auk: from Serial0/1/0
Corp#
*Mar 23 01:20:45.507: OSPF: rcv. v:2 t:1 l:48 rid:172.16.10.2
      aid:0.0.0.0 chk:8076 aut:0 auk: from Serial0/0/0
*Mar 23 01:20:45.531: OSPF: rcv. v:2 t:1 l:48 rid:172.16.10.2
      aid:0.0.0.0 chk:8076 aut:0 auk: from Serial0/0/1
*Mar 23 01:20:45.531: OSPF: rcv. v:2 t:1 l:48 rid:172.16.10.4
      aid:0.0.0.0 chk:8074 aut:0 auk: from Serial0/2/0
*Mar 23 01:20:52.199: OSPF: rcv. v:2 t:1 l:48 rid:172.16.10.3
      aid:0.0.0.0 chk:8075 aut:0 auk: from Serial0/1/0
*Mar 23 01:20:55.507: OSPF: rcv. v:2 t:1 l:48 rid:172.16.10.2
      aid:0.0.0.0 chk:8076 aut:0 auk: from Serial0/0/0
*Mar 23 01:20:55.527: OSPF: rcv. v:2 t:1 l:48 rid:172.16.10.2
      aid:0.0.0.0 chk:8076 aut:0 auk: from Serial0/0/1
*Mar 23 01:20:55.531: OSPF: rcv. v:2 t:1 l:48 rid:172.16.10.4
      aid:0.0.0.0 chk:8074 aut:0 auk: from Serial0/2/0
```

Yukarıdaki çıktıda, her iki router'ımızın da, komşu router'lardan her 10 sn.'de Hello paketi alıp gönderdiğini görebiliriz. Şimdiki komut bize aynı çıktıyı, biraz daha detaylı sağlayacaktır. Örneğin, kullanılan (224.0.0.5) multicast adresini ve area'yı görebiliriz:

```
Corp#debug ip ospf hello
*Mar 23 01:18:41.103: OSPF: Send hello to 224.0.0.5 area 0 on
   Serial0/1/0 from 10.1.4.1
*Mar 23 01:18:41.607: OSPF: Send hello to 224.0.0.5 area 0 on
   FastEthernet0/1 from 10.1.1.1
*Mar 23 01:18:41.607: OSPF: Send hello to 224.0.0.5 area 0 on
   Serial0/0/0 from 10.1.2.1
*Mar 23 01:18:41.611: OSPF: Send hello to 224.0.0.5 area 0 on
```

```
   Serial0/2/0 from 10.1.5.1
*Mar 23 01:18:41.611: OSPF: Send hello to 224.0.0.5 area 0 on
   Serial0/0/1 from 10.1.3.1
*Mar 23 01:18:42.199: OSPF: Rcv hello from 172.16.10.3 area 0
from
   Serial0/1/0 10.1.4.2
*Mar 23 01:18:42.199: OSPF: End of hello processing
*Mar 23 01:18:45.519: OSPF: Rcv hello from 172.16.10.2 area 0
from
   Serial0/0/0 10.1.2.2
*Mar 23 01:18:45.519: OSPF: End of hello processing
*Mar 23 01:18:45.543: OSPF: Rcv hello from 172.16.10.2 area 0
from
   Serial0/0/1 10.1.3.2
*Mar 23 01:18:45.543: OSPF: End of hello processing
*Mar 23 01:18:45.543: OSPF: Rcv hello from 172.16.10.4 area 0
from
   Serial0/2/0 10.1.5.2
*Mar 23 01:18:45.543: OSPF: End of hello processing
```

Size göstereceğim son debug komutu, `debug ip ospf adj` size, broadcast ve non-broadcast multi-access network'lerde olan seçimleri gösterecektir:

```
Corp#debug ip ospf adj
OSPF adjacency events debugging is on
*Mar 23 01:24:34.823: OSPF: Interface FastEthernet0/1 going Down
*Mar 23 01:24:34.823: OSPF: 172.16.10.1 address 10.1.1.1 on
   FastEthernet0/1 is dead, state DOWN
*Mar 23 01:24:34.823: OSPF: Neighbor change Event on interface
   FastEthernet0/1
*Mar 23 01:24:34.823: OSPF: DR/BDR election on FastEthernet0/1
*Mar 23 01:24:34.823: OSPF: Elect BDR 0.0.0.0
*Mar 23 01:24:34.823: OSPF: Elect DR 0.0.0.0
*Mar 23 01:24:34.823: OSPF: Elect BDR 0.0.0.0
*Mar 23 01:24:34.823: OSPF: Elect DR 0.0.0.0
*Mar 23 01:24:34.823:         DR: none    BDR: none
*Mar 23 01:24:34.823: OSPF: Flush network LSA immediately
*Mar 23 01:24:34.823: OSPF: Remember old DR 172.16.10.1 (id)
*Mar 23 01:24:35.323: OSPF: We are not DR to build Net Lsa for
   interface FastEthernet0/1
*Mar 23 01:24:35.323: OSPF: Build router LSA for area 0, router
ID
   172.16.10.1, seq 0x80000006
*Mar 23 01:24:35.347: OSPF: Rcv LS UPD from 172.16.10.2 on
Serial0/0/1
   length 148 LSA count 1
*Mar 23 01:24:40.703: OSPF: Interface FastEthernet0/1 going Up
```

```
*Mar 23 01:24:41.203: OSPF: Build router LSA for area 0, router
ID
   172.16.10.1, seq 0x80000007
*Mar 23 01:24:41.231: OSPF: Rcv LS UPD from 172.16.10.2 on
Serial0/0/1
   length 160 LSA count 1
```

Devam edelim ve bir OSPF network'ünde seçimlerin nasıl olduğunu keşfedelim.

# OSPF DR ve BDR Seçimleri

Bu bölümde OSPF'i detaylı şekilde ele aldık, bununla beraber, şimdiye kadar kısaca bahsettiğim designated router ve backup designated router'larla ilgili bölümü açmaya gerek vardır. Ayrıca, hem seçim prosesini daha derin işleyeceğim hem de bu prosesi daha iyi anlamanıza yardımcı olması için modülün sonundaki pratik lab'ları vereceğim.

Başlamak için, DR ve BDR seçim prosesinde kritik öneme sahip olduklarından, neighbor ve adjacency terimlerini tamamıyla anladığınızdan emin olmam gerekiyor. Seçim prosesi, bir broadcast yada non-broadcast multi-access network'ü bir router'a bağlandığında yada link aktif olduğunda olur.(Ethernet yada Frame Relay'i düşünün).

## Neighbors

Ortak bir segmenti paylaşan router'lar, bu segmentte komşu olurlar. Bu komşular, Hello protokolü yoluyla seçilirler. Hello paketleri, IP multicast kullanarak her interface'in çıkışından periyodik olarak gönderilir.

İki router, şunlarda anlaşmaya varmadıkları sürece komşu olamazlar:

**Area ID:** Buradaki düşünce, iki router interface'inin, belirli bir segmentteki aynı area'da olmak zorunda olmasıdır. Ve tabii ki, bu interface'ler aynı segmente dahildir.

**Kimlik Denetimi:** OSPF, belirli bir area için şifre konfigürasyonunu mümkün kılar. Router'lar arasında kimlik denetimi gerekmemesine rağmen, bunu yapmak istediğinizde ayarlama seçeneğine sahipsinizdir. Ayrıca, router'ların komşu olmaları için kimlik denetimi kullanıyorsanız, bir segmentte aynı şifrenin kullanılması gerektiğini unutmayın.

**Hello ve Dead süreleri:** OSPF, her segmentte Hello paketleri gönderir. Bu, bir segmentte varlıklarını onaylamak ve broadcast ve non-broadcast multi-access segmentlerde bir designated router (DR) seçimi için router'ların kullandıkları keep-alive sistemidir.

Hello periyodu, Hello paketleri arasındaki süreyi saniye olarak belirtir. Dead aralığı, komşularının OSPF router'ı ölü (down) deklare etmeden önce, bir router'ın Hello paketlerinin gidebileceği saniye cinsinden süredir. OSPF, bu değerlerin, iki komşu arasında tamamıyla aynı olmasını gerektirir. Şayet bunlardan biri değişik olursa, router'lar, bu segmentte komşu olmayacaklardır. Bu timer'ları, `show ip ospf interface` komutu ile görebilirsiniz.

## Adjacencies

Seçim prosesinde, adjacency, komşuluk prosesinden sonraki adımdır. Adjacent router'lar, basit Hello değişiminin ötesine geçen ve veritabanı değişim prosesini çalıştıran router'lardır. Belirli bir segmentte değiştirilen bilgi miktarını minimuma indirmek için OSPF, her multi-access segmentteki bir router'ı designated router (DR) ve bir router'ı da backup designated router (BDR) olarak seçer.

BDR, DR'ın arızalanması durumunda yedek bir router olarak seçilir. Bu düşüncenin arkasındaki, router'ların bilgi değişimi için merkezi bir irtibat noktasına sahip olmasıdır. Her router'ın güncelle-

meleri segmentteki diğer router'larla değiştirmesi yerine, her router bilgiyi, DR ve BDR ile değiştirir. Sonra, DR ve BDR diğer herkese yayar.

### DR ve BDR Seçimi

DR ve BDR seçimi, Hello paketleri yardımıyla tamamlanır. Hello paketleri, her segmentteki IP multicast paketleri ile değiştirilir. Bununla beraber, sadece broadcast ve non-broadcast multi-access network segmentleri (Ethernet ve Frame Relay gibi), DR ve BDR seçimi çalıştıracaktır. Örneğin, seri WAN gibi point-to-point linkler, DR seçim prosesine sahip olmayacaktır.

Broadcast ya da non-broadcast multi-access network'te, bir segmentteki en yüksek OSPF priority'sine sahip router, bu segment için DR olacaktır. Bu priority, `show ip ospf interface` komutu ile görülebilir. Varsayılan olarak 1'e ayarlanmıştır. Şayet tüm router'lar, varsayılan priority'e sahipse en yüksek Router ID'ye (RID) sahip router seçimi kazanacaktır.

Bildiğiniz gibi RID, OSPF'in başlaması sırasında herhangi bir interface'deki en yüksek IP adresi ile belirlenir. Bu, bir loopback (mantıksal) interface tarafından geçersiz kılınabilir (sonraki bölümde bahsedeceğim).

Şayet router interface'inin priority'sini 0'a ayarlarsanız, bu router interface'i, DR ve BDR seçiminde yer almayacaktır. Sıfır(0) priority'li interface'in durumu, DROTHER olacaktır.

Şimdi, bir OSPF router'ındaki RID ile oynayalım.

## OSPF ve Loopback Interface'leri

OSPF routing protokolü kullandığınızda, loopback interface'i yapılandırmak önemlidir ve Cisco, bir router'da her OSPF yapılandırdığınızda onları kullanmanızı tavsiye eder.

Loopback interface'leri sanal, yazılımsal olan mantıksal interface'lerdir. Onlar gerçek router interface'leri değildir. OSPF konfigürasyonu ile loopback interface'leri kullanmak, OSPF prosesi için bir interface'in daima aktif olmasını garanti eder.

Onları hem sistem tanı amacı hem de OSPF konfigürasyonu için kullanabilirsiniz. Bir router'da loopback interface'i yapılandırmayı istemenizin sebebi, şayet yoksa bir router'da en yüksek IP adresinin, router'ın RIDi olacağıdır. RID, hem route'ları yayınlamak hem de DR ve BDR seçmek için kullanılır.

NOT *OSPF, varsayılan olarak OSPF'in başlaması sırasında aktif bir interface'indeki en yüksek IP adresini kullanır. Bununla beraber, bu mantıksal bir interface ile geçersiz kılınabilir. Mantıksal bir interface'in en yüksek IP adresi, daima bir router'ın RID'i olacaktır.*

Şimdiki bölümde, loopback interface'inin nasıl yapılandırıldığını ve loopback adresleri ve RID'lerin nasıl doğrulandığını göreceksiniz.

### Loopback Interface'lerinin Konfigürasyonu

OSPF konfigürasyonunun en kolay bölümü olduğundan loopback interface'lerini yapılandırmak çoğu zaman ürkütür. Hepimizin bir araya ihtiyacı var, değil mi? Bekleyin, son bölümdeyiz!

İlk olarak, `show ip ospf` komutu ile Corp router'ındaki RID'in ne olduğuna bakalım:

```
Corp#sh ip ospf
 Routing Process "ospf 132" with ID 10.1.5.1
[output cut]
```

RID'in, 10.1.5.1 ya da router'ın serial0/2/0 interface'i olduğunu görebiliriz. Tamamıyla farklı IP adresleme sistemi kullanarak bir loopback interface'ini yapılandıralım:

```
Corp(config)#int loopback 0
*Mar 22 01:23:14.206: %LINEPRAUTO-5-UPDOWN: Line protocol on
Interface
```

```
   Loopback0, changed state to up
Corp(config-if)#ip address 172.16.10.1 255.255.255.255
```

Burada, IP adresleme sistemi gerçekten önemli değildir. Fakat her router, ayrı bir subnette olmalıdır. /32 maskını kullanarak, adresler herhangi iki router'da aynı olmadığı sürece, istediğimiz IP adresini kullanabiliriz.

Gelin, diğer router'ları yapılandıralım:

```
R1#config t
R1(config)#int loopback 0
*Mar 22 01:25:11.206: %LINEPRAUTO-5-UPDOWN: Line protocol on
Interface
   Loopback0, changed state to up
R1(config-if)#ip address 172.16.10.2 255.255.255.255
Here's the configuration of the loopback interface on R2:
R2#config t
R2(config)#int loopback 0
*Mar 22 02:21:59.686: %LINEPRAUTO-5-UPDOWN: Line protocol on
Interface
   Loopback0, changed state to up
R2(config-if)#ip address 172.16.10.3 255.255.255.255
```

R3'teki loopback interface konfigürasyonu şöyledir:

```
R3#config t
R3(config)#int loopback 0
*Mar 22 02:01:49.686: %LINEPRAUTO-5-UPDOWN: Line protocol on
Interface
   Loopback0, changed state to up
R3(config-if)#ip address 172.16.10.4 255.255.255.255
```

871W router'ında mantıksal interface ayarlamasını kullanmayacağım. Birazdan sebebini anlayacaksınız.

255.255.255.255 (/32) IP adres maskının ne olduğunu ve onun yerine neden 255.255.255.0 (/24) maskını kullanmadığımızı merak ettiğinize eminim. Her iki mask da çalışır, fakat /32 maskı bir host maskı olarak tanımlanır ve loopback interface'leri için daha iyi çalışır. Ayrıca, subnetlerden kazanmamızı sağlar. 172.16.10.1, .2, .3 ve. 4'ü nasıl kullanabildiğime dikkat ettiniz mi? Şayet /32 maskı kullanmasaydım, her router için ayrı subnet kullanmak zorunda kalacaktım!

Şimdi, devam etmeden önce, loopback interface'i ayarlayarak, router'larımızın RID'lerini gerçekten değiştirebilir miyiz? Corp'un RID'ine bakarak bunu kontrol edelim:

```
Corp#sh ip ospf
 Routing Process "ospf 132" with ID 10.1.5.1
```

Ne oldu? Mantıksal interface oluşturduğumuzdan, mantıksal interface'deki IP adresinin, router'ın RID'i olduğunu düşünüyorsunuzdur, değil mi? Öyle sayılır, fakat sadece birkaç şey yapmanız gerekirr: Ya router'ı reboot edeceksiniz ya da OSPF'i silip router'ınızdaki veritabanını tekrar oluşturacaksınız. İkisi de oldukça iyi seçenektir.

Daha kolay olduğundan, Corp router'ını reboot edeceğim.

Şimdi bakalım ve RID'in ne olduğunu görelim:

```
Corp#sh ip ospf
 Routing Process "ospf 132" with ID 172.16.1.1
```

Şimdi oldu. Corp router'ı şimdi yeni bir RID'e sahip! Sanırım, gideceğim ve RID'lerini, mantıksal adresleri ile değiştirmeleri için 871W hariç tüm router'larımı reboot edeceğim.

İstemiyorsanız, bir yol daha var. Router için yeni bir RID eklemek için `router ospf process-id` komutunu kullanmaya ne dersiniz? Deneyelim derim! 871W router'da, bununla ilgili örneği görebilirsiniz:

```
871W#sh ip ospf
 Routing Process "ospf 1" with ID 10.1.12.1
871W#config t
871W(config)#router ospf 1
871W(config-router)#router-id 172.16.10.5
Reload or use "clear ip ospf process" command, for this to take
effect
871W(config-router)#do clear ip ospf process
Reset ALL OSPF processes? [no]: yes
*Mar 23 01:33:00.051: OSPF: Rcv LS UPD from 172.16.10.4 on
Serial0/2/0
   length 76 LSA count 1
*Mar 23 01:33:00.071: OSPF: Rcv LS UPD from 172.16.10.2 on
Serial0/0/1
   length 76 LSA count 1
871W(config-router)#do sh ip ospf
Routing Process "ospf 1" with ID 172.16.10.5
```

Şuna bakın, çalıştı! Router'ı reload etmeden RID'si değişti! Fakat bekleyin, bizim henüz bir loopback oluşturmadığımızı hatırlayın. Şimdi deneyelim, mantıksal interface IP adresi ayarlayalım, router'ı reload edelim ve loopback interface'inin, şimdi kullandığımız `router-id` komutunu geçersiz kılıp kılmadığına bakalım:

```
871W(config-router)#int lo0
871W(config-if)#ip address 172.16.10.6 255.255.255.255
871W(config-if)#^Z
871W#reload
System configuration has been modified. Save? [yes/no]: y
Building configuration...


871W#sh ip ospf
 Routing Process "ospf 1" with ID 172.16.10.5
```

İşte cevabımız. Bir mantıksal (loopback) interface'i, `router-id` komutunu geçersiz kılmayacaktır ve biz, RID olarak etkilemesi için router'ı reboot etmek zorunda kalmayacağız.

Şimdi kalan tek şey, loopback interface'inin, OSPF altında yayınlanmasını isteyip istemediğinize karar vermektir. Yayınlanmayacak bir adresin kullanılmasına karşı, kullanılmasını savunan lehte ve aleyhte görüşler vardır. Yayınlanmayan bir adres kullanımı gerçek IP adres uzayını korur fakat adres, OSPF routing tablosunda görünmeyecektir, yani ping'lenemeyecektir.

Basit olarak, burada karşı karşıya kaldığınız durum, network'te hata ayıklamayı kolaylaştırma ile adres uzayını koruma arasında bir tercih yapmak zorunda kalmanızdır. Gerçekten iyi bir strateji, benim yaptığım gibi private IP adres düzenlemesi kullanmaktır. Bunu yapın, her şey güzel olacaktır.

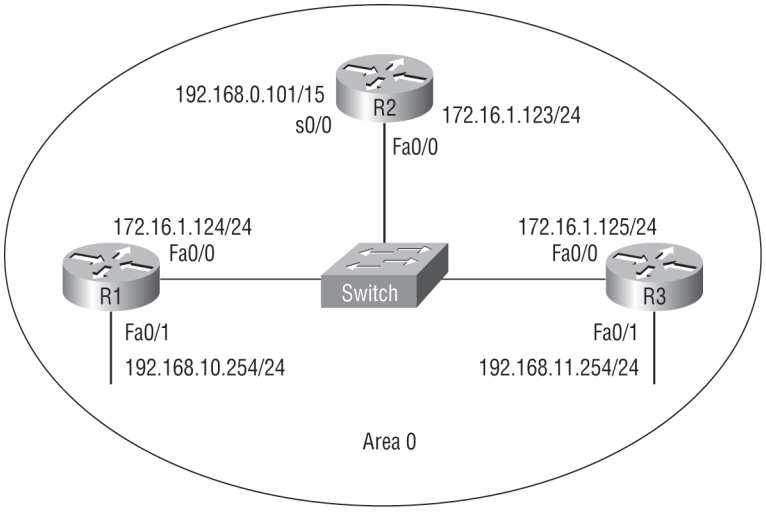
## OSPF Interface Priority'leri

OSPF teki DR ve BDR'ları yapılandırmanın diğer yolu, loopback interface'i kullanmak yerine seçimleri ayarlamaktır. Bunu, seçim olduğunda, diğer router'lar karşısında daha iyi bir priority'e sahip olması için router'ımızdaki interface'leri yapılandırarak yapabiliriz. Diğer bir deyişle, bir network'te belirli bir router'ın DR veya BDR olmasını zorlamak için mantıksal adresler yerine priority'leri kullanabiliriz.

Şekil 7.6'yı örnek olarak kullanalım. Şekil 7.6'ya bakarak, LAN (broadcast multi-access segmenti) için R2 router'ının designated router (DR) olarak seçilmesini garantilemek için hangi seçenekleri kullanırdınız? Yapmanız gereken ilk şey, her router'ın RID'sini ve hangi router'ın, 172.16.1.0 LAN'ı için varsayılan DR olduğunu belirlemektir.

Bu noktada, en yüksek RID olan 192.168.11.254'e sahip olduğundan, R3 router'ı varsayılan olarak, DR'dır. Bu bize, R2 router'ının, 172.16.1.0/24 LAN segmenti için DR seçilmesini garantilemek için üç seçenek verir:

- R2 router'ının Fa0/0 interface'inin priority değerini, Ethernet network'ündeki diğer interface'lerden daha yüksek bir değere ayarlayın.
- R2 de bir loopback interface'ini, diğer router'lardaki IP adreslerinden daha yüksek bir IP adresiyle yapılandırın.
- R1 ve R3'ün interface'lerinin priority değerini sıfır olarak değiştirin.




***Şekil 7.6:*** *Designated router'ınızdan emin olmak.*

Şayet R1 ve R3 router'larınızdaki priority'i sıfıra ayarlarsak, seçim prosesine katılmalarına izin verilmeyecektir. Fakat bu en iyi yol olmayabilir, bir ve iki seçenekleri bizim için daha iyi olabilir.

Bir loopback (mantıksal) interface'in nasıl yapılandırılacağını zaten bildiğinizden, burada R2 router'ındaki Fa0/0 interface'indeki priority'i nasıl ayarlanacağı vardır:

```
R2#config t
R2(config)#int f0/0
```

```
R2(config-if)#ip ospf priority ?
  <0-255>  Priority
R2(config-if)#ip ospf priority 2
```

Tüm router interface'lerinin priority'leri 1'e ayarlandı, bu interface'i 2'ye ayarlayarak, onun otomatik olarak LAN segmentinin DR ve BDR'ı olacağını garantiye aldım. Bir interface'in 255'e ayarlanmasının anlamı, router'ınızı kimsenin yenemeyeceğidir.

Interface'in priority'sini değiştirseniz bile, mevcut DR ve BDR kapatılana kadar router LAN segmentinin DR'ı olmayacaktır. Bir seçim olunca, DR ve BDR reload olana ve /veya kapanana kadar tekrar seçim olmayacaktır. Network'ünüzde daha iyi bir RID'le ortaya çıkan bir router'a sahip olmak, DR veya BDR'ın değişeceği anlamına gelmez!

NOT *Broadcast ya da non-broadcast multi-access network'ünüzde olan seçimleri, debug ip ospf adj komutu ile görebileceğinizi hatırlayın.*

Priority'nizi, show ip ospf interface komutu ile görebilirsiniz:

```
R2(config-if)#do show ip ospf int f0/0
FastEthernet0/0 is up, line protocol is up
  Internet Address 10.1.13.1/24, Area 0
  Process ID 132, Router ID 172.16.30.1, Network Type
BROADCAST,Cost:1
  Transmit Delay is 1 sec, State UP, Priority 2
```

# OSPF Hata Tespiti

Bu bölüm, OSPF bağlantılı problemleri tespit etmek ve düzeltmek için örnek OSPF konfigürasyonlarını ve konfigürasyon çıktılarının doğruluğunu kontrol etmenizi sağlayacaktır.

Şayet, burada göründüğü gibi bir konfigürasyon görürseniz, wildcard yanlış olduğundan, router'ın bu kaydı kabul etmekten başka çaresi olmadığını bilmelisiniz:

```
Router(config)#router ospf 1
Router(config-router)#network 10.0.0.0 255.0.0.0 area 0
Doğrusu şöyle olacaktır:
Router(config)#router ospf 1
Router(config-router)#network 10.0.0.0 0.255.255.255 area 0
```

Şekle bakalım ve router'lardan hangisinin, area'nın designated router'u olduğunu görelim. Şekil 7.7, iki switch ve bir WAN linki ile bağlı altı router olan bir network'ü göstermektedir.

Şekil 7.7'ye bakarak, hangi router, designated router (DR) seçilmeye uygundur? OSPF router'larının hepsi varsayılan priority değerine sahiptir.

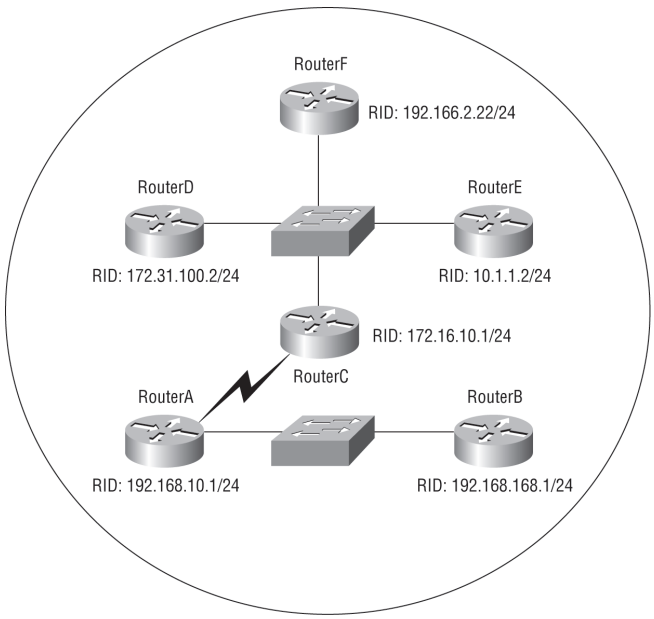
Her router'ın RID'ine dikkat edin. En yüksek IP adreslerine sahip olduklarından en yüksek RID'li router'lar, Router A ve Router B'dir. Varsayılan olarak, point-to-point linklerde seçim olmadığından üstteki LAN kendi seçimini yapacaktır. Fakat CCNA sınav konuları için bunu okuduğunuzdan, Router B, en iyi cevap olacaktır.

OSPF konfigürasyonunun doğruluğunu kontrol etmek için başka bir komut kullanalım: `show ip ospf interface` komutu. Aşağıdaki çıktıya bakın ve direk bağlı router'ların neden bir adjacency kuramayacaklarını belirlemeye çalışın:

```
RouterA#sh ip ospf interface e0/0
Ethernet0/0 is up, line protocol is up
  Internet Address 172.16.1.2/16, Area 0
```

```
  Process ID 2, Router ID 172.126.1.1, Network Type BROADCAST,
Cost: 10
  Transmit Delay is 1 sec, State DR, Priority 1
  Designated Router (ID) 172.16.1.2, interface address 172.16.1.1
  No backup designated router on this network
  Timer intervals configured, Hello 5, Dead 20, Wait 20,
Retransmit 5

RouterB#sh ip ospf interface e0/0
Ethernet0/0 is up, line protocol is up
  Internet Address 172.16.1.1/16, Area 0
  Process ID 2, Router ID 172.126.1.1, Network Type BROADCAST,
Cost: 10
  Transmit Delay is 1 sec, State DR, Priority 1
  Designated Router (ID) 172.16.1.1, interface address 172.16.1.2
  No backup designated router on this network
  Timer intervals configured, Hello 10, Dead 40, Wait 40,
Retransmit 5
```

İki çıktıdaki her şey, Hello ve Dead timer'larının farklı olmaları dışında, oldukça güzel görünmektedir. Router A, Hello ve Dead timer için, 5 ve 20'ye sahipken, Router B, OSPF için varsayılan olan 10 ve 40 sürelerine sahiptir. Şayet iki direk bağlı router, aynı timer'lara sahip değilse, bir adjacency oluşturamayacaklardır. Show ip ospf interface komutunun, area'nız için kimlerin, designated ve backup designated router (DR/BDR) olduklarını göstereceğine dikkat edin.




***Şekil 7.7:*** *Designated router örneği.*

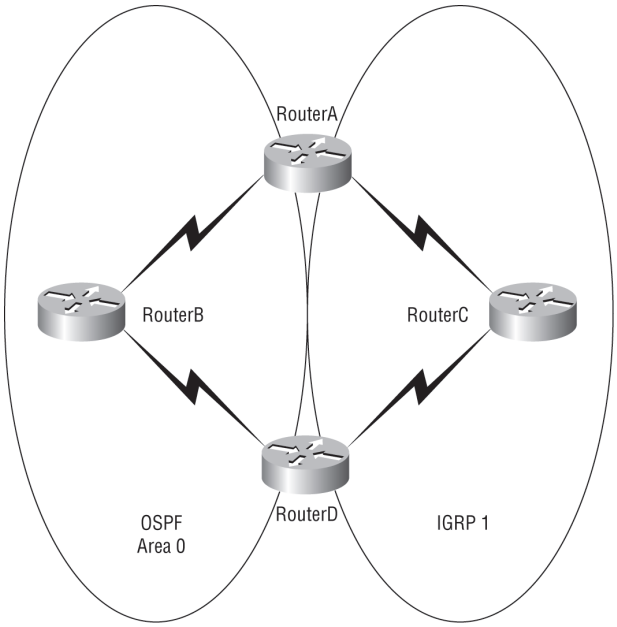
Şekil 7.8'deki, dört router ve iki farklı routing protokolü çalışan network'e bakalım.

Şayet tüm parametreler varsayılan olarak ayarlandıysa ve redistribution yapılandırılmadıysa, Router A'nın, Router D'ye erişmek için hangi yolu kullanacağını düşünürsünüz? IGRP, 100 ve OSPF, 110 AD değerine sahip olduğundan Router A paketleri Router C üzerinden Router D'ye gönderecektir.

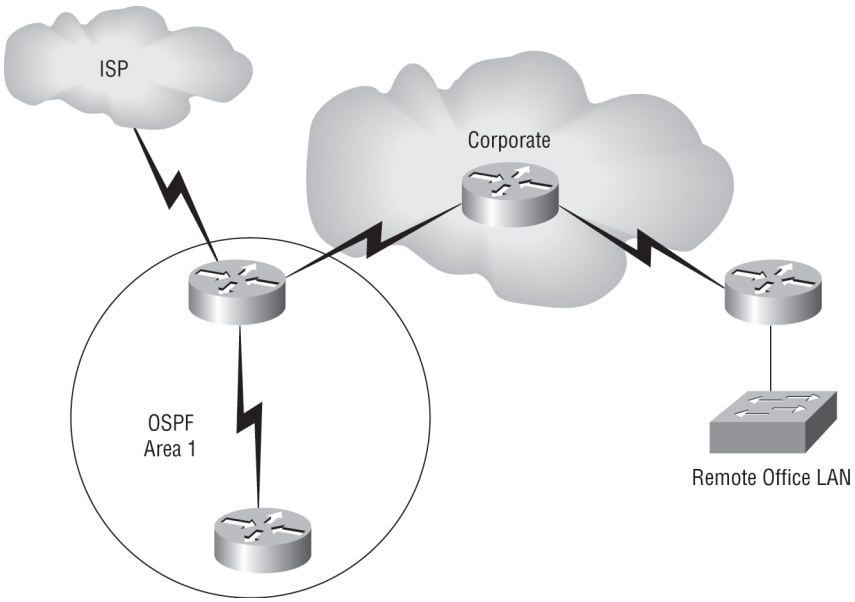
Şekil 7.9'u dikkatle çalışın. Şekilde göründüğü gibi router'larda OSPF çalıştırıyorsunuz ve bir ISDN linki uzaktaki satış ofisine bağlantı sağlamaktadır.

Şekil 7.9'da görünen ISDN linkindeki network yükünü minimuma indirirken, satış ofisinin uzak network'üne bağlanmak için Corporate router'ında nasıl bir route yapılandırılmalıdır?

Bu problem için en iyi çözüm, ISDN linkini iptal etmek, uzak ofisten Internete geniş bant bir link bağlamak ve sonra Corporate ofisinden uzak ofise internet üzerinden bir VPN oluşturmaktır. Daha iyi olmaz mı? Her neyse, soru bunu ISDN linki ile nasıl çalışır hale getirebileceğimizi ve network yükünü minimuma nasıl indirebileceğimizi sormaktadır. Bunu yapabilmemizin tek yolu, uzak network'e bağlanmak için Corporate router'ında bir statik route oluşturmaktır. Bunun dışındaki çözümler yüksek miktarlarda bant genişliği kullanacaktır.




***Şekil 7.8:*** *Çoklu routing protokolleri ve OSPF.*




***Şekil 7.9:*** *OSPF ve ISDN bağlanabilirliği.*

# EIGRP ve OSPF Summary Route'larını Yapılandırmak

Bu bölüm size, hem EIGRP hem de OSPF'i summarize etmek için kullanılan komutları gösterecektir. OSPF, birkaç farklı yolla summarize edilmesine rağmen, en yaygın OSPF summary komutunu göstereceğim. Bu, çok area'lı OSPF network'lerini area 0'a summarize eder.

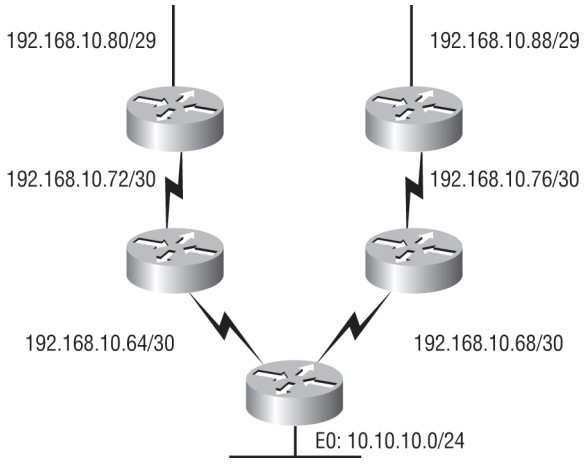
Bir network için özet route'ların nasıl belirlendiğini, bölüml 3'te öğrendiniz. Bu bölüm size özet route'ların, bir router konfigürasyonunda uygulanmasını gösterecektir.

Şekil 7.10, bir ardışık network tasarımını göstermektedir. Ardışık network'ler şansa olmazlar, onların planlanması gerekir! Şekil 7.10, 4 blok boyutunda (WAN linkleri) dört ve 8 blok boyutunda (LAN bağlantıları) iki network'ü göstermektedir. Bu network tasarımı, 32 boyutunda bir bloğa rahatlıkla sığar. Kullanılan network adresi, 192.168.10.64'tür ve 32 blok boyutundadır. Mask, 255.255.255.224 olmalıdır, çünkü bildiğiniz gibi 224, 32 blok boyutu sağlar.

Merkezdeki (omurga bağlantısı) router'da, EIGRP için Ethernet0'da summary route yerleştireceğiz. Ethernet0, summary route'umuzu, omurga network'e (10.10.10.0 network'üne) yayınlayacaktır. Bu, altı network'ümüzün teker teker yayınlanmasını durdurur ve onun yerine ağ topluluğundaki diğer router'lara tek bir route olarak yayınlanmasını sağlar. Bununla beraber, ardışık network'lerimiz dışında hiçbir router'ın, bu yayınlanmış blokta bir subnet'e sahip olmaması önemlidir. Yoksa bu çakışan route'ların yayınlanmasına izin verir.

Merkez router'daki komple EIGRP konfigürasyonu şöyledir:

```
Core#config t
Core(config)#router eigrp 10
Core(config-router)#network 192.168.10.0
Core(config-router)#network 10.0.0.0
Core(config-router)#no auto-summary
Core(config-router)#interface ethernet 0
Core(config-if)#ip summary-address eigrp 10 192.168.10.64
255.255.255.224
```




***Şekil 7.10:*** *Ardışık network tasarımı.*

Autonomous system 10 için yukarıdaki EIGRP konfigürasyonu, 192.168.10.0 ve 10.0.0.0 direk bağlı network'lere yayınlanır. EIGRP, classful sınırlarda auto-summarization yaptığından, no `auto-summary` komutunu da kullanmak zorundasınız. Omurga network'e yayınlayacağımız summary route, omurgaya bağlı interface'e yazılmalıdır, routing prosesinin altına değil. Bu özet route, EIGRP'nin 32 blok boyutlu 192.168.10.64 network'ündeki tüm network'leri bulacağını ve onları interface0'dan tek bir route gibi yayınlayacağını söyler. Yani basit olarak, 192.168.10.64'ten

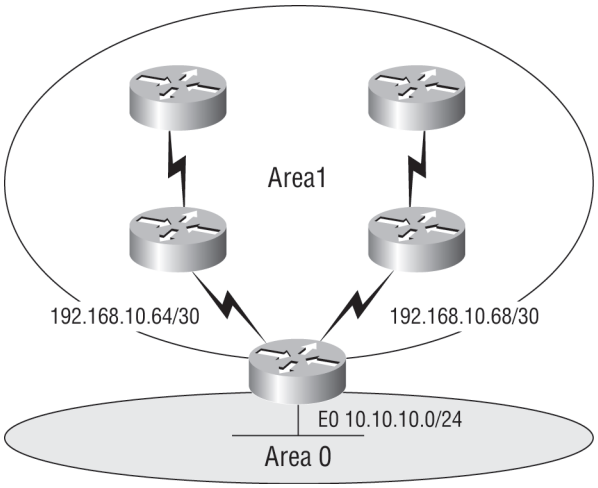
192.168.10.95'e kadar hedef adresine sahip herhangi bir paket, bu özet route yardımıyla gönderilecektir.

EIGRP örneğinde kullandığımız ardışık network'ü, OSPF ile summarize etmek için, şekil 7.11'de gösterildiği gibi, OSPF'i çok sayıda area ile yapılandırmamız gerekir.

Area1'i area 0'a summarize etmek için, OSPF Process ID altında, aşağıdaki komutu kullanın. Core (omurga) router için komple OSPF konfigürasyonu şöyledir:

```
Core#config t
Core(config)#router ospf 1
Core(config-router)#network 192.168.10.64 0.0.0.3 area 1
Core(config-router)#network 192.168.10.68 0.0.0.3 area 1
Core(config-router)#network 10.10.10.0 0.0.0.255 area 0
Core(config-router)#area 1 range 192.168.10.64 255.255.255.224
```

Varsayılan olarak, OSPF, herhangi bir sınırda summarize yapmadığından, no auto-summary kullanmaya gerek yoktur. Yukarıdaki OSPF konfigürasyonu, area 1'den tüm network'leri, omurga area'ya bir 192.168.10.64/27 kaydı olarak summarize edecektir.




***Şekil 7.11:*** *OSPF çoklu area tasarımı.*

# Özet

Bu bölümün kapsamlı ele alındığını söyleyebileceğinizi biliyorum. Fakat gerçekten çok önemlidir! Bölümün ana odağı olan EIGRP, link-state ve distance vector protokollerinin karmasıdır. Eşit olmayan cost değerli yük dengelemesini, kontrollü routing güncellemelerini ve komşu adjacency'leri sağlar.

EIGRP, komşular arasında iletişim için, Transport Protocol (RTP) özelliğini kullanır ve tüm uzak netwoklere en iyi yolu hesaplamak için Diffusing Update Algorithm'den (DUAL) faydalanır.

EIGRP ayrıca, VLSM, discontigious (ardışık olmayan) network'ler ve summarization desteği gibi özelliklerle büyük network'leri destekler. NBMA network'lerinde EIGRP konfigürasyon kabiliyeti onu, büyük network'ler için çok kullanışlı yapar.

Ayrıca EIGRP konfigürasyonunu inceledim ve birçok hata tespit komutu keşfettim.

Bu bölüm, OSPF hakkında da çok sayıda bilgi sağladı. OSPF hakkında her şeyi içermesi gerçekten zordur. Çünkü birçok konu bu bölüm ve kitabın kapsamı dışındadır. Fakat size bazı yerlerde

ipuçları verdim, yani size sunduklarımı hatırlayacağınızdan emin olduğunuz sürece iyi durumdasınız.

Hem terminoloji, operasyonlar ve konfigürasyon hem de doğrulama ve görüntülemeyi içeren birçok OSPF konusundan bahsettim.

Bu konuların hepsi, çok sayıda bilgi içermektedir (terminoloji bölümü, OSPF'ten yüzeysel bahsetmektedir). Fakat single-area OSPF, VLSM uygulaması ve ardışık sınırların summarize edilmesi gibi çalışmalarınızda başarılı olmanız gerekmektedir. Son olarak, OSPF operasyonunda kullanışlı komutların detaylı incelenmesini verdim. Böylece, işlerin olması gerektiği gibi gittiğini doğrulayabiliyorsunuz. Onların hepsini adınız gibi bilin, artık hazırsınız!

# Sınav Gereklilikleri

**EIGRP özelliklerini bilmek:** EIGRP, IP, IPX, AppleTalk ve şimdide IPv6'yı destekleyen bir classless, gelişmiş distance-vector protokolüdür. EIGRP, route bilgisini devam ettirmek için DUAL denilen benzersiz algoritma ve EIGRP router'larının diğerleriyle güvenli haberleşmesi için RTP kullanır.

**EIGRP'nin nasıl yapılandırıldığını bilmek:** Temel EIGRP'yi yapılandırabilin. Bu, classful adresleriyle, aynı şekilde yapılandırılır.

**EIGRP operasyonunun nasıl doğrulanacağını bilmek:** Tüm EIGRP show komutlarını bilin ve çıktıları ile çıktılarının ana bileşenlerinin yorumuna aşina olun.

**OSPF ve RIPv1'i mukayese etmek:** OSPF, VLSM ve classless routing'i destekleyen link-state protokoldür. RIPv1, VLSM'i desteklemeyen, distance vector protokoldür ve sadece classful routing'i destekler.

**OSPF router'ların nasıl komşu ve/veya adjacent olduklarını bilmek:** Her router diğerinin Hello paketlerini gördüğünde, OSPF router'lar komşu olurlar.

**Farklı OSPF NBMA tiplerini bilmek:** Cisco router'ların desteklemek için yapılandırılabileceği beş farklı OSPF network tipi vardır. Bunlardan iki tanesi, marka bağımsız (non-broadcast ve point-to-point) ve üç tanesi Cisco'nun tescilli network tipleridir (broadcast, point-to-point ve point-to-multipoint non-broadcast). Her network tipi, router'ların nasıl adjacent olacakları ve DR/BDR seçimi gerektirip gerektirmeyeceğine göre tanımlanmaktadır.

**Single-area OSPF'i yapılandırabilmek:** Minimum bir single-area konfigürasyonu sadece iki komutu içerir: `router ospf` ***`process-id`*** ve ***`network x.x.x.x y.y.y.y`*** `area Z`.

**OSPF operasyonunu doğrulayabilmek:** OSPF'te faydalı detaylara erişmek için kullanılan birçok show komutu vardır ve çıktılarına tamamıyla aşina olmak çok faydalıdır: `show ip ospf`, `show ip ospf database`, `show ip ospf interface`, `show ip ospf neighbor` ve `show ip protocols`.

# Yazılı Lab 7

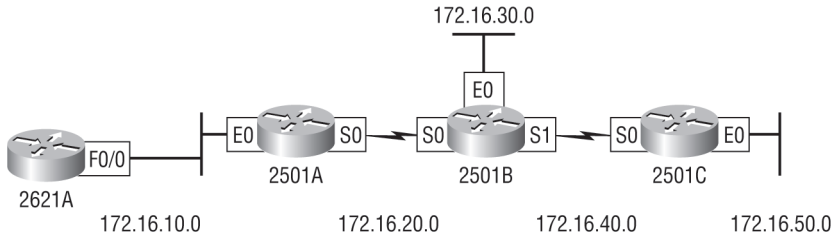
1. EIGRP tarafından desteklenen dört routed protokol nedir?
2. Redistribution, EIGRP için ne zaman gerekmektedir?
3. Hangi komut, 300 autonomous sistem numarasına sahip EIGRP'yi etkinleştirmek için kullanılmaktadır?
4. Hangi komut EIGRP'ye, 172.10.0.0 network'üne bağlı olduğunu söyler?
5. Hangi tip EIGRP paketi, ne Hello paketi gönderecektir ne de alacaktır?
6. Bir router'da, OSPF process 101'i etkinleştirecek komutu yazın.

7. Bir router'da, etkinleştirilen tüm OSPF routing proseslerini gösterecek komutu yazın.
8. Interface'e bağlı OSPF bilgilerini gösterecek komutu yazın.
9. Tüm OSPF komşularını gösterecek komutu yazın.
10. Router tarafından bilinen farklı OSPF route tiplerini yazın.

(Yazılı lab7'nin cevapları, bu bölüm için gözden geçirme sorularına cevaptan sonra bulunabilir.)

# Pratik Lab'lar

Bu bölümde, aşağıdaki network'leri kullanacaksınız ve EIGRP ile OSPF routing'i ekleyeceksiniz.




İlk lab (Lab 7.1), üç router'ı, EIGRP ile yapılandırmanızı ve konfigürasyonu gözden geçirmenizi gerektirir. Son dört lab'da aynı network'te, OSPF network'ü etkinleştirmeniz istenecektir. Bu modüldeki lab'lar, gerçek ekipmanlarla kullanılmak için yazılmıştır.

NOT *Routing protokolleri, OSPF'ten daha düşük administrative distance'a sahip olduğundan, 7.2-7.4 Lab'larına başlamadan önce, EIGRP'yi kaldırmalısınız.*

Bu modüldeki lab'lar şunlardır:

Lab 7.1: EIGRP'yi Yapılandırmak ve Doğrulamak

Lab 7.2: OSPF Prosesini Etkinleştirmek.

Lab 7.3: OSPF Komşularını Yapılandırmak

Lab 7.4: OSPF Operasyonlarını Doğrulamak

Lab 7.5: OSPF DR ve DBR Seçimleri

Tablo 7.5, her router için IP adreslerimizi gösterir (her interface, /24 mask kullanır).

**Tablo 7.5:** IP Adreslerimiz

| Router | Interface | Ip Adresleri |
|---|---|---|
| 2621 | F0/0 | 172.16.10.1 |
| 2501A | E0 | 172.16.10.2 |
| 2501A | S0 | 172.16.20.1 |
| 2501B | E0 | 172.16.30.1 |
| 2501B | S0 | 172.16.20.2 |
| 2501B | S1 | 172.16.40.1 |
| 2501C | S0 | 172.16.40.2 |
| 2501C | E0 | 172.16.50.1 |

## Pratik Lab 7.1: EIGRP'yi Yapılandırmak ve Doğrulamak

1. 2621A'da EIGRP çalıştırın:

```
2621A#conf t
Enter configuration commands, one per line.
  End with CNTL/Z.
2621A(config)#router eigrp 100
2621A(config-router)#network 172.16.0.0
2621A(config-router)#^Z
2621A#
```

2. 2501A'da EIGRP çalıştırın:

```
2501A#conf t
Enter configuration commands, one per line.
  End with CNTL/Z.
2501A(config)#router eigrp 100
2501A(config-router)#network 172.16.0.0
2501A(config-router)#exit
2501A#
```

3. 2501B'de EIGRP çalıştırın:

```
2501B#conf t
Enter configuration commands, one per line.
  End with CNTL/Z.
2501B(config)#router eigrp 100
2501B(config-router)#network 172.16.0.0
2501B(config-router)#^Z
2501B#
```

4. 2501C'de EIGRP çalıştırın:

```
2501C#conf t
Enter configuration commands, one per line.
  End with CNTL/Z.
2501C(config)#router eigrp 100
2501C(config-router)#network 172.16.0.0
2501C(config-router)#^Z
2501C#
```

5. 2501B için topoloji tablosunu gösterin:

```
2501B#show ip eigrp topology
```

6. 2501B router'daki routing tablosunu gösterin:

```
2501B#show ip route
```

7. 2501B router'daki routing tablosunu gösterin:

```
2501B#show ip eigrp neighbor
```

## Pratik Lab 7.2: OSPF Prosesini Etkinleştirmek.

1. 2621A'da OSPF process 100'ü etkinleştirmek:

```
2621A#conf t
Enter configuration commands, one per line.
  End with CNTL/Z.
2621A(config)#router ospf 100
2621A(config-router)#^Z
```

2. 2501A'da OSPF process 101'i etkinleştirmek:

```
2501A#conf t
Enter configuration commands, one per line.
  End with CNTL/Z.
2501A(config)#router ospf 101
2501A(config-router)#^Z
```

3. 2501B'de OSPF process 102'yi etkinleştirmek:

```
2501B#conf t
Enter configuration commands, one per line.
  End with CNTL/Z.
2501B(config)#router ospf 102
2501B(config-router)#^Z
```

4. 2501C'de OSPF process 103'ü etkinleştirmek:

```
2501C#conf t
Enter configuration commands, one per line.
  End with CNTL/Z.
Router(config)#router ospf 103
2501C(config-router)#^Z
```

## Pratik Lab 7.3: OSPF Komşularını Yapılandırmak

1. 1. 2621A ve 2501A arasındaki network'ü yapılandırın. Onu, area 0'a atayın:

```
2621A#conf t
Enter configuration commands, one per line.
  End with CNTL/Z.
2621A(config)#router ospf 100
2621A(config-router)#network 172.16.10.1 0.0.0.0 area 0
2621A(config-router)#^Z
2621A#
```

2. 2501A router'undaki network'leri yapılandırın. Onları, area 0'a atayın:

```
2501A#conf t
Enter configuration commands, one per line.
  End with CNTL/Z.
2501A(config)#router ospf 101
2501A(config-router)#network 172.16.10.2 0.0.0.0 area 0
2501A(config-router)#network 172.16.20.1 0.0.0.0
```

```
  area 0
2501A(config-router)#^Z
2501A#
```

3. 2501B router'undaki network'leri yapılandırın. Onları, area 0'a atayın:

```
2501B#conf t
Enter configuration commands, one per line.
  End with CNTL/Z.
2501B(config)#router ospf 102
2501B(config-router)#network 172.16.20.2 0.0.0.0 area 0
2501B(config-router)#network 172.16.30.1 0.0.0.0 area 0
2501B(config-router)#network 172.16.40.1 0.0.0.0 area 0
2501B(config-router)#^Z
2501B#
```

4. 2501C router'undaki network'leri yapılandırın. Onları, area 0'a atayın:

```
2501C#conf t
Enter configuration commands, one per line.
  End with CNTL/Z.
2501C(config)#router ospf 103
2501C(config-router)#network 172.16.40.2 0.0.0.0 area 0
2501C(config-router)#network 172.16.50.1 0.0.0.0 area 0
2501C(config-router)#^Z
2501C#
```

## Pratik Lab 7.4: OSPF Operasyonlarını Doğrulamak

1. 2621 router'undan `show ip ospf neighbors` komutunu çalıştırın ve sonuca göz atın:

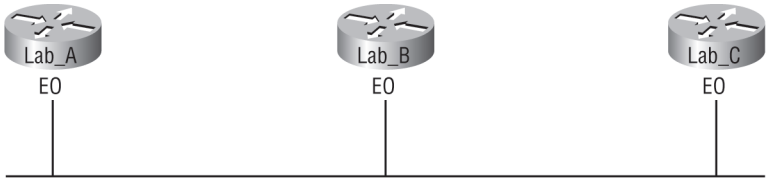
```
2621A#sho ip ospf neig
```

2. Tüm router'ların tüm route'ları öğrendiğini doğrulamak için `show ip route` komutunu kullanın:

```
2621A#sho ip route
```

## Pratik Lab 7.5: OSPF DR ve DBR Seçimleri

Bu lab'da seçim prosesini zorlayarak ve doğrulayarak, test network'ünüzdeki DR ve BDR seçimlerini izleyeceksiniz. Şekil 7.12'yi kullanarak, network'ünüzü oluşturmaya başlayacaksınız. Fazla router'a sahip olmak daha iyi olacaktır, fakat bu lab'ı tamamlamak için bir LAN segmenti yardımıyla en az üç router'a ihtiyacınız vardır.




***Şekil 7.12:*** *OSPF pratik lab network diyagramı.*

*Bu lab'ta, 2500 serisi router'lar kullanıyorum, fakat herhangi bir LAN interface tipine sahip, herhangi bir LAN tipi kullanabilirsiniz. Ya da, gerçek router'lar yerine Sybex veya RouterSim yazılım ürünlerini kullanabilirsiniz.*

NOT

1. İlk olarak, Şekil 7.12'de gösterilen network'ü birbirine bağlayın. Network için bir IP düzenlemesi oluşturun. Örneğin, 10.1.1.1/24, 10.1.1.2/24 ve 10.1.1.3/24 iyi çalışacaktır.
2. Şimdi OSPF'i yapılandırın ve tüm router'ları area 0'a yerleştirin. Bildiğiniz gibi, seri bağlantılarda seçim yapılmadığından, bu lab'ta sadece Ethernet LAN interface'lerinin yapılandırılması gerekmektedir.
3. Sonra, Area ID'si, DR, BDR bilgisinin ve LAN network'üne bağlı interface'in Hellove Dead timer'larının doğruluğunu kontrol etmek için her router'da show ip ospf interface e0 yazın.
4. `show ip ospf interface e0` çıktısına bakarak, hangi router'ın DR, hangisinin BDR olduğunu belirleyin.
5. Şimdi, router'ınızın network tipini doğrulayın. Bağlantı, bir Ethernet LAN'ında `olduğundan`, network tipi BROADCAST'tir. Şayet bir seri bağlantı görürseniz, point-to-point network istiyorsunuzdur.
6. Burada, router priority'sini ayarlamalısınız. Tüm router'ların priority'si varsayılan olarak, 1 dir. Şayet priority'i 0 (sıfır) olarak değiştirirseniz, router LAN için seçim prosesine asla katılmayacaktır. (seri point-to-point linklerde seçim olmadığını hatırlayın).
7. Şimdi, hangi router'ın yeni DR olacağına karar vermeniz gerekir.
8. Sonra, DR ve BDR seçimini görmenizi sağlayan debugging prosesini etkinleştirin.Tüm router'larda `debug ip ospf adjacency` yazın.

**NOT** *Diğer router'lara telnet yaparak, birden fazla konsol bağlantısı açmayı deneyin. Telnet oturumunda terminal monitor komutunu kullanmayı unutmayın, yoksa hiçbir debugging çıktısı göremezsiniz.*

9. Burada, `ip ospf priority` 3 yazarak yeni DR Ethernet 0 interface'inin priority'sini 3'e ayarlayın.
10. Sonra, DR router'ının Ethernet interface'ini kapatın ve `no shutdown` komutu ile onu tekrar etkinleştirin. Şayet bu router'a telnet yaparsanız, bu noktada oturumunuzu kaybedeceksiniz.
11. Burası, seçimin olacağı yerdir ve DR seçtiğiniz router, gerçekten DR olmalıdır.
12. Son olarak, her router'daki DR ve BDR bilgisini doğrulamak için show ip ospf interface e0 yazın.

**NOT** *Router interface'inin priority'sini, 255'e ayarlayabilirsiniz. Yani, daima area'nın DR'ı olacaktır. Sonra, test ağınızdaki bir router'ı, daha yüksek bir priority ile ayarlayabilir ve bir loopback (mantıksal) interface kullansanız dahi, bir router'daki yüksek RID'e üstünlük sağlayacağını görün.*

# Gözden Geçirme Soruları

1. Firmanız, 10 AS kullanarak IGRP çalıştırıyor.Network'ünüzdeEIGRP yapılandırmak istiyorsunuz, fakat EIGRP'ye yavaşça geçmek istiyorsunuz ve redistribution kullanmak istemiyorsunuz. Hangi komut, redistribution kullanmadan EIGRP'ye geçmenizi sağlar?

   A. `router eigrp 11`

   B. `router eigrp 10`

   C. `router eigrp 10 redistribute igrp`

   D. `router igrp combine eigrp 10`

*Aşağıdaki sorular, bu modülün materyallerini anladığınızı test etmek için tasarlanmıştır. İlave sorulara nasıl sahip olacağınızla ilgili daha fazla bilgi için lütfen bu kitabın Giriş bölümüne bakın.*

NOT

2. Hangi EIGRP bilgisi, RAM'de tutulmaktadır ve Hello ile güncelleme paketlerinin kullanımıyla devamlılığı sağlanmaktadır?(İki şık seçin)

   A. Neighbor table

   B. STP table

   C. Topology table

   D. DUAL table

3. Aşağıdakilerden hangisi, bir router'da OSPF çalıştırmak için kullanılan process ID'yi tanımlamak için kullanılır?(İki şık seçin)

   A. Lokal olarak önemlidir.

   B. Global olarak önemlidir.

   C. OSPF veritabanının eşsiz örneğini belirlemek için gereklidir.

   D. Sadece, router'da, çoklu OSPF prosesi çalıştığında gereken, isteğe bağlı bir parametredir.

   E. Şayet routing bilgilerini değiştireceklerse, aynı OSPF area'sındaki router'lar, aynı Process ID'ye sahip olmalıdırlar.

4. EIGRP successor route'u nerede tutulur?

   A. Sadece routing table'da

   B. Sadece neighbor table'da

   C. Sadece topology table'da

   D. Routing table ve neighbor table'da

   E. Routing table ve topology table'da

   F. Topology table ve neighbor table'da

5. Hangi komut, bir router'a bilinen tüm EIGRP feasible successor route'larını gösterecektir?

   A. `show ip routes *`

   B. `show ip eigrp summary`

   C. `show ip eigrp topology`

   D. `show ip eigrp adjacencies`

   E. `show ip eigrp neighbors detail`

6. Router'ına aşağıdakileri yazdığını söyleyen bir network yöneticisinden telefon alıyorsunuz:

```
Router(config)#router ospf 1
Router(config-router)#network 10.0.0.0 255.0.0.0 area 0
```

Size, routing tablosunda hala bir şey göremediğini söylüyor. Yöneticinin yaptığı hata nedir?

A. Wildcard mask yanlıştır.

B. OSPF area'sı yanlıştır.

C. OSPF Process ID'si yanlıştır.

D. AS konfigürasyonu yanlıştır.

7. Aşağıdaki protokollerden hangisi, VLSM, summarization ve discontiguous network kurulumunu destekler? (Üç şık seçin)

A. RIPv1

B. IGRP

C. EIGRP

D. OSPF

E. BGP

F. RIPv2

8. Aşağıdakilerden hangisi OSPF area'ları hakkında doğrudur? (Üç şık seçin.)

A. Her area'da ayrı loopback interface'lerine sahip olmalısınız.

B. Bir area'ya atayabileceğiniz numaralar 65,535'e kadardır.

C. Backbone area, ayrıca area 0 olarak belirtilmektedir.

D. Şayet, tasarımınız hiyerarşik ise, çoklu area'lara ihtiyacınız yoktur.

E. Tüm area'lar, area 0'a bağlanmalıdır.

F. Şayet sadece bir area'nız varsa, onun area 1 olarak tanımlanması gerekir.

9. Aşağıdaki network tiplerinden hangisi, bir designated router ve bir backup designated router'a sahiptir? (İki şık seçin)

A. Broadcast

B. Point-to-point

C. NBMA

D. NBMA point-to-point

E. NBMA point-to-multipoint

10. Bir network yöneticisinin, bir router'ı, classless routing'e izin veren distance-vector protokol ile yapılandırması gerekmektedir. Aşağıdakilerden hangisi bu gereklilikleri yerine getirmektedir?

A. IGRP

B. OSPF

C. RIPv1

D. EIGRP

E. IS-IS

11. Router'ın bir adjacency kuracağı cihazların IP adresine ihtiyacınız var. Ayrıca, adjacent router'lar için, retransmit interval ve queue count'larının kontrol edilmesi gerekmektedir. Hangi komut, gereken bilgileri görüntülemektedir?

    A. show ip eigrp adjacency

    B. show ip eigrp topology

    C. show ip eigrp interfaces

    D. show ip eigrp neighbors

12. Bazı nedenlerden, iki router arasındaki Ethernet linkinde bir adjacency kuramazsınız. Aşağıdaki çıktıya bakılarak, problemin sebebi ne olduğu söylenebilir?

```
RouterA#
EthernetO/O is up, line protocol is up
  Internet Address 172.16.1.2/16, Area O
  Process ID 2, Router ID 172.126.1.1, Network Type BROADCAST,
Cost: 1O
  Transmit Delay is 1 sec, State DR, Priority 1
  Designated Router (ID) 172.16.1.2, interface address 172.16.1.1
  No backup designated router on this network
  Timer intervals configured, Hello 5, Dead 2O, Wait 2O,
Retransmit 5

RouterB#
EthernetO/O is up, line protocol is up
  Internet Address 172.16.1.1/16, Area O
  Process ID 2, Router ID 172.126.1.1, Network Type BROADCAST,
Cost: 1O
  Transmit Delay is 1 sec, State DR, Priority 1
  Designated Router (ID) 172.16.1.1, interface address 172.16.1.2
  No backup designated router on this network
  Timer intervals configured, Hello 1O, Dead 4O, Wait 4O,
Retransmit 5
```

    A. OSPF area, tam anlamıyla yapılandırılmamıştır.

    B. RouterA'daki priority, daha yüksek değere ayarlanmalıdır.

    C. RouterA'nın cost değeri, daha yüksek olmalıdır.

    D. Hello ve Dead timer'ları, tam olarak yapılandırılmamıştır.

    E. Bir backup designated router'un network'e eklenmesi gerekmektedir.

    F. OSPF Process ID numaraları eşleşmelidir.

13. EIGRP successor route'ları hakkında hangisi doğrudur? (İki şık seçin)

    A. Bir successor route, EIGRP tarafından, trafiği bir hedefe göndermek için kullanılmaktadır.

    B. Successor route'lar, öncelikli route'un arızalanması durumunda kullanılmak için topoloji tablosuna kaydedilmektedir.

    C. Successor route'lar, routing tablosunda "aktif" olarak işaretlenirler.

    D. Bir successor route, bir feasible successor route ile yedeklenebilir.

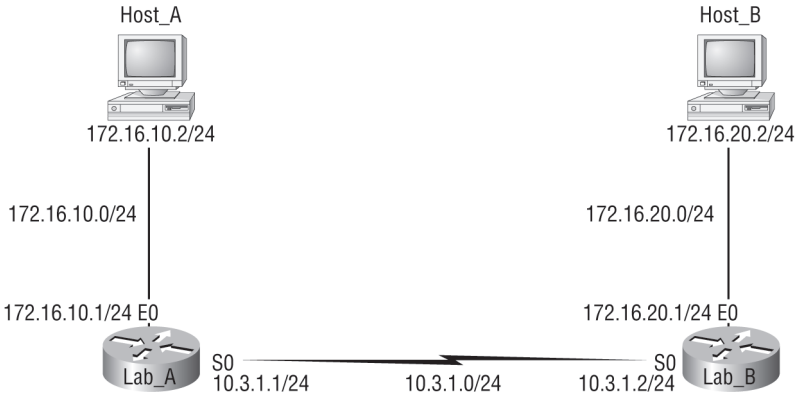
    E. Successor route'lar, discovery prosesinin ardından, neighbor tablosunda tutulmaktadır.

14. Hangi OSPF network tipi, bir backup designated router seçecektir? (İki şık seçin)
    A. Broadcast multi-access
    B. Non-broadcast multi-access
    C. Point-to-point
    D. Broadcast multipoint
15. Aşağıdaki komutlardan hangisi, 10.2.3.0/24 network'ünü, area 0'a yerleştirecektir? (İki şık seçin.)

```
A. router eigrp 10
B. router ospf 10
C. router rip
D. network 10.0.0.0
E. network 10.2.3.0 255.255.255.0 area 0
F. network 10.2.3.0 0.0.0.255 area0
G. network 10.2.3.0 0.0.0.255 area 0
```

16. Hangi network tipiyle, OSPF router adjacencey kuracaktır, fakat DR/BDR seçim prosesi çalıştırmayacaktır?
    A. Point-to-point
    B. Backbone area 0
    C. Broadcast multi-access
    D. Non-broadcast multi-access
17. OSPF için hiyerarşik bir tasarım, oluşturmanın üç nedeni nedir? (Üç şık seçin.)
    A. Routing yükünü azaltmak.
    B. Convergence'ı hızlandırmak.
    C. Network kararsızlığını, tek network area'larına sınırlandırmak.
    D. OSPF konfigürasyonunu basitleştirmek.
18. OSPF'in administrative distance'ı nedir?
    A. 90
    B. 100
    C. 110
    D. 120

19. Aşağıdaki grafikte görünen bir ağ topluluğuna sahipsiniz. İki network, routing tablosu route kayıtlarını paylaşmamaktadır. Problemi çözmek için hangi komut gerekmektedir?




A. `version 2`

B. `no auto-summary`

C. `redistribute eigrp 10`

D. `default-information originate`

20. Tek bir area'daki router'lar, aynı priority değerine sahiplerse loopback interface'in yokluğunda bir router, OSPF Router ID için hangi değeri kullanır?

A. Herhangi bir fiziksel interface'in en düşük IP adresini.

B. Herhangi bir fiziksel interface'in en yüksek IP adresini.

C. Herhangi bir mantıksal interface'in en düşük IP adresini.

D. Herhangi bir mantıksal interface'in en yüksek IP adresini.

# Gözden Geçirme Sorularının Cevapları

1. B Şayet aynı autonomous system (AS) ile EIGRP'yi etkinleştirdiyseniz, EIGRP, otomatik olarak IGRP'yi EIGRP'ye redistribute edecektir. IGRP route'ları, 170 EIGRP AD ile external (EX) route olarak göreceksiniz. Bu, ilave konfigürasyon yapmadan, EIGRP'ye yavaşça geçmenizi sağlayan güzel bir özelliktir.

2. A, C EIGRP, RAM'de üç tablo tutar; neighbor, topoloji ve routing. Neighbor ve topoloji tablosu, Hello paketlerinin kullanılması ile oluşturulur ve devamlılığı sağlanır.

3. A, C Bir router'daki OSPF Process ID'si, sadece lokal olarak önemlidir ve aynı numarayı her router'da kullanabilirsiniz yada her router farklı bir numaraya sahip olabilir (bunun önemi yoktur). Kullanabileceğiniz numaralar, 1'den 65,535'e kadardır. Bunu, 0'dan 4,2 milyara kadar olan area numarası ile karıştırmayın.

4. E Successor route'lar, uzak bir network'e en iyi yol olduğundan, routing tablosunda olacaklardır. Bununla beraber, topoloji tablosu, her network için bir linke sahiptir, bu yüzden en iyi cevap, topoloji tablosu ve routing tablosudur. Uzak bir network için herhangi bir ikincil route, feasible route olarak kabul edilir. Bu route'lar sadece topoloji tablosunda bulunurlar ve birincil route'un arızalanması durumunda, yedek route olarak kullanılırlar.

5. C Uzak bir network için, herhangi bir ikincil route, feasible route olarak kabul edilir. Bu route'lar sadece topoloji tablosunda bulunurlar ve birincil route'un arızalanması durumunda, yedek route olarak kullanılırlar. Topoloji tablosunu, `show ip eigrp topology` komutu ile görebilirsiniz.

6. A Yönetici, yanlış wildcard mask yazmıştır. Wildcard, 0.0.0.255 olmalıydı.

7. C,D,F.RIPv1 ve IGRP, gerçek distance-vector routing protokolüdür ve routing tabloları oluşturmak, devam ettirmek ve bol bol bant genişliği kullanmak dışında çok fazla şey yapmazlar. RIPv2, EIGRP ve OSPF, routing tablolarını oluşturur ve sürdürürler. Ayrıca VLSM'e, summarization'a ve discontigious network kullanımına izin veren classless routing'de sağlarlar.

8. C, D, E Loopback interface'leri, bir router'da oluşturulur ve bir loopback(mantıksal) interface'deki en yüksek IP adresi router'ın RID'i olur. Area'larla ilgisi yoktur ve isteğe bağlıdır, bu nedenle A şıkkı yanlıştır. Bir area oluşturabileceğiniz sayı, 0 ile 4,294,967,295 arasındadır. B şıkkı yanlıştır. Backbone area, area 0 olarak belirtilir, bu nedenle C şıkkı doğrudur. Tüm arealar, area 0'a bağlanmalıdır, E şıkkı doğrudur. Sadece bir area'nız varsa bunun area 0 olması gerekir, yani F şıkkı yanlıştır. Geriye doğru olması gereken D şıkkı kalır. Çok mantıklı değildir ama en iyi cevaptır.

9. A, C Hiçbir point-to-point linkte, DR atanmaz. Hub/spoke topoloji olduğundan, NBMA point-to-multipoint'e DR/BDR atanmaz. DR ve BDR, broadcast ve non-broadcast multi-access network'lerde seçilir. Frame Relay, varsayılan olarak, non-broadcast multi-access network'tür.

10. D Bu soruda, EIGRP'yi eski distance-vector olarak belirtiyoruz. EIGRP, gelişmiş distance-vector routing protokolüdür. Hem distance-vector hem de link-state protokollerinin özelliklerini kullandığından, bazen karma (hibrit) bir routing protokolü olarak belirtilmektedir.

11. D `show ip eigrp neighbors` komutu, hem IP adreslerini hem de bir adjacency kuran komşular için tekrar aktarma süresi ve kuyruk sayılarını kontrol etmenizi sağlar.

12. D Hello ve Dead timer'ları, aynı linkteki iki router'da da aynı olmalıdır. Yoksa bir adjacency kuramayacaklardır. OSPF için varsayılan timer'lar, Hello timer için 10, Dead timer için 40 saniyedir.

13. A, D Successor route, uzak bir ağa en iyi route olarak, topoloji tablosundan alınan route'dur. Bu nedenle uzak ağa IP trafiği göndermek için routing tablosundan öğrenilip kullanılan route'lardır. Topoloji tablosu, successor route kadar iyi olmayan tüm routeları içerir ve bun-

lar, feasible successor ya da yedek route olarak kabul edilirler. Tüm route'ların (successor route'ların bile) topoloji tablosunda olduğunu hatırlayın.

14. A, B DR ve BDR, broadcast ve non-broadcast multi-access network'lerde seçilir. Frame Relay varsayılan olarak bir non-broadcast multi-access(NBMA) network'tür. Herhangi bir point-to-point link tipinde DR atanmaz. Hub/spoke topoloji olduğundan, NBMA point-to-multi point'e DR/BDR atanmaz.

15. B, G OSPF'i etkinleştirmek için ilk olarak bir Process ID kullanarak, OSPF'i başlatmalısınız. Numaranın ne olduğu önemli değildir. Sadece 1'den 65,535'e kadar seçin ve devam edin. OSPF prosesini başlattıktan sonra, wildcard ve area komutunu kullanarak OSPF yoluyla yayınlamayı istediğiniz network'leri yapılandırmalısınız. F şıkkı yanlıştır. Çünkü area parametresi ile area numarası arasında boşluk olmalıdır.

16. A point-to-point link türlerine DR atanmamaktadır. Hub/spoke topoloji olduğundan, NBMA point-to-multipoint'lere DR/BDR atanmamaktadır. DR ve BDR, broadcast ve non-broadcast multi-access network'lerde seçilir. Frame Relay varsayılan olarak, bir non-broadcast multi-access(NBMA) network'tür.

17. A, B, C OSPF, hiyerarşik bir tasarımda oluşturulur, RIP gibi düz tasarımda olmaz. Bu, routing yükünü azaltır, convergence'i hızlandırır ve network tutarsızlığını, tek area'lı network'lerle sınırlandırır.

18. C Administrative distance (AD), routing protokolünde çok önemli bir parametredir. AD'si ne kadar düşükse, o kadar güvenli route olacaktır. Şayet IGRP ve OSPF çalıştırıyorsanız, varsayılan olarak, 100 AD'sine sahip IGRP, routing tablosuna yerleştirilecektir. RIPv1 ve RIPv2'nin AD'si 120, EIGRP'ninki en düşük olan 90'dır.

19. B Diyagramdaki network, discontigious network olarak kabul edilir. Çünkü subnetlenmiş ve diğer classful adresle bölünmüş bir classful adrese sahipsiniz. Sadece RIPv2, OSPF ve EIGRP discontigious network'lerle çalışabilir, fakat RIPv2 ve EIGRP, varsayılan olarak çalışmazlar. Routing protokol konfigürasyonu altında, no auto-summary komutunu kullanmalısınız.

20. B OSPF prosesi başladığı an, aktif interface'lerdeki en yüksek IP adresi, router'ın Router ID'si (RID) olacaktır. Şayet yapılandırılmış bir mantıksal (loopback) interface'iniz varsa, interface IP adresini geçersiz kılacaktır ve otomatik olarak, router'ın RID'i olacaktır.

# Yazılı Lab 7 Cevapları

1. EIGRP tarafından desteklenen dört routeable protokol şunlardır: IP, IPv6, IPX ve AppleTalk.
2. Birden fazla EIGRP oturumu ya da prosesi çalıştığında, redistribution gereklidir ve onlar, farklı ASN'ler ile tespit edilmektedir. Redistribution, EIGRP oturumları arasındaki topoloji bilgisini paylaşır.
3. `router eigrp 300`
4. `network 172.10.0.0`
5. Passive interface
6. `router ospf 101`
7. `show ip ospf`
8. `show ip ospf interface`
9. `show ip ospf neighbor`
10. `show ip ospf database`

# 8 Katman2 Switching ve Spanning Tree Protokolü (STP)

# 8 Katman2 Switching ve Spanning Tree Protokolü (STP)

- Katman2 Switching Öncesi
- Switching Servisleri
- Spanning Tree Protokolü (STP)
- Catalyst Switch'leri Yapılandırmak
- Cisco Network Assistant
- Özet
- Sınav Gereklilikleri
- Yazılı Lab 8
- Gözden Geçirme Soruları
- Gözden Geçirme Sorularının Cevapları
- Yazılı Lab 8.1'in Cevapları

# Katman2 Switching ve Spanning Tree Protokolü (STP)

Cisco çalışanları switching'i tartıştığında aksini söylemedikçe, katman2 switching'ten bahsediyorlardır. Katman2 switching, bir ağı segment'lemek için bir LAN'daki cihazların donanım adresi kullanmasının bir prosesidir. Temel kavramları bilmek zorunda olduğunuzdan, katman2 switching'in detaylarına ve çalışmasına odaklanacağım.

Switching'in geniş domain'leri daha küçük domain'lere ayırdığını ve bir collision domain'inin, aynı bant genişliğinin iki ya da daha fazla cihazın paylaştığı bir network segment'i olduğunu biliyorsunuz. Bir hub ağı, bu teknoloji türüne tipik bir örnektir. Fakat bir switch'teki her port kendi collision domain'inde olduğundan, hub'ların yerine switch'ler yerleştirerek daha iyi bir Ethernet LAN ağı oluşturabilirsiniz. Switch'ler gerçekte, ağların tasarlanma ve kurulum tarzını değiştirir. Sadece switch'lerden oluşan bir network kurulduysa, o tamamen temiz, uygun maliyetli ve kendini çabuk toparlayan bir ağ topluluğu olacaktır. Bu bölümde, switching teknolojisinin ortaya çıkmasından önce ve sonra ağların nasıl tasarlandığını inceleyip, mukayesesini yapacağım.

Routing protokolleri (Bölüm 6, "IP Routing"de gördüğümüz RIP gibi), Network katmanında olan network döngülerini durduran proseslere sahiptir. Bununla birlikte, switch'leriniz arasında fiziksel yedek linkler varsa routing protokolleri, Data Link katmanında olan döngüleri durdurmak gibi bir şey yapmayacaktır. Katman2 switch ağındaki kısır döngüleri durdurmak için Spanning Tree Protokolünü geliştirildi. Bu çok önemli protokolün temeli ve bir switch ağında nasıl çalıştığı, bu bölüm boyunca işleyeceğimiz önemli konulardandır.

Bir switch network konfigürasyonumuza başlamak için üç switch kullanacağım ve onların konfigürasyonuna bölüm 9 "Virtual LAN'larda (VLAN) devam edeceğim.

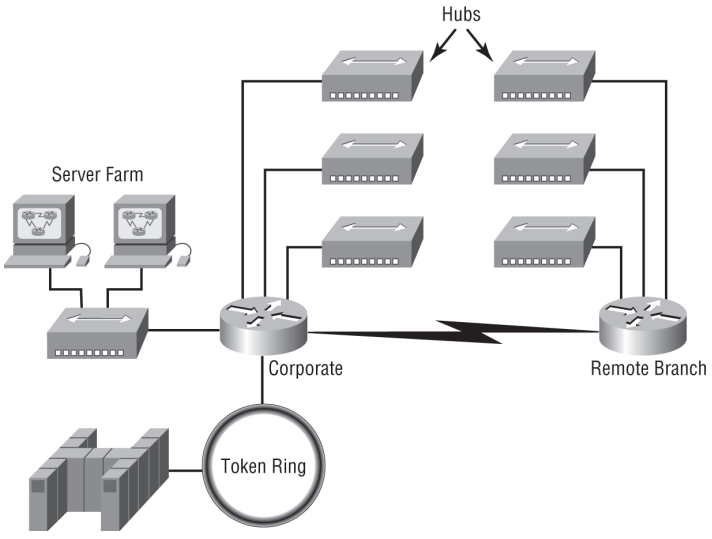
*Bu bölümle ilgili son güncellemeler için www.lammle.com ve/veya www.sybex.com 'e bakınız.*

NOT

## Katman2 Switching Öncesi

Gelin bir miktar geriye gidelim, switch'lerden önce ağların durumuna ve switch'lerin, firma LAN'larını segment'lemeye nasıl yardımcı olduğuna bir bakalım. LAN switching'ten önce, tipik bir network tasarımı, Şekil 8.1'deki gibiydi.

Şekil 8.1'deki tasarım, collapsed backbone (omurga) olarak tanımlanmaktaydı. Çünkü tüm host makinelerinin, herhangi bir network servisine (hem LAN'a hem de anabilgisayara) ulaşmak için, Corporate omurgasına gitmeleri gerekmektedir.




***Şekil 8.1:*** *Switching öncesi.*

Daha da geriye gidersek, Şekil 8.1'de gösterilen router ve hub'lar gibi fiziksel segment'lere bölen cihazlara sahip ağlardan önce, mainframe ağı vardı. Bu network, anabilgisayar (IBM, Honeywell, Sperry, DEC, vs), controller'lar ve controller'lara bağlanan aptal terminallerden oluşmaktaydı. Uzak lokasyonlar, anabilgisayara bridge'lerle bağlanırdı.

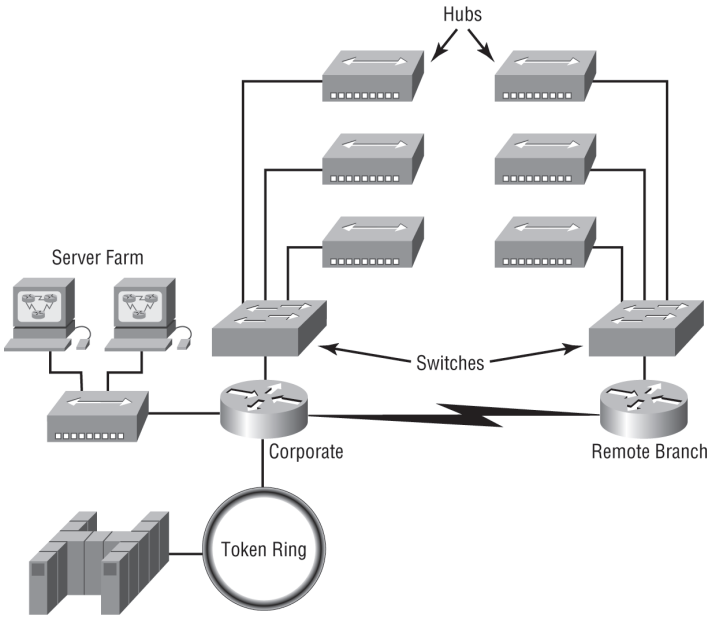
Daha sonra PC'nin yıldızı parlamaya başladı ve anabilgisayar, sunucu makinelerinin kurulu olduğu Ethernet ya da Token Ring LAN'ına bağlandı. Bu sunucular genellikle, NT öncesi olduğundan, OS/2 veya LAN Manager'dı. Bir binanın her katı, firma omurgasına ve sonra bir router'a, koaksiyel ya da sarmal-çift kablolama ile bağlıydı. PC'ler, kendilerinin ana bilgisayara bağlanmasını sağlayan ve ana bilgisayar ile LAN'daki servislere erişim kabiliyeti sağlayan bir emülasyon yazılımı çalıştırırdı. Sonunda PC'ler, uygulama geliştiricilere port uygulamaları için daha önce yapabildiklerinden daha etkili şekilde, izin verecek kadar güçlü oldular. Network kurulum maliyetini önemli derecede düşürecek ve sektörün daha hızlı büyümesini sağlayacak kadar ilerleme sağladılar.

Novell, 1980'lerin sonunda ve 1990'ların başında daha popüler olduğunda, OS/2 ve LAN Manager sunucuların yerini ağırlıklı olarak NetWare sunucular aldı. Nowell3.x sunucuların istemci/sunucu yazılımıyla haberleşmek için kullanılmasından dolayı bu, Ethernet ağlarını daha da popüler yaptı.

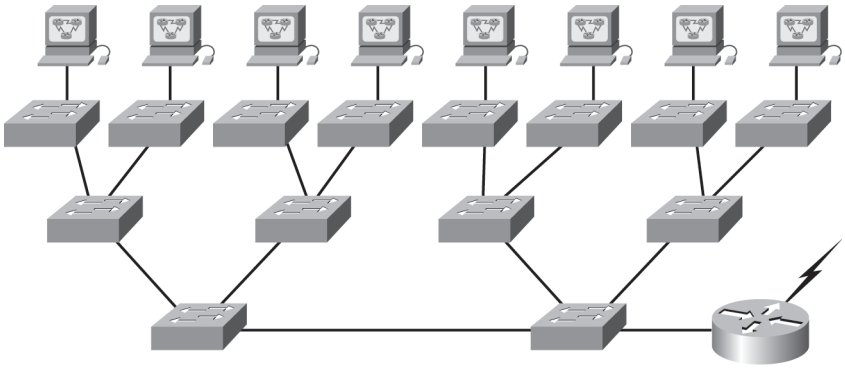
Şekil 8.1'deki networ'ke nasıl gelindiğinin hikayesi böyledir. Yalnız, bir problem vardı; şirket omurgası sürekli büyüyordu ve bu büyümeyle servisler yavaşlıyordu. Bunun en büyük nedeni, büyümedeki bu büyük patlamayla, LAN servislerinin daha hızlı olması gerekmesi ve network'ün tamamıyla doymuş olmasıydı. Herkes, şirket omurgasına ve network servislerine daha kolay bağlanabilecekleri bu akıllı yeni PC'ler için Mac'lerini ve mainframe servisi için kullanılan aptal terminal'lerini satıyordu.

Bütün bunlar, internetin oldukça popüler olmasından önce oluyordu. Bu nedenle şirketteki herkesin, şirketin network servislerine erişmesi gerekiyordu. Niçin? Çünkü, internetsiz tüm network servisleri dahiliydi, yani şirket ağıyla sınırlıydı. Bu, eski ve ağır router'larla bağlı şirket ağının çok çalışarak segment'lenmesi ihtiyacını doğurdu. İlk önce, Cisco, daha hızlı router'lar geliştirdi (bundan hiç kuşkumuz yok), fakat özellikle de Ethernet LAN'larında, daha çok segment'lemeye ihtiyaç vardı. FastEthernet'in icadı çok güzel ve faydalıydı, fakat network segment'leme ihtiyaçları için tamamıyla yeterli değildi.

Fakat bridge denilen cihazlar yeterliydi ve ilk olarak collision domain'lerini ayırmak için kullanıldılar. Bridge'ler, katman2 switch'ler yardıma yetiştiğinde, port sayıları ve sağlayabildikleri network servisleri oldukça sınırlıydı. Switch'ler, bir bridge gibi collision domain'lerini her port için ayırarak, günü kurtardı. Switch'ler, yüzlerce port sağlayabiliyordu. Bu eski switch network'leri, Şekil 8.2'de gösterilene benziyordu.




***Şekil 8.2:*** *İlk switch LAN'ı.*



***Şekil 8.3:*** *Tipik switch'lerden oluşan bir network tasarımı.*

Her hub, bir switch port'una bağlıdır. Bu, network'ü çok geliştiren bir yenilikti. Artık, her binanın aynı collision domain'e sıkışması yerine, her hub kendi collision domain'ine sahipti. Fakat burada bir tuzak vardı, switch port'ları oldukça yeniydi, bu nedenle inanılmaz pahalıydılar. Bundan dolayı bunları binanın her katına kolayca eklemek mümkün değildi. Bütün bunlar için her kime teşekkür etmeyi tercih edersiniz, ona teşekkürler, fiyatlar çarpıcı şekilde düştü böylece, bir switch port'una bağlı kullanıcılara sahip olmak hem güzel hem de uygun hale geldi.

Şayet bir network tasarımı oluşturacak ve onu kuracaksanız, switching servislerini içermesi bir zorunluluktur. Tipik çağdaş bir network tasarımı, Şekil 8.3'teki gibi komple switch'lerden meydana gelen network tasarımı ve kurulumundan oluşmalıdır.

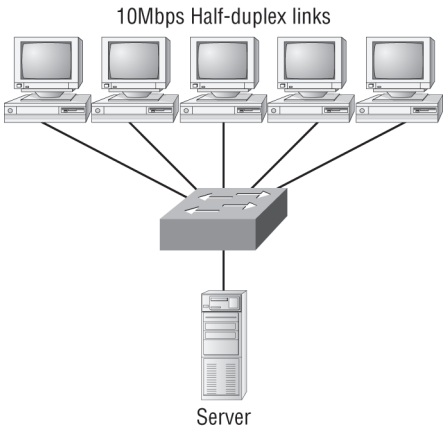
"Fakat hala, orada bir router görüyorum" diyebilirsiniz! Evet, bu bir serap değil, orada bir router vardır. Fakat onun görevi değişti. Fiziksel segment'leme yapmak yerine şimdi, mantıksal bir segment'leme oluşturmak için kullanılmaktadır. Bu mantıksal segment'ler, VLAN olarak bilinir ve söz veriyorum, bu bölüm ve bölüm 9 boyunca onlardan bahsedeceğim.

# Switching Servisleri

Bir filtre tablosu oluşturmak ve yönetmek için yazılım kullanan bridge'lerin tersine switch'ler, kendi filtreleme tablolarını oluşturmak ve korumak için application-specific integrated circuits (ASIC) kullanır. Kullanılma sebeplerinin aynı (collision domain'lerini ayırmak) olmasından dolayı, çok port'lu bir bridge olarak, katman2 switch şeklinde düşünülmesi doğaldır.

Katman 2 switch'ler ve bridge'ler, router'lardan hızlıdır. Çünkü onlar, Network katmanı başlık bilgilerine bakmaya zaman harcamazlar. Onun yerine, frame'i ,iletmeye, yaymaya ya da iptal etmeye karar vermeden önce, farme'in donanım adreslerine bakarlar.

Switch'ler özel, adanmış collision domain'ler oluşturur ve hub'ların tersine her port için bağımsız bant genişliği sağlar. Şekil 8.4, tamamı sunucuya 10 Mbps half-duplex çalışan, switch'e bağlı beş host makinasını göstermektedir.




***Şekil 8.4:*** *Switch'ler, özel domain'ler oluşturur.*

10Mbps Half-Dublex linkler

Katman2 switching şunları sağlamaktadır:

- Donanım-tabanlı bridging (ASIC)
- Wire speed
- Düşük latency
- Düşük cost

Katman2 switching'i bu kadar etkili yapan, veri paketlerinde değişiklik yapmamasıdır. Cihaz sadece, paketi enkapsüle eden frame'i okur. Bu, switching prosesini, routing prosesinden daha hızlı ve daha az hata eğilimli yapar.

Şayet katman2 switching'i, hem workgroup bağlantısı hem de network segment'lemesi (collision domain'lerini ayırmak) için kullanıyorsanız, geleneksel routing kullanılan network'lerle olandan daha çok network segment'ine sahip bir network tasarımı oluşturabilirsiniz.

Artı, katman2 switching, her kullanıcı için bant genişliğini artırır. Çünkü switch'e her bağlantı (interface), kendi collision domain'ine sahiptir. Bu özellik sizin, her interface'e birçok cihaz bağlamanıza imkan verir.

Aşağıdaki bölümlerde, katman2 switching terminolojisine derinlemesine gireceğim.

## Katman2 Switching'in Limitleri

Genelde katman2 switching'i bridge network'lerle aynı katagoriye koyduğumuzdan, bridge network'leriyle aynı güçlük ve sorunlara sahip olduğunu düşünürüz. Şayet ağınızı, doğru bir şekilde tasarlarsanız, özelliklerini ve sınırlarını aklınızda tutatarak, bridge'lerin iyi ve kullanışlı olduğunu unutmayın. Bridge'lerle iyi bir tasarım için göz önünde bulundurulması gereken iki önemli konu vardır:

- Collision domain'leri, doğru bir şekilde ayırmalıyız.
- İşlevsel bir bridge network'ü tasarlamanın doğru yolu, kullanıcıların zamanının %80'nini lokal segment'te geçirdiklerinden emin olmaktır.

Bridge network'leri collision domain'lerini ayırır fakat network'ün, hala büyük bir broadcast domain'i olduğunu hatırlayın. Ne katman2 switch'ler ne de bridge'ler, varsayılan olarak broadcast domain'leri ayırmaz. Broadcast domain'in büyük olması, sadece network boyutunuzu ve büyüme potansiyelinizi sınırlamaz, ayrıca onun baştan sona performansını da düşürür.

Spanning tree'nin yavaş covergence zamanı ile broadcast ve multicast'ler, ağınızın büyümesiyle sizi oldukça zor durumda bırakabilir. Bu, ağ topluluklarında, katman2 switch ve bridge'lerin yerine, router'ların (katman3 cihazların) kullanılmasının ana sebebidir.

## LAN Switching ile Bridging'in Karşılaştırılması

Katman2 switch'lerin daha fazla sayıda port'la, bridge'lerle neredeyse aynı oldukları doğrudur. Fakat unutmamanız gereken bazı önemli farklılıklar vardır:

- Switcler, donanım tabanlıyken, bridge'ler, yazılım tabanlıdır. Switch'ler, filtreleme kararlarında yardımcı olması için ASIC yongaları kullanırlar.
- Bir switch, çok port'lu bir bridge olarak sayılabilir.
- Switch'ler çok sayıda sahip olabilirken, her bridge için sadece bir spanning-tree instance olabilir. (Spanning tree'den yakında bahsedeceğim.)
- Switch'ler, birçok bridge'ten daha çok sayıda port'a sahiptir.
- Hem bridge'ler hem de switch'ler, katman2 broadcast'ini geçirirler.
- Bridge'ler ve switch'ler, alınan her frame'in kaynak adresini inceleyerek MAC adresini öğrenir.
- Hem bridge'ler hem de switch'ler, katman2 adresi bazında forwarding kararları alır.

## Katman2'deki Üç Switch Fonksiyonu

Katman2 switching'in belli başlı üç fonksiyonu vardır (bunları hatırlamanız gerekir !): Adres öğrenme, forward/filter kararları ve döngüden kaçınma.

**Adres öğrenme:** Katman2 switch ve bridge'ler, bir interface'den alınan her frame'in kaynak donanım adresini hatırlarlar ve bu bilgiyi, forward/filter tablosu denilen bir MAC veritabanına girer.

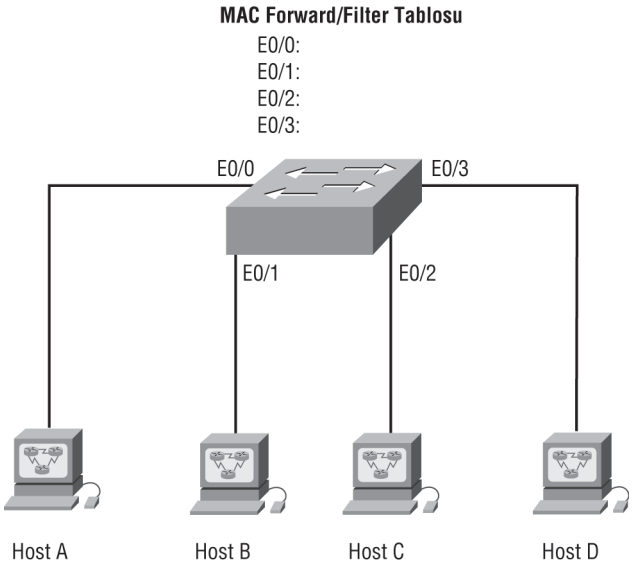
**Forward/filter kararları:** Bir frame, interface'den alındığında switch, hedef donanım adresine bakar ve MAC veritabanındaki çıkış interface'ini bulur. Frame, belirli hedef port'undan gönderilir.

**Döngüden kaçınma:** Şayet switch'ler arasındaki çoklu bağlantılar, yedeklilik amacıyla oluşturulduysa, network kısır döngüleri olabilir. Spanning Tree Protocol (STP) yedekliliğe hala izin verilirken, network döngülerini durdurmak için kullanılmaktadır.

Adres öğrenme, forward/filter kararları ve kısır döngüden kaçınmadan gelecek bölümlerde, detaylı olarak bahsedeceğim.

## Adres Öğrenme

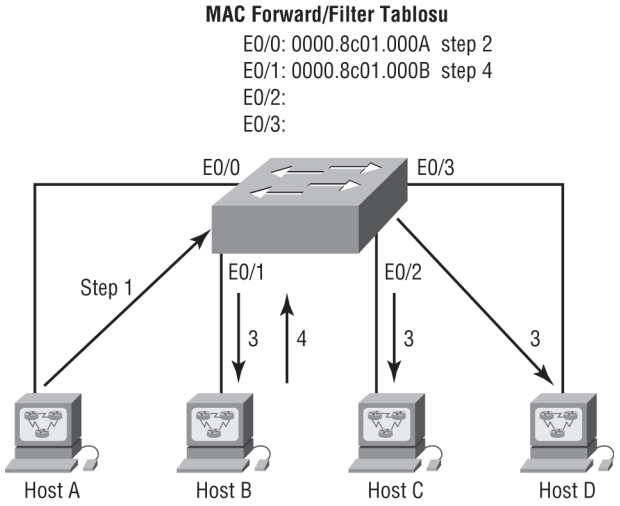
Switch ilk olarak açıldığında Şekil 8.5'de görüldüğü gibi, MAC forward/fitler tablosu boştur.




***Şekil 8.5:*** *Bir switch'teki boş MAC forward/fitler tablosu.*

Bir cihaz bir frame'i aktardığında ve bir interface bir frame aldığında, switch, frame'in kaynak adresini, gönderen cihazın hangi interface'de olduğunu hatırlatmasına izin vermesi için forward/fitler tablosu'na yerleştirir. Bundan sonra, switch'in, bu frame'i, kaynak port dışında diğer tüm port'lardan network'e göndermekten başka seçeneği yoktur. Çünkü hedef cihazın nerede olduğuyla ilgili hiçbir fikri yoktur.

Şayet cihaz, gönderilen bu frame'e yanıt verirse, tekrar bir frame gönderecektir ve sonra switch, frame'den kaynak adresini alacaktır. Bu MAC adresini, alınan frame'in interface'iyle bu adresi ilişkilendirerek, veri tabanına yerleştirecektir. Switch şimdi, filtreleme tablosunda, her iki MAC adresine de sahip olduğundan, iki cihaz point-to-point bağlantı kurabilir. Switch'in, ilk olarak yaptığı gibi, frame'i tüm port'lardan göndermesine gerek yoktur, frame'ler şimdi, sadece iki cihaz arasında gönderilecektir. Bu, switch'lerin, hub'lardan tamamıyla daha iyi olmalarını sağlar. Bir hub network'ünde, tüm frame'ler, tüm port'lardan gönderilirler. Şekil 8.6, bir MAC veritabanı oluşturmayı içeren prosesi göstermektedir.




***Şekil 8.6:*** *Switch'ler, host'ların lokasyonunu nasıl öğrenir.*

Şekilde bir switch'e bağlı dört host'u görebilirsiniz. Switch açıldığında, Şekil 8.5'de görüldüğü gibi, MAC forward/fitler tablosunda hiçbir şey yoktur. Fakat host'lar, iletişime başladığında, switch, her frame'in kaynak donanım adresini, frame adresinin ilgili interface adresiyle beraber tabloya yerleştirir.

Bir forward/fitler tablosunun nasıl yerleştirildiğiyle ilgili bir örnek vereyim:

1. HostA, HostB'ye bir frame gönderir. HostA'nın MAC adresi, 0000.8c01.000A ve HostB'nin MAC adresi, 0000.8c01.000B'dir.
2. Switch, E0/0 interface'inden bir frame alır ve kaynak adresini, MAC adresi tablosuna yerleştirir.
3. Hedef adresi MAC adresi veritabanında olmadığından frame, kaynak port dışında tüm port'lardan gönderilir.
4. HostB, HostA'ya cevap verir. Switch, bu frame'i E0/1 interface'inden alır ve kaynak donanım adresini MAC veritabanına yerleştirir.
5. HostA ve HostB, şimdi point-to-point bir bağlantı kurabilir ve sadece iki cihaz frame'leri alacaktır. HostC ve HostD, henüz switch'e bir frame göndermediğinden, frame'leri görmeyecektir.

Şayet HostA ve HostB, belirli bir süre içinde switch'le iletişim kurmazlarsa switch, onu mümkün olduğu kadar güncel tutmak için kayıtlarını silecektir.

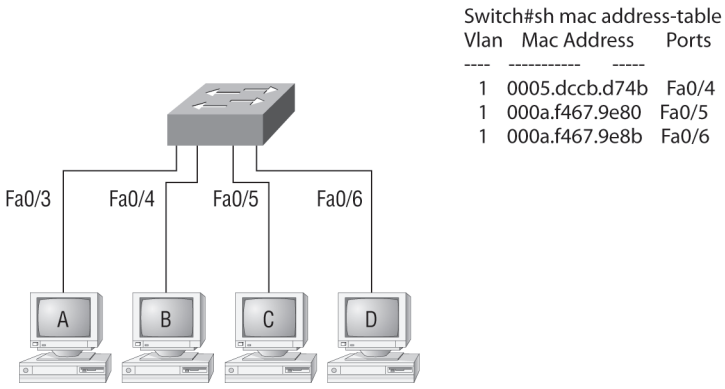
## Forward/Filter Kararları

Frame, bir switch interface'ine ulaştığında hedef donanım adresi, forward/fitler MAC veritabanıyla karşılaştırılır. Şayet hedef donanım adresi, biliniyor ve veritabanı listesindeyse frame sadece uygun çıkış interface'inden gönderilecektir. Switch, frame'i, hedef interface'i dışında herhangi bir interface'den göndermeyecektir. Bu, diğer network segment'lerindeki bant genişliğini korur ve frame filtreleme olarak bilinir.

Şayet, hedef donanım adresi, MAC veritabanında listelenmediyse frame, alındığı haricinde tüm aktif port'lardan dağıtılacaktır. Şayet bir cihaz, gönderilen frame'e cevap verirse, MAC veritabanı, cihazın lokasyonu (interface'i ) ile güncellenecektir.

Bir host ya da sunucu, LAN'a bir broadcast gönderirse, switch, frame'i, varsayılan olarak, kaynak port'u dışında, tüm aktif port'larından gönderecektir. Switch'in, daha küçük collision domain'leri oluşturduğunu hatırlayın, fakat varsayılan olarak, o hala geniş bir broadcast domain'idir.

Şekil 8.7'de, HostA, HostD'ye bir veri frame'i göndermektedir. Switch, HostA'dan bir frame aldığında, ne yapacaktır?




***Şekil 8.7:*** *İletme/filtreleme tablosu.*

HostA'nın MAC adresi, forward/fitler tablosunda olmadığından, switch, kaynak adresini ve port'u MAC adres tablosuna ekleyecektir ve sonra, frame'i HostD'ye gönderecektir. Şayet HostD'nin MAC adresi, forward/fitler tablosunda bulunmasaydı, switch, frame'i, Fa0/3 port'u dışında, tüm port'lardan gönderecekti.

Şimdi, bir show mac address-table çıktısına bakalım:

```
Switch#sh mac address-table
Vlan    Mac Address       Type        Ports
   1    0005.dccb.d74b    DYNAMIC     Fa0/1
   1    000a.f467.9e80    DYNAMIC     Fa0/3
   1    000a.f467.9e8b    DYNAMIC     Fa0/4
   1    000a.f467.9e8c    DYNAMIC     Fa0/3
   1    0010.7b7f.c2b0    DYNAMIC     Fa0/3
   1    0030.80dc.460b    DYNAMIC     Fa0/3
   1    0030.9492.a5dd    DYNAMIC     Fa0/1
   1    00d0.58ad.05f4    DYNAMIC     Fa0/1
```

Yukarıdaki switch'in, aşağıdaki MAC adresleriyle bir frame aldığı farz edilmektedir:

```
Kaynak  MAC: 0005.dccb.d74b
Hedef    MAC: 000a.f467.9e8c
```

Switch, bu paketi ne yapacaktır? Cevap: hedef MAC adresi, MAC adresi tablosunda bulunacaktır ve frame, sadece Fa0/3 port'undan gönderilecektir. Hedef MAC adresinin, forward/filter tablosunda bulunmaması durumunda frame'i, hedef cihazı bulmak için tüm port'lardan göndereceğini hatırlayın. Şimdi, MAC adresi tablosunu, switch'lerin host adreslerini forward/fitler tablosuna nasıl eklediğini görebiliriz. Onu, yetkisiz kullanıcılardan nasıl koruyabiliriz?

## Port Güvenliği

Birinin kolayca switch port'larınızdan birine bağlanmasını ya da daha kötüsü, bir hub, switch veya access point'in, ofisinizdeki Ethernet prizine eklenmesini nasıl durdurursunuz? Varsayılan olarak MAC adresleri dinamik olarak forward/fitler veritabanında görünecektir. Onları, port security kullanarak, çabucak durdurabilirsiniz:

```
Switch#config t
Switch(config)#int f0/1
Switch(config-if)#switchport port-security ?
   aging            Port-security aging commands
   mac-address      Secure mac address
   maximum          Max secure addresses
   violation        Security violation mode
   <cr>
```

Yukarıdaki çıktıda açıkça görebilirsiniz ki, `switchport port-security` komutu, dört seçenekle kullanılabilmektedir. Ağımdaki kullanıcıları kolayca kontrol etmemi sağladığından, kişisel olarak, `port-security` komutundan hoşlanırım. Her port'a ayrı MAC adresleri tanımlamak için `switchport port-security mac-address mac-address` komutunu kullanabilirsiniz. Şayet bunu yapmak isterseniz, bol zamanınız olsa iyi olur!

Her port için sadece bir host makinesi kabul etmesi için bir switch port yapılandırmayı ve bu kuralın ihlali durumunda port'un kapatılmasını isterseniz, aşağıdaki komutu kullanın:

```
Switch#config t
Switch(config)#int f0/1
Switch(config-if)#switchport port-security maximum 1
Switch(config-if)#switchport port-security violation shutdown
```

Bu komutlar, muhtemelen en popüler olanlardır. Çünkü kullanıcıları, ofislerindeki bir switch'e ya da access point'e bağlanmasını engellemektedirler. Maximum'un 1'e ayarlanması, bu port'ta sadece bir MAC adresinin kullanılabileceği anlamına gelir. Kullanıcı, diğer bir host'u bu network segment'ine eklemeye çalışırsa switch portu, kapanacaktır. Şayet bu olursa, switch'e gitmek ve port'u `no shutdown` komutu ile etkinleştirmelisiniz.

Herhalde benim en favori komutlarımdan biri `sticky` komutudur. Sadece, iyi bir performans sağlamaz, güzel bir isme de sahiptir. Bu komutu, `mac-address` komutu altında bulabilirsiniz:

```
Switch(config-if)#switchport port-security mac-address sticky
Switch(config-if)#switchport port-security maximum 2
Switch(config-if)#switchport port-security violation shutdown
```

Aslında bunun yaptığı, ağdaki birinin MAC adresini yazmak zorunda kalmaksızın, statik MAC adresi güvenliği sağlamasıdır. Söylediğim gibi, çok güzel!

NOT *Port güvenliğine, bu modülün sonlarında, tekrar değineceğim.*

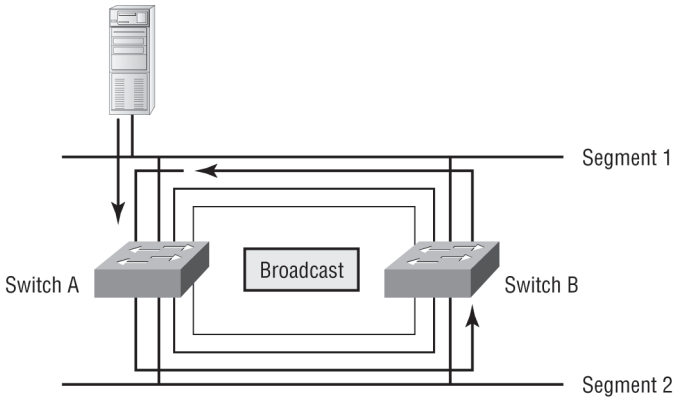
Yukarıdaki örnekte, ilk iki MAC adresi, statik adres gibi porta "yapıştırılacak" ve komutu tekrar yapılandırana kadar kalacaktır. Onu 2'ye neden ayarladım? Birine, PC/veri diğerine, telephony/telefon için ihtiyacım vardı. Bu tip konfigürasyona, sonraki VLAN'lerle ilgili modülde daha fazla değineceğim.

## Kısır Döngüden Kaçınma

Switch'ler arasındaki yedek linkler, bir linkin çalışmaması durumunda, bütün network'ün kullanılamaz duruma düşmesinden korudukları için iyidirler.

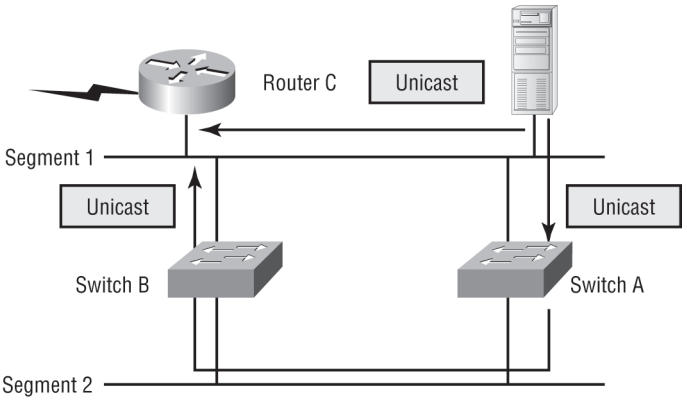
Kulağa hoş geliyor fakat yedek linkler, çok faydalı olmalarına rağmen çözdüklerinden daha fazla probleme neden olurlar. Frame'lerin tüm yedek linklerden eşzamanlı olarak gönderilmesinden dolayı, network kısır döngüleri ve diğer problemlere sebep olur. Korkunç problemlerin bazıları aşağıda listelenmiştir:

- Şayet, kısır döngü engelleme mekanizması yoksa switch'ler, broadcast'leri ağtopluluğu boyunca durmadan gönderecektir. Bu, bazen broadcast fırtınası olarak tanımlanmaktadır. (Fakat çoğu zaman, tekrarlanmasını istemediğimiz bir şey olarak beliritilir). Şekil 8.8 bir broadcast'in, network boyunca nasıl yayıldığını göstermektedir. Bir frame'in, ağ topluluğunun fiziksel network ortam aracı boyunca nasıl sürekli dolaştığını gözlemleyin.




***Şekil 8.8:*** *Broadcast fırtınası.*

- Frame, aynı anda farklı segment'lerden gelebileceğinden, bir cihaz aynı frame'in birçok kopyasını alabilir. Şekil 8.9, tüm frame demetinin, eşzamanlı olarak birçok segment'ten nasıl gelebileceğini göstermektedir. Şekildeki sunucu, RouterC'ye bir unicast frame'i gönderir. Bunun bir unicast frame'i olmasından dolayı, SwitchA, frame'i geçirir ve SwitchB'de aynı hizmeti sağlar. Bu kötü bir durumdur, çünkü RouterC, unicast frame'ini iki kez alır bu da, network'te ilave yüke sebep olur.




***Şekil 8.9:*** *Multiple frame copies.*

- Şunu düşünmüş olabilirdiniz: Switch'in frame'i, birden fazla linkten almasından dolayı, MAC adresi filtreleme tablosu, cihazın lokasyonu hakkında şaşkın olabilir. Dahası, şaşkın switch, bir frame'i gönderemeyeceği kaynak donanım adresi lokasyonuyla, MAC filtreleme tablosunu sürekli olarak güncellemek durumunda kalabilir. Bu MAC tablosunu yumruklama olarak isimlendirilir.
- Olabilecek en kötü şeylerden birisi de, bir network boyunca, çok sayıda döngünün üretilmesidir. Bunun anlamı, döngülerin, diğer döngülerde oluşmasıdır. Şayet bir broadcast fırtınası da oluşursa, network, frame switching'i yapamayacaktır.

Tüm bu problemler, trajediye dönüşür (en azından ona yaklaşır) ve kaçınılması ya da birinin çözmesi gereken kötü bir durumdur. Burası Spanning Tree Protokolünün oyuna girmesi gereken yerdir. STP, size bahsettiğim problemleri çözmek için geliştirilmiştir.

# Spanning Tree Protokolü (STP)

Bir zamanlar, Digital Equipment Corporation (DEC) olarak bilinen bir şirket satın alındı ve adı Compaq olarak değiştirildi. Fakat bundan önce, DEC, Spanning Tree Protocol(STP)'nin orjinal versiyonunu geliştirdi. IEEE, daha sonra 802.1D dediği kendi versiyonunu oluşturdu. Kötü haber, varsayılan olarak Cisco switch'lerin, DEC versiyonu ile uyumlu olmayan IEEE 802.1D çalışmasıdır. İyi haber, Cisco'nun, yeni switch'lerinde, 802.1w denilen diğer endüstri standardına doğru gitmesidir. Bu bölümde, STP versiyonlarına da değineceğim, fakat ilk olarak, bazı önemli STP temellerini açıklayalım.

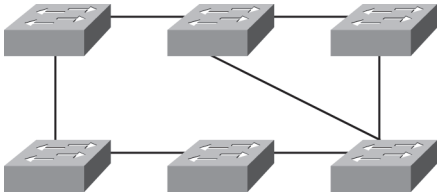
STP'nin ana görevi, katman2 ağınızda (bridge'ler ya da switch'ler) oluşan network döngülerini durdurmaktır. Tüm linkleri bulmak için, ağı sürekli izler ve yedek linkleri kapatarak, kısır döngü olmadığından emin olur. STP, ilk olarak bir topoloji veritabanı oluşturmak ve sonra araştırıp yedek linkleri ortadan kaldırmak için spanning-tree algoritması (STA) kullanır. STP ile frame'ler sadece öncelikli, STP'nin seçtiği linklerden gönderilecektir.

NOT *STP, döngü olmayan bir switch ağını korumak için kullanılan katman2 protokolüdür.*

Şimdiki bölümlerde, Spanning Tree Protokolü'nün önemli kısımlarına gireceğim.

Spanning Tree Protokolü, Şekil 8.10'da gösterilen ağlarda gerekmektedir.

Şekil 8.10'da, yedekli topoloji'de (switching döngüleri) bir switch ağına sahibiz. Network kısır döngülerini durdurmak için bazı katman2 mekanizmaları olmadan, önceden bahsettiğimiz şu problemleri yaşarız: Broadcast fırtınası ve birçok frame kopyası.



***Şekil 8.10:*** *Kısır döngü oluşan bir switch ağı.*

NOT *Şekil 8.10'daki network, her ne kadar yavaş olsa da, bir şekilde çalışır. Bu, açıkça switching döngüsü tehlikesini göstermektedir. Daha kötüsü, başladığında bu problemi tespit etmek çok zor olabilir.*

## Spanning Tree Terimleri

STP'nin, network'te nasıl çalıştığıyla ilgili detayları açıklamadan önce, bazı temel düşünceler ile terimleri ve bunların katman2 switch network'le nasıl ilgileri olduğunu anlamanız gerekmektedir:

**Root bridge:** Root bridge, en iyi bridge ID'ye sahip bridge'dir. STP ile ağda merkezi nokta olan bir root bridge seçilmesi, ağdaki tüm switch'ler için önemlidir. Hangi port'un bloklanacağı, hangisinin forwarding moda konacağı gibi ağdaki tüm kararlar, bu root bridge'in perspektifinden yapılmaktadır.

**BPDU:** Tüm switch'ler, hem root switch seçiminde hem de sonraki network konfigürasyonunda kullanmak için bilgileri değişirler. Her switch, bir komşusundan aldığını, diğer komşusuna gönderdiği Bridge Protocol Data Unit'deki (BPDU) parametreleri karşılaştırır.

**Bridge ID:** Bridge ID, ağdaki tüm switch'ler için STP'nin tuttuğu kayıttır. Bu, bridge priority'si (Cisco switch'lerde varsayılan olarak 32,768'dir) ve MAC adresi kombinasyonuyla belirlenmektedir. En düşük bridge ID'sine sahip bridge, ağdaki root bridge olmaktadır.

**Nonroot bridge'ler:** Bunlar, root olmayan bridge'lerdir. Nonroot bridge'ler, tüm bridge'lerle BPDU'larını değişir ve tüm switch'lerdeki STP topoloji veritabanını günceller. Döngüleri engeller ve link arızalarına karşı savunma önlemi sağlarlar.

**Port cost'u:** Port cost'u, switch'ler arasında çoklu linkler kullanıldığında ve linklerden hiçbirinin root port'u olmadığında, en iyi yolu belirler. Bir linkin cost'u, bir linkin bant genişliğiyle belirlenmektedir.

**Root port:** Root port daima, bridge'e direkt bağlı linktir ya da root bridge'e en kısa yoldur. Şayet, root bridge'e birden fazla link bağlıysa port cost'u, her linkin bant genişliği kontrol edilerek belirlenir. En düşük cost'lu port, root port olur. Şayet çoklu linkler, aynı cost'a sahiplerse, düşük bridge ID'li bridge kullanılacaktır. Çoklu linkler, aynı cihazdan olabileceğinden, en düşük port numarası kullanılacaktır.

**Designated port:** Bir designated port, en iyi (düşük) cost'a sahip olarak belirlenendir. Bir designated port, forwarding port olarak işaretlenecektir.

Nondesignated port: Bir nondesignated port, designated port'tan daha yüksek cost'lu bir port'tur. Nondesignated portlar, blocking moda konurlar. Forwarding port değillerdir.

**Forwarding port**: Bir forwarding port, frame'leri iletir.

**Blocked port:** Bir blocked port, kısır döngüleri engellemek için frame'leri iletmeyecektir. Bununla beraber bir blocked port, frame'leri daima dinleyecektir.

## Spanning Tree Operasyonları

Daha önce söylediğim gibi, STP'nin işi, ağdaki tüm linkleri tespit etmek ve yedek olanlarını kapatmaktır. Böylelikle, network kısır döngülerinin olması engellenmektedir.

STP bunu, ilk olarak, tüm port'ları boyunca iletecek ve STP domain'indeki tüm diğer cihazlar için referans noktası gibi davranacak bir root bridge seçerek başarır. Tüm switch'ler, root bridge'in kim olacağı konusunda anlaşınca, her bridge, kendi seçilmiş root port'unu bulmak zorundadır. İki switch arasındaki her link, root'a en yüksek bant genişliğini sağlayan bir tane designated port'a sahip olmalıdır. Bir bridge'in, root'a ulaşmak için diğer bridge'leri inceleyebileceğini hatırlamak çok önemlidir. Yani, her zaman en kısa olan değil, en hızlı (en yüksek bant genişliğine sahip) yol kullanılacaktır.

Root olmadan, root'a, root olmaktan daha yakın olamayacağınızdan, root switch'teki her port, bir designated port'tur. Olaylar yatıştıktan sonra, ne root ne de designated port olan herhangi bir nonroot, nondesignated port, blocking durumuna yerleştirilmiştir. Böylece, switching döngüleri engellenir.

Eğer, kararlar veren birden fazla kişi yoksa işler daha düzgün yürüme eğilimindedir. Benzer şekilde, verilen network'te sadece bir root Bridge olabilir. Gelecek bölümde, root bridge seçim prosesini tamamıyla inceleyeceğiz.

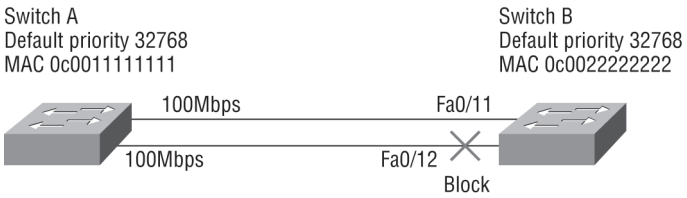
### Root Bridge Seçimi

Bridge ID, STP domain'inde root bridge'i seçmek ve STP domain'indeki diğer cihazların her biri için root portunu belirlemek için kullanılmaktadır. Bu ID, 8 byte boyutundadır ve priority ve cihazın MAC adresini içermektedir. IEEE STP çalışan tüm cihazlarda varsayılan priority, 32,768'tir.

Root bridge'i belirlemek için, her bridge'in priority'sini, MAC adresi ile beraber kullanırsınız. Şayet iki switch ya da bridge, aynı priority değerine sahipse, hangisinin en düşük (iyi) ID'ye sahip olduğunu anlamak için, MAC adresi, belirleyici olacaktır. Bu, şöyle olmaktadır: Adı A ve B olan iki switch, 32768, varsayılan priority'i kullanıyorlarsa, MAC adresleri kullanılacaktır. Şayet, SwitchA'nın MAC adresi, 0000.0c00.1111 ve SwitchB'nin MAC adresi, 0000.0c00.2222 ise, SwitchA, root bridge olacaktır. Root bridge seçimi olduğunda daha düşük değerin, en iyi olduğunu hatırlayın.

Varsayılan olarak, BPDU'lar, bridge/switch'teki tüm aktif portlardan her iki saniyede bir gönderilecektir. (En düşük (iyi) bridge ID'si olan bridge, root bridge'tir.) Bridge'in ID'sini, priority'sini değiştirerek değiştirebilirsiniz. Böylece o, otomatik olarak root bridge olacaktır. Bunu yapabilmek büyük network'lerde önemlidir. Burada sizin istediğiniz verimliliktir.

Şekil 8.11, yedek yolları olan, tipik bir switch network'ünü göstermektedir. İlk olarak, hangi switch'in root olduğunu anlayalım ve sonra, switch'in priority'sini değiştirerek, nonroot bridge'i root bridge yapalım.




***Şekil 8.11:*** *Yedek yollara sahip bir switch ağı.*

Şekil 8.11'e bakarak, en düşük bridge ID'ye sahip olduğundan, SwitchA'nın root bridge olduğunu söyleyebilirsiniz. SwitchB, bir switching döngüsü olmasını engellemek için, SwitchA'ya bağlı port'larından birini kapatmalıdır. SwitchB'nin, blocked port'larından iletmese bile, oradan, BPDU'ları alacağını hatırlayın.

STP'nin, SwitchB'deki hangi port'u kapatacağını belirlemek için, ilk olarak tüm linklerin, bant genişliğini kontrol edecektir ve sonra düşük bant genişliği değerine sahip linki kapatacaktır. SwitchA ve SwitchB arasındaki her iki link 100Mbps olduğundan, STP tipik olarak, daha yüksek port numaralı olanı kapatacaktır, fakat her zaman değil. Bu örnekte 12, 11'den yüksektir, bu nedenle port 12, blocking moda yerleştirilecektir.

Varsayılan priority'i değiştirmek, bir root bridge seçmenin en iyi yoludur. Bu önemlidir, çünkü network'ünüzde, core switch'in, root switch olmasını istersiniz. Böylece, STP daha hızlı converge olacaktır.

Gelin biraz eğlenelim ve SwitchB'yi, network'ümüzde root yapalım. Aşağıda, varsayılan priority'e sahip SwitchB'den çıktı vardır. `Show spanning-tree` komutunu kullanacağız:

```
Switch B(config)#do show spanning-tree
VLAN0001
 Spanning tree enabled protocol ieee
 Root ID     Priority    32769
             Address     0005.74ae.aa40
             Cost        19
             Port        1 (FastEthernet0/1)
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

 Bridge ID   Priority    32769  (priority 32768 sys-id-ext 1)
             Address     0012.7f52.0280
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
             Aging Time 300
[output cut]
```

Başlangıçta dikkat etmek gereken iki şey vardır: SwitchB, IEEE 801.d protokolü çalışıyor ve ilk çıktı (RootID), switch network'ü için root bridge bilgisidir. Fakat o, SwitchB değildir. SwitchB'nin root bridge'e giden port'u (root port olarak belirtilir), port1'dir. Bridge ID, SwitchB için, gerçek spanning tree bilgisidir ve VLAN1 (VLAN0001 olarak listelenmiştir) için, her VLAN, nadir olsa da, farklı bir root bridge'e sahip olabilir. SwitchB'nin MAC adresi de listelenmiştir ve root bridge'in MAC adresinden farklı olduğunu görebilirsiniz.

SwitchB'nin priority'si, her switch için varsayılan olan 32,768'dir. Onu, 32769 olarak listelenmiş görürsünüz, fakat gerçek VLAN ID'si eklenmiştir, bu nedenle bu örnekte o, VLAN1 için 32769 olarak görünür. VLAN2, 32770 olacaktır ve öyle devam eder.

Söylediğim gibi, STP network'ünüzün root'u yapmak amacıyla bir switch'i zorlamak için, priority'i değiştirebilirsiniz. Gelin bunu SwitchB için yapalım. Bir Cataylst switch'teki bridge priority'sini değiştirmek için aşağıdaki komutu kullanın:

```
Switch B(config)#spanning-tree vlan 1 priority ?
  <0-61440>  bridge priority in increments of 4096
Switch B(config)#spanning-tree vlan 1 priority 4096
```

Priority'i 0'dan 61440'a kadar herhangi bir değere ayarlayabilirsiniz. Onu sıfıra ayarlamanın anlamı, switch'in daima root bridge olacağıdır. Bridge priority'si, 4096'nın katları ile ayarlanır. Şayet bir switch'in, network'ünüzdeki tüm VLAN'ler için root bridge olmasını istiyorsanız, her VLAN için priority'i, kullanabileceğiniz en düşük priority olan sıfıra ayarlamalısınız. Tüm switch'lerin, priority'lerini sıfıra ayarlamak çok avantajlı olmayacaktır.

Aşağıdaki çıktıyı kontrol edin. Şimdi SwitchB'nin priority'sini, VLAN1 için 4096 olarak değiştirdik. Bu switch'in root olmasını başarıyla zorladık:

```
Switch B(config)#do show spanning-tree
VLAN0001
 Spanning tree enabled protocol ieee
 Root ID     Priority    4097
             Address     0012.7f52.0280
             This bridge is the root
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

 Bridge ID   Priority    4097   (priority 4096 sys-id-ext 1)
             Address     0012.7f52.0280
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
             Aging Time 15
[output cut]
```

SwitchB'nin hem MAC adresi hem de bridge ID'si, aynıdır. Yani, SwitchB şimdi, root bridge'tir. Show spanning-tree komutunu bilmek, çok önemlidir. Bu komutu, bölümün sonuna doğru tekrar kullanacağız.

*İster inanın ister inanmayın, root bridge'inizi ayarlamak için başka bir komut daha vardır. Switch konfigürasyonumu gösterdikten sonra, söz veriyorum ondan bahsedeceğim.*

NOT

## Spanning Tree Port Durumları

STP çalışan bridge ya da switch portlar, beş farklı duruma geçerler:

**Blocking:** Bloklanmış bir port, frame'leri iletmeyecektir, sadece BPDU'ları dinleyecektir. Blocking port'un amacı, döngüdeki yolların kullanımını engellemektir. Switch ilk açıldığında, varsayılan olarak tüm port'lar, blocking durumundadır.

**Listening:** Port, veri frame'lerini geçirmeden önce, network'te döngü olmadığından emin olmak için BPDU'ları dinler. Listening durumundaki bir port, veri frame'lerini, MAC adres tablosuna yerleştirmeden iletmek için hazırlar.

**Learning:** Switch port'u, BPDU'ları dinler ve switch network'ündeki tüm yolları öğrenir. Learning durumundaki bir port, MAC adresi tablosuna yerleşir, fakat veri frame'lerini iletmez. Forward delay, varsayılan olarak 15 saniyeye ayarlanan, bir port'un listening'ten, learning moda dönünceye kadar geçen zamandır ve show spanning-tree çıktısında görülebilir.

**Forwarding:** Port, bridge port'taki tüm veri frame'lerini alır ve gönderir. Şayet port, learning durumunun sonunda, hala designated ya da root port ise, forwarding durumuna girer.

NOT *Switch'ler, MAC adresi tablosuna, sadece learning ve forwarding modda yerleşirler.*

**Disabled:** Disabled durumundaki bir port, STP'nin ya da iletilen frame'in bir parçası olmaz. Disabled durumdaki bir port, hemen hemen işlevsel değildir.

Switch port'ları, genelde blocking ve forwarding durumdadırlar. Bir forwarding port, root bridge için en düşük (iyi) cost'a sahip olduğu belirlenen port'tur. Şayet topoloji değişikliği olduğunda (bir linkin arızalanması ya da birilerinin yeni bir switch eklemesiyle), bir switch'in port'larını, listening ve learning olarak bulursunuz.

Açıkladığım gibi, blocking port'lar, network döngülerini engellemek için bir stratejidir. Bir switch, root bridge için en iyi yolu belirleyince, diğer yedek port'ların hepsi, blocking modda olacaktır. Bloklanmış portlar, hala BPDU'ları alabilecektir. Sadece herhangi bir frame gönderemezler.

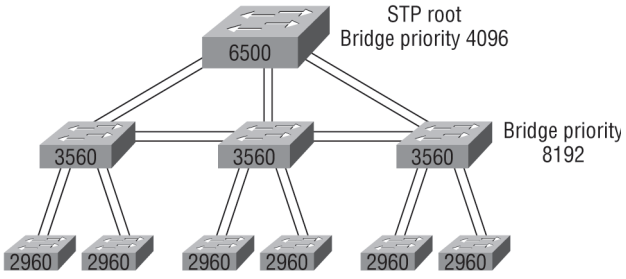
Şayet switch, bir topoloji değişikliği ile bloklanmış bir port'un, şimdi designated ya da root olduğunu fark ederse, listening moda geçecektir ve port, forwarding moda dönünce, bir döngü oluşturmayacağına emin olmak için, aldığı tüm BPDU'ları kontrol eder.

## Convergence

Bridge ve switch'lerdeki tüm port'lar, forwarding ya da blocking moda geçtiğinde, convergence olur. Convergence tamamlanıncaya kadar, hiçbir data iletilmeyecektir. Ve verinin tekrar iletilmeye başlamasından önce, tüm cihazların güncellenmeleri gerekir. Evet, doğru okudunuz: STP, converge olurken, tüm host verisi, aktarımı durdurur! Şayet, network kullanıcılarınızla bağlantıda kalmak (ya da sürekli çalışıyor durumda olmak) istiyorsanız, switch network'ünüzün, fiziksel olarak oldukça iyi tasarlandığından ve böylece STP'nin hızlı bir şekilde converge olduğundan emin olmalısınız.

Şekil 8.12 size, STP'nin etkin olarak converge olmasını sağlayacak, switch network'ünüzün tasarımı ve kurulumunda göz önünde bulundurulması gereken bazı önemli faktörleri göstermektedir.

Tüm cihazların aynı veritabanına sahip olmasını garantilediğinden, convergence gerçekten önemlidir. Fakat özellikle belirttiğim gibi, o size maliyet getirecektir. Genellikle, blocking'ten, forwarding'e geçmesi, 50 sn alır ve ben STP timer'larının değiştirilmesini tavsiye etmem.(fakat gerektiğinde bu timer'ları değiştirebilirsiniz). Fiziksel switch tasarımınızı, Şekil 8.12'de görüldüğü gibi, hiyerarşik bir tarzda oluşturarak, core switch'inizi, STP root'u yapabilirsiniz. Bu, STP converge zamanını hızlandıracaktır.




Daha hızlı STP convergence için core switch'i STP root olarak oluşturun.

***Şekil 8.12:*** *Optimal bir hiyerarşik switch tasarımı.*

Normal spanning tree topoloji'sinin, bir switch port'undaki blocking moddan forwarding moda converge süresi 50 saniye olduğundan, sunucu ya da host makinelerinde, zaman aşımı oluşturabilir (örneğin, onları reboot ettiğinizde). Bu aksaklığı engellemek için spanning tree'yi, `PortFast` kullanarak, belirli port'larda pasif hale getirebilirsiniz.

## Spanning Tree PortFast

STP devreden çıktığında, switching loop oluşturmayacağından tamamıyla emin olduğunuz switch'inize sunucu ya da başka cihazlar bağlandıysa, bu port'larda, `portfast` kullanabilirsiniz.

Onu kullanmanın anlamı, port'ların, STP converge olurken, forwarding moda erişmek için her zamanki 50 saniyeyi harcamayacağıdır.

Aşağıda, oldukça basit olan, komutları görebilirsiniz:

```
Switch(config-if)#spanning-tree portfast ?
  disable  Disable portfast for this interface
  trunk    Enable portfast on the interface even in trunk mode
  <cr>
```

Trunk port'ları henüz tartışmadık. Fakat bunlar, basit olarak, switch'leri birbirine bağlamak ve birbirlerine VLAN bilgilerini aktarmak için kullanılmaktadır. Şayet onu bir trunk port'ta etkinleştirmek istiyorsanız, özellikle `portfast` demek zorundasınız. Switch'ler arasındaki port'lar, genelde STP çalıştırdığından, bu tipik bir konfigürasyon değildir. Bu nedenle, bir interface'de portfast'i açtığımda, aldığım mesajlara bakalım:

```
Switch(config-if)#spanning-tree portfast
%Warning: portfast should only be enabled on ports connected to a
   single host. Connecting hubs, concentrators, switches, bridges,
   etc... to this interface  when portfast is enabled, can cause
   temporary bridging loops.
 Use with CAUTION
%Portfast has been configured on FastEthernet0/1 but will only
 have effect when the interface is in a non-trunking mode.
Switch(config-if)#
```

Portfast, f0/1 port'unda etkinleştirilmiştir, fakat dikkatli olmanızı söyleyen, oldukça uzun mesaja dikkat edin. Size bahsedeceğim son faydalı interface komutu, `range`'dir. Onu, aynı anda birçok port'u yapılandırmanıza yardımcı olması için switch'lerde kullanabilirsiniz. İşte bir örnek:

```
Switch(config)#int range fastEthernet 0/1 - 12
Switch(config-if-range)#spanning-tree portfast
```

Yukarıdaki `range` komutu bana, bir komut yazarak ve basitçe Enter'a basarak, switch portlarımın 12'sini de portfast moda sokmama izin verir. Herhangi bir döngü oluşturmadığımı umuyorum! `Portfast` komutuyla çok dikkatli olun. Ayrıca, herhangi bir komutla beraber, `interface range` komutunun kullanılabileceğini bilmenizi istiyorum. Örnek olarak, portfast komutu ile beraber kullandım.

## Spanning Tree UplinkFast

UplinkFast, link arızası durumunda, STP'nin converge zamanını geliştiren, Cisco'ya özel bir özelliktir. Ve portfast komutunda olduğu gibi, bu komutu kullandığınız yere çok dikkat etmek zorundasınız. UplinkFast özelliği, switch, en az bir alternatif/yedek root port'a (blocking durumda bir port'a) sahip olduğunda bir switch ortamında çalışması için tasarlanmıştır. Bundan dolayı, Cisco, UplinkFast'in sadece, bloklanmış port'lu switch'ler ve tipik olarak, Access katmanında olanlar için etkinleştirilmesini tavsiye etmektedir.

UplinkFast, bir switch'in, öncelikli linkin arızalanmasından önce, root bridge için alternatif yolları bulmasına izin verir. Yani, öncelikli link arızalanırsa, ikincil link, daha çabuk ortaya çıkacaktır. Port, normal 50 saniye STP converge zamanını beklemeyecektir. Şayet, 802.1d STP çalıştırıyorsanız ve Access katmanı switch'lerinizde yedek linkleriniz varsa, kesinlikle UplinkFast'i aktif yapmak istersiniz. Fakat Cisco multilayer tasarımındaki dağıtım ya da core switch'ler için kullanılan bir alternatif/yedek root linki belirten topoloji bilgisi olmadan, onu switch'lerde kullanmayın.

### Spanning Tree BackboneFast

Lokal switch'teki link arızalarını belirlemek ve hızlıca düzeltmek için kullanılan UplinkFast'in tersine, switch'e direkt bağlı olmayan bir link arızalandığında, convergence'i hızlandırmak için, BackboneFast denilen, diğer Cisco tescilli STP eklentisi kullanılır. Şayet BackboneFast çalışan bir switch, designated bridge'den ikinci derece bir BPDU alırsa, root'a giden yoldaki bir linkin arızalandığını anlar. Bunu netleştirdiğinizden emin olmanız için, ikinci derece bir BPDU, root bridge ve designated bridge için aynı switch'i listeleyen BPDU' dur.

Ve tekrar, sadece Access katmanı switch'lerde ve yedek linkleri ve blocking modda en az bir linki olan switch'lerde yapılandırılan UplinkFast'in tersine, BackboneFast, direkt olmayan link hatalarının tespitine izin vermesi için tüm Catalyst switch'lerde etkinleştirilmelidir. BackboneFast'i etkinleştirmek, Spanning Tree tekrar konfigürasyonunu daha hızlı başlattığından ayrıca faydalıdır. Varsayılan 50 saniye STP converge süresinden, 20 saniye kazandırabilir.

### Rapid Spanning Tree Protocol (RSTP) 802.1w

Switch network'ünüzde (switch markasına bakmaksızın) çalışan iyi bir STP konfigürasyonunuzun ve henüz bahsettiğimiz ve her switch'te etkin olan tüm özelliklere sahip olmak istermisiniz? Kesinlikle evet! Güzel, o zaman, Rapid Spanning Tree Protocol'ün (RSTP) dünyasına hoş geldiniz.

Cisco, sunulan IEEE 802.1 standartlarının boşluklarını ve dezavantajlarını düzeltmek için PortFast, UplinkFast ve BackboneFast'i geliştirdi. Bunların dezavantajı, sadece Cisco tescilli olmaları ve ilave konfigürasyona ihtiyaç duymalarıdır. Fakat yeni 802.1w standardı (RSTP), bütün bu sorunları bir pakette inceler. Sadece RSTP'yi etkinleştirin ve devam edin. Network'ünüzdeki tüm switch'lerin, 802.1w'nun tam anlamıyla çalışması için 802.1w çalıştığını garantilemeniz önemlidir.

NOT *Bir sürpriz olarak gelebilir, fakat RSTP, eski STP protokolleri ile birlikte çalışabilir. Sadece şunu bilin ki, 802.1'in aslında hızlı convergence kabiliyeti, eski bridge'lerle kullanıldığında kaybolmaktadır.*

RSTP'nin nasıl yapılandırılacağını, bu bölümde ilerde göstereceğim.

### EtherChannel

Yedek linklere sahip olmak ve BLK(blocked) moddaki linklerden birine koymak için STP'ye izin vermek yerine, linkleri birleştirebilir ve mantıksal bir aggregation oluşturabilirsiniz. Böylece, birçok linkimiz, tek gibi görünecektir. Bunu yapmak, STP ile aynı yedekliliği sağlayacaksa, yedek linklerimizi neden birleştirmeyelim?

Her zamanki gibi, Etherchannel'in Cisco versiyon'u ve IEEE versiyonu vardır. Cisco versiyonu, Port Aggregation Protocol (PAgP) olarak tanımlanır ve IEEE 802.3ad standardı, Link Aggregation Control Protocol (LACP) olarak isimlendirilir. Her iki standart da eşit olarak çalışır, ikisinin konfigürasyonu farklıdır. Bu bölümün sonuna doğru, bunu göstermek için bazı linkleri birleştireceğim. Endişelenmeyin, STP eklentileri için tüm konfigürasyonları şimdiki bölümde inceleyeceğim.

## Catalyst Switch'leri Yapılandırmak

Cisco Catalyst switch'ler, çok geniş yelpazede olabilirler. Sarmal-çift ve fiber kombinasyonunda, bazıları 10Mbps çalışır, bazısı, 10Gbps switched porta sahip olabilir. Bu yeni switch'ler (özellikle 2960 ve 3560'lar), daha akıllıdırlar, bu nedenle size hem daha hızlı veri hem de video ve ses servisleri sağlayabilirler.

Şimdi ona başlama zamanıdır. Command line interface (CLI) kullanarak, bir Cisco Catalyst switch'in nasıl başlatılacağını ve yapılandırılacağını göstereceğim. Bu modülde, temel komutları iyice kavradıktan sonra, virtual LAN(VLAN)'ların, artı Inter-Switch Link (ISL), 802.1q routing ile Cisco's Virtual Trunk Protocol (VTP)'nin nasıl yapılandırılacağını göstereceğim.

Aşağıda, bu bölümde işleyeceğimiz temel görevlerin listesini bulabilirsiniz.

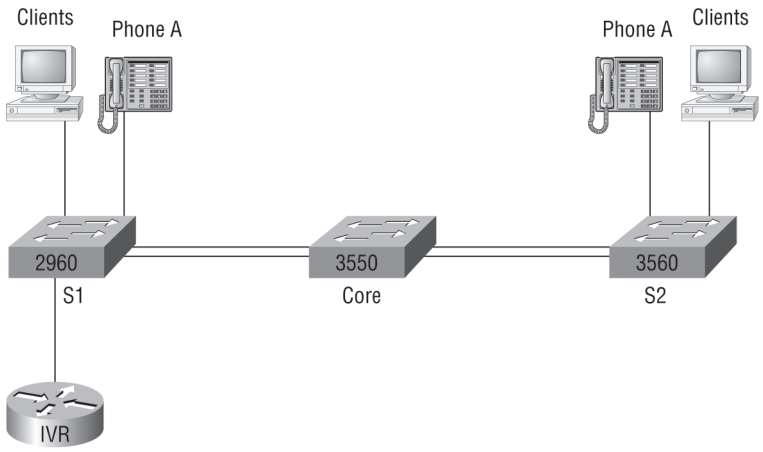
- Yönetimsel fonksiyonlar
- IP adresi ve subnet maskı yapılandırmak
- IP varsayılan ağ geçidini ayarlamak
- Port güvenliğini ayarlamak
- PortFast ayarlamak
- BPDUGuard ve BPDUFilter'ı etkinleştirmek
- UplinkFast'i etkinleştirmek
- BackboneFast'i etkinleştirmek
- RSTP (802.1w)'i etkinleştirmek
- EtherChannel'ı etkinleştirmek
- Bir STP root switch yapılandırmak
- Bir switch'i yapılandırmak için CNA'i kullanmak

*Cisco Catalyst ailesi hakkında her şeyi, www.cisco.com/en/US/products/hw/switches/index.html adresinden öğrenebilirsiniz.*

NOT

## Catalyst Switch'lerin Konfigürasyonu

Bölüm 6 ve 7'de yapılandırdığımız router'larla yaptığımız gibi, hem bu modülde hem de bölüm 9, "Virtual LAN'lar (VLAN'lar)"da yapılandırmak için bir diyagram ve switch kurulumu kullanacağız. Şekil 8.13, çalışacağımız switch ağını göstermektedir.




***Şekil 8.13:*** *Switch ağımız.*

Yeni olan 3560, 2960 ve 3550 switch'leri kullanacağım. Ağ'da görülen host, telefon ve router'ların, Modül9'a geçtiğimizde daha önemli olacağını aklınızda tutun.

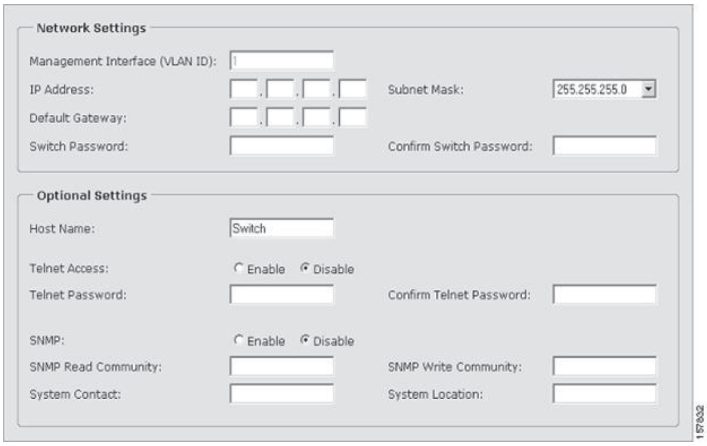
Catalyst switch'lerden birinin konfigürasyonuna geçmeden önce, bölüm 4'te router'larla yaptığım gibi, bu switch'lerin boot prosesi hakkında size bilgi vereceğim. Şekil 8.14, tipik bir Cisco Catalyst switch'in detayını göstermektedir ve bu ürünün farklı interface'leri ve özellikleri hakkında size bilgi vermek gerekiyor.


***Şekil 8.14:*** *Bir Cisco Catalyst switch.*

Bilmenizi istediğim ilk şey, Catalyst switch'ler için konsol port'unun, genellikle switch'in arkasında olmasıdır. Fakat şekilde gösterilen 3560 gibi daha küçük switch'lerde konsol, daha kolay kullanılması için önde sağ taraftadır (sekiz port'lu 2960, tamamıyla aynı görünür). Şayet POST başarıyla tamamlanırsa, sistem LED'i yeşile döner. Şayet POST'ta hata olursa,amber rengine dönecektir. Amberin parlak olarak görülmesi, oldukça kötüdür (genellikle büyük bir hata var demektir). Bu

nedenle, etrafta yedek bir switch tutmayı isteyebilirsiniz (özellikle de bunun gerçek switch'lerde olması durumunda). Alttaki buton, Power over Ethernet (PoE) içeren ışıkları göstermesi için kullanılmaktadır. Bunu, Mode butonuna tıklayarak görebilirsiniz. PoE, bu switch'lerdeki güzel bir özelliktir. POE, sadece Ethernet kablosuyla switch'e bağlayarak, access point ve telefonuma güç sağlamama izin vermektedir.

Bir switch, boot olduktan sonra, Express Setup HTTP ekranını kullanabilirsiniz. Şekil 8.15, yeni bir switch'e bağlandığınızda ve browser'ınızın HTTP alanında, 10.0.0.1 kullandığınızda alacağınız ekranı göstermektedir. Host'larınızın aynı subnet'te olması gerekmektedir.




***Şekil 8.15:*** *Express Setup HTTP ekranı.*

Ekran bize, bazı temel fonksiyonları ayarlayabileceğimizi göstermektedir. Bana, CLI'dan konfigürasyon daha kolay gelmektedir fakat bu, seçeneklerinizden sadece bir tanesidir. Bu switch'in IP adresini, mask'ını, varsayılan ağ geçidini ve şifrelerini yapılandırabilirsiniz. Ayrıca, yönetim VLAN'ını da yapılandırabilirsiniz. Fakat şimdilik bunu yapmanızı ertelemenizi isteyeceğim ve bunu nasıl yapacağınızı, bir sonraki bölümde göstereceğim. Devam ederek, isteğe bağlı olarak hostname, sistem sorumlusu bilgisi ile lokasyonu yapılandırabilir ve Telnet erişimi kurabilirsiniz. Ve son olarak, Express Setup HTTP ekranı, size, SNMP ile switch kurulumunuzda bazı basit faydalar sağlayabilir. Böylece, Network Management System (NMS), onu bulabilir.

Şimdi, switch'lerimizi birbirine bağlayacaksak, Şekil 8.13'te görüldüğü gibi ilk olarak, switch'ler arasında çapraz kabloya ihtiyacımız olacağını hatırlayın. 2960 ve 3560 switch'lerim, bağlantı tiplerini otomatik olarak algılar, bu nedenle, düz kablo kullanabilirdim. Fakat bir 2950 ve 3550 switch, kablo tipini otomatik algılamaz. Farklı switch'lerin, farklı ihtiyaçları ve yetenekleri vardır, farklı switch'lerinizi birbirine bağladığınızda, bunu aklınızda tutun.

Switch port'larını diğerleriyle ilk bağladığınızda, link ışıkları turuncudur ve sonra yeşile dönmesi, normal operasyonu belirtir. Bu, Spanning Tree converge işlemidir ve bildiğiniz gibi bazı eklentiler etkinleştirilmediyse bu proses 50 saniye civarındadır. Fakat bir switch port'unu bağladınız ve switch port LED'i, yeşil ve amber arasında değişiyorsa, bu, port'ta bir hata olduğu anlamına gelir. Şayet bu olursa, host'un NIC kartını ya da kabloyu kontrol edin.

## S1

Gelin, her switch'e bağlanarak ve yönetimsel fonksiyonları ayarlayarak, konfigürasyonumuza başlayalım. Ayrıca, her switch'e, IP adresi de vereceğiz, fakat bu, ağımızı işlevsel hale getirmek için gerekmemektedir. Bunu yapmamızın tek sebebi, onu yönetebilmemizdir. 192.168.10.16/28 gibi basit bir IP tasarımı kullanalım. Bu mask, size tanıdık gelecektir. Aşağıdaki çıktıyı kontrol edin:

```
Switch>en
Switch#config t
Enter configuration commands, one per line. End with CNTL/Z.
```

```
Switch(config)#hostname S1
S1(config)#enable secret todd
S1(config)#int f0/1
S1(config-if)#description 1st Connection to Core Switch
S1(config-if)#int f0/2
S1(config-if)#description 2nd Connection to Core Switch
S1(config-if)#int f0/3
S1(config-if)#description Connection to HostA
S1(config-if)#int f0/4
S1(config-if)#description Connection to PhoneA
S1(config-if)#int f0/8
S1(config-if)#description Connection to IVR
S1(config-if)#line console 0
S1(config-line)#password console
S1(config-line)#login
S1(config-line)#exit
S1(config)#line vty 0 ?
  <1-15>  Last Line number
  <cr>
S1(config)#line vty 0 15
S1(config-line)#password telnet
S1(config-line)#login
S1(config-line)#int vlan 1
S1(config-if)#ip address 192.168.10.17 255.255.255.240
S1(config-if)#no shut
S1(config-if)#exit
S1(config)#banner motd # This is the S1 switch #
S1(config)#exit
S1#copy run start
Destination filename [startup-config]? [enter]
Building configuration...
[OK]
S1#
```

Bununla ilgili dikkat çeken ilk şey, switch interface'lerinde ayarlanmış IP adresi olmamasıdır. Bir switch'teki tüm port'lar, varsayılan olarak etkindir, daha fazla konfigürasyona gerek yoktur. IP adresi, bir yönetim domain'i ya da VLAN olarak belirtilen, mantıksal bir interface altında ayarlanmaktadır. Bizim burada yaptığımız gibi, bir switch ağını yönetmek için, varsayılan olarak VLAN1'i kullanırsınız. Geri kalan konfigürasyon, router konfigürasyonu için kullandığınız prosesle, nerdeyse aynıdır. Hatırlayın, switch interface'lerinde IP adresi yok, routing protokolü yok vs. Bu noktada, katman2 switching çalıştırıyoruz, routing değil. Ayrıca, Cisco switch'lerde aux port'u olmadığını da not alın.

## S2

S2 konfigürasyonu şöyledir:

```
Switch#config t
Switch(config)#hostname S2
S2(config)#enable secret todd
S2(config)#int fa0/1
2(config-if)#description 1st Connection to Core
S2(config-if)#int fa0/2
S2(config-if)#description 2nd Connection to Core
S2(config-if)#int fa0/3
S2(config-if)#description Connection to HostB
S2(config-if)#int fa0/4
S2(config-if)#description Connection to PhoneB
S2(config-if)#line con 0
S2(config-line)#password console
S2(config-line)#login
S2(config-line)#exit
S2(config)#line vty 0 ?
  <1-15>  Last Line number
  <cr>
S2(config)#line vty 0 15
S2(config-line)#password telnet
S2(config-line)#login
S2(config-line)#int vlan 1
S2(config-if)#ip address 192.168.10.18 255.255.255.240
S2(config-if)#no shut
S2(config-if)#exit
S2(config)#banner motd # This is my S2 Switch #
S2(config)#exit
S2#copy run start
Destination filename [startup-config]?[enter]
Building configuration...
[OK]
S2#
```

Şimdi, S2'den S1'i pingleyebilmemiz gerekir. Gelin bunu deneyelim:

```
S2#ping 192.168.10.17

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.10.17, timeout is 2
seconds:
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 1/1/1 ms
S2#
```

Size, iki sorum var: onu yapılandırmadığım halde, core switch'i nasıl pingleyebiliyorum ve neden beş yerine sadece dört ping gitti? (ilk nokta [.], bir zaman-aşımı'dır, ünlem işareti[!], başarı demektir)

İkisi de güzel sorudur. Sebebi şudur: İlk olarak, switch'i çalışır hale getirmeye ihtiyacınız yoktur. Tüm port'lar, varsayılan olarak etkindir. Bu nedenle, switch'i açarak, host'lar arasındaki iletişimi sağlayabilirsiniz. İkincisi, ARP'ın, IP adresini, MAC adresine çözümlemek için geçen zamandan dolayı, ilk ping çalışmaz.

### Core

Core switch konfigürasyonu şöyledir:

```
Switch>en
Switch#config t
Switch(config)#hostname Core
Core(config)#enable secret todd
Core(config)#int f0/5
Core(config-if)#description 1st Connection to S2
Core(config-if)#int fa0/5
Core(config-if)#description 2nd Connection to S2
Core(config-if)#int f0/7
Core(config-if)#desc 1st Connection to S1
Core(config-if)#int f0/8
Core(config-if)#desc 2nd Connection to S1
Core(config-if)#line con 0
Core(config-line)#password console
Core(config-line)#login
Core(config-line)#line vty 0 15
Core(config-line)#password telnet
Core(config-line)#login
Core(config-line)#int vlan 1
Core(config-if)#ip address 192.168.10.19 255.255.255.240
Core(config-if)#no shut
Core(config-if)#exit
Core(config)#banner motd # This is the Core Switch #
Core(config)#exit
Core#copy run start
Destination filename [startup-config]?[enter]
Building configuration...
[OK]
Core#
```

Şimdi, core switch'ten, S1 ve S2'yi pingleyelim ve ne olduğuna bakalım:

```
Core#ping 192.168.10.17
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.10.17, timeout is 2
seconds:
```

```
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 1/1/1 ms
Core#ping 192.168.10.18
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.10.18, timeout is 2
seconds:
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 1/1/1 ms
Core#sh ip arp
Protocol  Address          Age(min)  Hardware Addr   Type   Interface
Internet  192.168.10.18          0   001a.e2ce.ff40  ARPA   Vlan1
Internet  192.168.10.19          -   000d.29bd.4b80  ARPA   Vlan1
Internet  192.168.10.17          0   001b.2b55.7540  ARPA   Vlan1
Core#
```

Switch konfigürasyonlarını doğrulamadan önce bir router'ımız olmadığından, mevcut ağımızda ihtiyacımız olmamasına rağmen, bilmeniz gereken bir komut daha var. O, `ip default-gateway` komutudur. Şayet switch'inizi, LAN'ınız dışından yönetmek isterseniz switch'lerinizde, bir host makinesinde yaptığınız gibi bir varsayılan ağ geçidi ayarlamanız gerekir. Bunu, global moddan yaparsınız. Aşağıda, subnet aralığımızdaki son IP adresi kullanarak yapılandırdığımız router'ımızla ilgili örneği bulabilirsiniz (router'ı, bir sonraki, VLAN'larla ilgili bölümde kullanacağız):

```
Core#config t
Enter configuration commands, one per line.  End with CNTL/Z.
Core(config)#ip default-gateway 192.168.10.30
Core(config)#exit
Core#
```

Şimdi, basitçe yapılandırılmış üç switch'imiz var. Gelin, onlarla biraz eğlenelim.

## Port Güvenliği

Bu modülde daha önce söylediğim gibi switch'lerinizin, birileri tarafından kablo takılıp, onunla uğraşabilmesi, genelde iyi bir şey değildir. Yani, wireless güvenliği istediniz, neden switch güvenliği istemeyesiniz?

Port security kullanarak, statik bir MAC adresi ayarlayıp, politikalarınızı ihlal eden kullanıcılara yaptırımlar uygulayarak, bir port'a dinamik olarak atanabilen MAC adresi sayısını sınırlayabilirsiniz. Kişisel olarak, güvenlik politikası ihlal edildiğinde ve port'larını tekrar etkinleştirmeden önce, suistimal edenlerin, patronlarından bana, neden güvenlik politikasını ihlal ettiklerini açıklayan bir mesaj getirdiklerinde, port'un kapanması hoşuma gider. Bu genellikle, onlara davranışlarını hatırlatır.

Güvenli bir switch port'u, 1'den 8,192'e kadar, herhangi bir sayıda MAC adresiyle ilişkilendirilebilir. Fakat 50 serileri, sadece 192 adedi destekler (bu benim için yeterli gözüküyor). Switch'in, bu değerli otomatik öğrenmesini sağlayabilir ya da `switchport port-security mac-address mac-address` komutunu kullanarak her port için statik bir adres ayarlayabilirsiniz.

Şimdi, S1 switch'imizde, port güvenliği oluşturalım. Fa0/3 ve fa0/4 port'ları, lab'ımıza bağlı sadece bir cihaza sahiptir. Port security kullanarak, fa0/2 port'una bir host ve fa0/3 port'una bir telefon bağlanınca, başka bir cihazın bağlanamayacağını kesin olarak bilebiliriz. Bunu şöyle yaparız:

```
S1#config t
Enter configuration commands, one per line.  End with CNTL/Z.
S1(config)#int range fa0/3 - 4
S1(config-if-range)#switchport port-security maximum ?
  <1-8192>  Maximum addresses
S1(config-if-range)#switchport port-security maximum 1
S1(config-if-range)#switchport port-security mac-address sticky
S1(config-if-range)#switchport port-security violation ?
  protect   Security violation protect mode
  restrict  Security violation restrict mode
  shutdown  Security violation shutdown mode
S1(config-if-range)#switchport port-security violation shutdown
S1(config-if-range)#exit
```

Yukarıdaki komut, maksimum bir MAC adresine izin verilmesi için, fa0/3 ve fa0/4 port'unda port güvenliğini ayarlar ve sadece port'la ilgili ilk MAC adresi, frame'leri switch boyunca gönderecektir. Farklı MAC adrese sahip ikinci bir cihaz, switch'e bir frame göndermeyi denerse, `violation` komutumuzdan dolayı, port kapanacaktır. Her cihazın MAC adresini elle yazmak konusunda tembel olduğumdan, `sticky` komutunu kullandım.

Port'u kapatmak yerine kullanabileceğiniz iki mod daha vardır. Protect mode, diğer bir host'un bağlanabileceği, fakat frame'lerin atılacağı anlamına gelir. Restrict mode, bir port'ta ihlal olduğu konusunda, SNMP yardımıyla sizi uyarır. O zaman, ihlal eden kişiyi arayabilir ve onlara tutuklandıklarını (onları görebildiğinizi, ne yaptıklarını bildiğinizi ve başlarının büyük belada olduğunu) söyleyebilirsiniz!

Switch'ler arasındaki bağlantılarımızda, yedek linklere sahibiz, bu nedenle bu linklerde STP çalışmasına izin vermek en iyisidir. R1 ve R2 switch'lerimizde, fa0/3 ve fa0/4 port'larına (Core değil) bağlı host'larımızda vardır. Gelin, bu port'larda STP'yi kapatalım.

*Telefonların, tipik olarak Ethernet fişine sahip olmasından dolayı, host'ların, telefonların arkasına, fiziksel olarak direkt bağlanabileceğini bilin. Bu nedenle, her cihaz için switch'te sadece bir porta ihtiyacımız vardır. Buna, Modül9'daki telephony bölümünde gireceğim.*

NOT

## PortFast

Eğer switch'lerimizde `portfast` komutunu kullanırsak, host'larımızın bir DHCP adresi alabilmesi ihtimali problemini engelleriz. Çünkü STP, converge olmak için çok zaman harcar ve host'ların DHCP istek zamanını aşar. Bu nedenle, hem S1 hem de S2 switch'lerindeki fa0/3 ve fa0/4 portlarında PortFast'i kullanacağım:

```
S1#config t
S1(config)#int range f0/3-4
S1(config-if-range)#spanning-tree portfast ?
  disable  Disable portfast for this interface
  trunk    Enable portfast on the interface even in trunk mode
  <cr>
S1(config-if-range)#spanning-tree portfast
%Warning: portfast should only be enabled on ports connected to a
   single host. Connecting hubs, concentrators, switches, bridges,
   etc... to this interface when portfast is enabled, can cause
   temporary bridging loops.
 Use with CAUTION
```

```
%Portfast has been configured on FastEthernet0/2 but will only
 have effect when the interface is in a non-trunking mode.
S1(config-if-range)#
```

S1'i yapılandırdım, sonraki çıktıyı size göstermeyeceğim, fakat fa0/3-4'de de `PortFast`'i etkinleştirmek için, S2'ye gideceğim. PortFast kullandığınızda, dikkatli olmanızı tekrar hatırlatırım. Bir network kısır döngüsü oluşturmayı kesinlikle istemezsiniz! Neden? Şayet buna izin verirseniz, network hala çalışabilse de (bir çeşit), veri, çok yavaş gidecektir ve daha kötüsü, problemin kaynağını bulmanız için çok zaman harcamanıza neden olacaktır. Bu nedenle, dikkatli ilerleyin.

PortFast ayarlandığında birinin yanlışlıkla döngüye sebep olduğu durumda, kullanabileceğiniz bazı koruyucu komutları bilmeniz iyi olabilir:

## BPDUGuard

Bundan, biraz önce bahsettim: şayet bir port'unda, PortFast'i açarsanız, BPDUGuard'ı açmak, gerçekten iyi bir fikirdir. Şayet, PortFast etkinleştirilmiş bir switch port'u, bu port'tan bir BPDU alırsa, port'u, error disabled durumuna getirecektir. Bu, bir yöneticinin, yanlışlıkla başka switch ya da hub port'unu, PortFast yapılandırılmış bir switch port'una bağlamasını engeller. Aslında, siz, bunun olmasını ve ağınızın çökmesine neden olmasını ya da en azından ciddi bir şekilde hasar görmesini engelliyorsunuz. Bu komutu sadece, kullanıcıların direkt bağlı olduğu, Access katmanı switch'lerinizde yapılandırabilirsiniz. Bu nedenle, bunu, Core switch'imizde yapılandırmayacağız.

## BPDUFilter

PortFast'le kullanılması faydalı diğer komut, `BPDUFilter`'dır. PortFast'in etkin olduğu bir switch port'u, varsayılan olarak hala BPDU'ları alacağından, BPDU'ların bu port'a gelmesini ya da bu port'tan gitmesini tamamen durdurmak için BPDUFilter'ı kullanabilirsiniz. BPDUFilter filtrelemesi, bir BPDU alınırsa, port'u hemen PortFast durumundan çıkarır ve tekrar STP topoloji'sinin bir parçası olmasına zorlar. Port'u error disabled durumuna getiren BPDUGuard'ın tersine BPDUFilter, bir port'un PortFast olmadan çalışmasını devam ettirir. PortFast'le yapılandırılmış bir interface'den BPDU'lar almaya ihtiyaç yoktur. Tamamen açık olmam gerekirse, PortFast etkinleştirildiğinde, BPDUGuard ve BPDUFilter'ın varsayılan olarak neden etkin olmadığı konusunda hiçbir fikrim yok.

Gelin, zaten PortFast'le yapılandırılmış S1 ve S2 interface'lerimizi, BPDUGuard ve BPDUFilter ile yapılandıralım:

```
S1(config-if-range)#spanning-tree bpduguard ?
  disable  Disable BPDU guard for this interface
  enable   Enable BPDU guard for this interface
S1(config-if-range)#spanning-tree bpduguard enable
S1(config-if-range)#spanning-tree bpdufilter ?
  disable  Disable BPDU filtering for this interface
  enable   Enable BPDU filtering for this interface
S1(config-if-range)#spanning-tree bpdufilter enable

S2(config-if-range)#spanning-tree bpduguard enable
S2(config-if-range)#spanning-tree bpdufilter enable
```

`Bpduguard` ve `bpdufilter`'ın aynı şeyi başarmasından dolayı, genelde birini ya da diğerini kullanacağınızı bilin. Her iki komutu birlikte kullanmak düşmana karşı gereğinden fazla silah kullanmak olacaktır. Ayrıca, STP yapılandırıldığında, kullanabileceğiniz birkaç tane daha STP 802.1d eklentisi yapılandıracağız.

## UplinkFast

Aşağıda, Access katmanı switch'lerinizde (S1 ve S2) UplinkFast'i nasıl yapılandıracağınız vardır:

```
S1#config t
S1(config)#spanning-tree uplinkfast

S2#config t
S2(config)#spanning-tree uplinkfast
S1(config)#do show spanning-tree uplinkfast
UplinkFast is enabled

Station update rate set to 150 packets/sec.

UplinkFast statistics
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Number of transitions via uplinkFast (all VLANs)            : 1
Number of proxy multicast addresses transmitted (all VLANs) : 8

Name                 Interface List
- - - - - - -        - - - - - - - - - - - - - - - - - - - -
VLAN0001             Fa0/1(fwd), Fa0/2
S1(config)#
```

`Uplinkfast` komutu global bir komuttur ve her port'ta etkinleştirilmiştir.

## BackboneFast

Aşağıda, BackboneFast'i nasıl yapılandıracağınızı bulabilirsiniz:

```
S1(config)#spanning-tree backbonefast
S2(config)#spanning-tree backbonefast
Core(config)#spanning-tree backbonefast
S2(config)#do show spanning-tree backbonefast
BackboneFast is enabled

BackboneFast statistics
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Number of transition via backboneFast (all VLANs)          : 0
Number of inferior BPDUs received (all VLANs)              : 2
Number of RLQ request PDUs received (all VLANs)            : 0
Number of RLQ response PDUs received (all VLANs)           : 1
Number of RLQ request PDUs sent (all VLANs)                : 1
Number of RLQ response PDUs sent (all VLANs)               : 0
S2(config)#
```

UplinkFast'le yaptığımın tersine, BackboneFast'i, sadece Access katmanı switch'lerde değil de ağdaki tüm switch'te yapılandırdığıma dikkat edin. Lokal switch'teki link arızalarını, hem tespit edip hem de hızlıca düzeltmek için kullanılan UplinkFast'in tersine, BackboneFast'in, uzak bir switch'teki link arızalarını belirlemek için kullanıldığını hatırlayın.

## RSTP (802.1w)

RSTP'yi yapılandırmak, diğer 802.1d eklentilerinin yapılandırılması kadar kolaydır. Onun, 802.1d'den ne kadar iyi olduğunu hesap ederek, konfigürasyonun daha karmaşık olduğunu düşünebilirsiniz. Fakat biz şanslıyız, bizimki karmaşık değil. Gelin onu Core switch'imizde açalım ve ne olduğuna bakalım:

```
Core#config t
Core(config)#spanning-tree mode ?
  mst         Multiple spanning tree mode
  pvst        Per-Vlan spanning tree mode
  rapid-pvst  Per-Vlan rapid spanning tree mode
Core(config)#spanning-tree mode rapid-pvst
Core(config)#
1d02h: %LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan1,
 changed state to down
1d02h: %LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan1,
 changed state to up
```

Güzel! Core switch, şimdi 802.1w STP çalışıyor. Bunu doğrulayalım:

```
Core(config)#do show spanning-tree
VLAN0001
  Spanning tree enabled protocol rstp
  Root ID     Priority    32769
              Address     000d.29bd.4b80
              This bridge is the root
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID   Priority    32769  (priority 32768 sys-id-ext 1)
              Address     000d.29bd.4b80
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
              Aging Time 300

Interface        Role   Sts   Cost   Prio.Nbr   Type
---------        -----  ----  ----   --------   ------------
Fa0/5            Desg   FWD    19     128.5     P2p Peer(STP)
Fa0/6            Desg   FWD    19     128.6     P2p Peer(STP)
Fa0/7            Desg   FWD    19     128.7     P2p Peer(STP)
Fa0/8            Desg   FWD    19     128.8     P2p Peer(STP)
```

İlginç… Gerçekte hiçbir şey olmamış gibi görünüyor. Tüm port'ları converge olmuş, diğer iki switch'imi görebiliyorum. Her şey up olunca, her şey aynı gözüktü. 802.1d ve 802.1w, birlikte sorunsuz bir şekilde çalışabilir görünmektedir.

Fakat daha yakından baktığımızda, 802.1w switch'in, 802.1d çalışan diğer switch'lere bağlı port'lardaki 802.1w BPDU'ları, 802.1d BPDU'lara çevirdiğini görebilirsiniz.

S1 ve S2 switch'leri, Core switch'in, onları tekrar 802.1d BPDU'lara döndürdüğünden, Core switch'in aslında 802.1d çalıştığına inanırlar. Ve S1 ve S2 switch'leri, 802.1w BPDU'ları almasına rağmen, ondan anlamazlar ve onları sadece atarlar. Bununla beraber, hangi portun 802.1d çalıştığını bilen Core, 802.1d BPDU'larını alır ve onları, S1 ile S2'den kabul eder. Başka bir deyişle, 802.1w'un tek bir switch'te açılması, ağımızın tamamına faydası olmayacaktır.

Sinir bozucu ufak bir sorun, Core switch'in, 802.1d BPDU'larını, S1 ve S2'ye bağlı port'lardan göndereceğini anlayınca, S1 ve S2 sonradan 802.1w ile yapılandırılmadıysa, bunu otomatik olarak değiştirmeyeceğidir. 802.1d BPDU'larını durdurmak için, hala Core switch'i reboot etmemiz gerekecektir.

## EtherChannel

EtherChannel'ı yapılandırmanın en kolay yolu, Cisco Network Assistant iledir ve bunu, bölümün sonunda göstereceğim. Şimdi, CLI komutlarını da bilmeniz gerektiğinden, CLI ile yapacağım. EtherChannel'ın, Cisco ve IEEE olmak üzere iki versiyonu olduğunu hatırlayın. Ben Cisco versiyon'unu kullanacağım ve S1 ve Core switch arasındaki linkleri birleştireceğim.

`Interface port-channel` global komutunu ve S1 ve Core switch'lerde, `channel-group` ve `channel-protocol interface` komutlarını kullanacağım. Neye benzediği, aşağıdaki gibidir:

```
S1#config t
S1(config)#int port-channel 1
S1(config-if)#int range f0/1-2
S1(config-if-range)#switchport mode trunk
1dO3h: %SPANTREE_FAST-7-PORT_FWD_UPLINK: VLANOOO1 FastEthernetO/2
 moved to Forwarding (UplinkFast).
S1(config-if-range)#switchport nonegotiate
S1(config-if-range)#channel-group 1 mode desirable
S1(config-if-range)#do sh int fa0/1 etherchannel
Port state    = Up Sngl-port-Bndl Mstr Not-in-Bndl
Channel group = 1      Mode = Desirable-Sl    Gcchange = O
Port-channel  = null  GC = OxOOO1OOO1   Pseudo port-channel = Po1
Port index    = O     Load = OxOO       Protocol =   PAgP
[output cut]

Core#config t
Core(config)#int port-channel 1
Core(config-if)#int range f0/7-8
Core(config-if-range)#switchport trunk encap dot1q
Core(config-if-range)#switchport mode trunk
1dO3h: %SPANTREE_FAST-7-PORT_FWD_UPLINK: VLANOOO1 FastEthernetO/2
 moved to Forwarding (UplinkFast).
Core(config-if-range)#switchport nonegotiate
Core(config-if-range)#channel-group 1 mode desirable
1dO4h: %SPANTREE_FAST-7-PORT_FWD_UPLINK: VLANOOO1 FastEthernetO/2
 moved to Forwarding (UplinkFast).
1dO4h: %SPANTREE_FAST-7-PORT_FWD_UPLINK: VLANOOO1 FastEthernetO/2
 moved to Forwarding (UplinkFast).
```

```
1d04h: %LINK-3-UPDOWN: Interface Port-channel1, changed state to up
1d04h: %LINEPROTO-5-UPDOWN: Line protocol on Interface
Port-channel1, changed state to up
Core(config-if-range)#do show int port-channel 1
Port-channel1 is up, line protocol is up (connected)
  Hardware is EtherChannel, address is 001b.2b55.7501 (bia
001b.2b55.7501)
  MTU 1500 bytes, BW 200000 Kbit, DLY 100 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  Full-duplex, 100Mb/s, link type is auto, media type is unknown
[output cut]
```

Switch'lerin, link tiplerini otomatik algılamasını ve ayrıca, trunking'i otomatik kurmasını durdurmak için `switchport nonegotiate` interface komutunu ekledim. Onun yerine, trunk linklerimi, statik olarak yapılandırdım. S1 ve Core arasındaki iki link, Cisco EtherChannel PAgP versiyonu kullanarak, birleştirildi.

Tamam, fakat bekleyin, sonraki bölümde Virtual LAN'lar hakkında bilgi almadan önce, switch konfigürasyonumuzu doğrulamamız ve root bridge'imizle oynamamız gerekir.

## Cisco Catalyst Switch'lerin Doğrulanması

Bir router ya da switch ile yapmaktan hoşlandığım ilk şey, `show running-config` komutu ile konfigürasyonları gözden geçirmektir. Niçin? Çünkü bunu yapmak bana her cihazla ilgili detaylı bilgileri verir. Bununla beraber, oldukça vakit alır ve size tüm konfigürasyonu göstermek, bu kitaptaki tüm sayfaları kaplar. Bunun yerine, bize hala güzel bilgiler verecek, diğer komutları kullanabiliriz.

Örnek olarak, bir switch'te ayarlanmış IP adresini doğrulamak için `show interface` komutunu kullanabiliriz. İşte çıktısı:

```
S1#sh int vlan 1
Vlan1 is up, line protocol is up
  Hardware is EtherSVI, address is 001b.2b55.7540 (bia
001b.2b55.7540)
  Internet address is 192.168.10.17/28
  MTU 1500 bytes, BW 1000000 Kbit, DLY 10 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set, reliability 255/255,
txload 1/255, rxload 1/255
  [output cut]
```

NOT *IP adreslerinin bir switch'te gerekmediğini hatırlayın. Bir IP adresi, mask ve varsayılan ağ geçidi ayarlamamızın tek sebebi, yönetim amaçlıdır.*

### *show mac address table*

Bu komutun, bu bölümün başlarında gösterildiğini hatırladığınıza eminim. Onu kullanmak, content addressable memory (CAM) tablosu olarak da belirtilen forward filter tablosunu görüntüler. S1 switch'inden çıktı aşağıdaki gibidir:

```
S1#sh mac address-table
          Mac Address Table
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Vlan    Mac Address       Type        Ports
----    --------------    ---------   -------------
 All    0100.0ccc.cccc    STATIC      CPU
 All    ffff.ffff.ffff    STATIC      CPU
[output cut]
   1    0002.1762.b235    DYNAMIC     Po1
   1    0009.b79f.c080    DYNAMIC     Po1
   1    000d.29bd.4b87    DYNAMIC     Po1
   1    000d.29bd.4b88    DYNAMIC     Po1
   1    0016.4662.52b4    DYNAMIC     Fa0/4
   1    0016.4677.5eab    DYNAMIC     Po1
   1    001a.2f52.49d8    DYNAMIC     Po1
   1    001a.2fe7.4170    DYNAMIC     Fa0/8
   1    001a.e2ce.ff40    DYNAMIC     Po1
   1    0050.0f02.642a    DYNAMIC     Fa0/3
Total Mac Addresses for this criterion: 31
S1#
```

Switch'ler, CPU'ya atanan temel MAC adreslerini kullanır ve 2960'lar, 20'yi kullanır. Yukarıdaki çıktıdan, EtherChannel port 1'e dinamik olarak atanmış, beş MAC adresimiz olduğunu görebilirsiniz. Fa0/3, Fa0/8 ve Fa0/4 port'ları, atanmış tek MAC adresine sahiptir ve tüm port'lar, VLAN 1'e atanmıştır.

S2 switch CAM'ine bakalım ve ne bulabileceğimize bakalım. S2 switch'inin, S1'i gibi yapılandırılmış EtherChannel'a sahip olmadığını aklınızda tutun. Bu nedenle STP, Core switch'e yedek linklerden bir tanesini kapatacaktır:

```
S2#sh mac address-table
          Mac Address Table
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Vlan    Mac Address       Type        Ports
----    --------------    --------    -------
 All    0008.205a.85c0    STATIC      CPU
 All    0100.0ccc.cccc    STATIC      CPU
 All    0100.0ccc.cccd    STATIC      CPU
 All    0100.0cdd.dddd    STATIC      CPU
[output cut]
   1    0002.1762.b235    DYNAMIC     Fa0/3
   1    000d.29bd.4b80    DYNAMIC     Fa0/1
   1    000d.29bd.4b85    DYNAMIC     Fa0/1
   1    0016.4662.52b4    DYNAMIC     Fa0/1
   1    0016.4677.5eab    DYNAMIC     Fa0/4
   1    001b.2b55.7540    DYNAMIC     Fa0/1
```

```
Total Mac Addresses for this criterion: 26
S2#
```

Yukarıdaki çıktıdan, Fa0/1'e atanmış dört MAC adresimiz olduğunu görebiliriz. Ve tabi ki, port3 ve 4'deki her host'a bir bağlantımız olduğunu da görebiliriz. Fakat port2 nerededir? Port2, yedek link olduğundan, STP, Fa0/2'yi, blocking moda koymuştur. Bundan, birazdan tekrar bahsedeceğim.

## Statik MAC Adresleri Atanması

MAC adres tablosunda, statik bir MAC adresi ayarlayabilirsiniz, fakat statik MAC port security gibi ayarlar ve tonlarca çalışma gerektirir. Bunu yapmak istediğinizde, yapılışı şöyledir:

```
S1#config t
S1(config)#mac-address-table static aaaa.bbbb.cccc vlan 1 int fa0/5
S1(config)#do show mac address-table
          Mac Address Table
- - - - - - - - - - - - - - - - - - - - - - - - - - - -
Vlan    Mac Address       Type        Ports
-----   --------------    --------    -------
 All    0100.0ccc.cccc    STATIC      CPU
[output cut]
   1    0002.1762.b235    DYNAMIC     Po1
   1    0009.b79f.c080    DYNAMIC     Po1
   1    000d.29bd.4b87    DYNAMIC     Po1
   1    000d.29bd.4b88    DYNAMIC     Po1
   1    0016.4662.52b4    DYNAMIC     Fa0/4
   1    0016.4677.5eab    DYNAMIC     Po1
   1    001a.2f52.49d8    DYNAMIC     Po1
   1    001a.2fe7.4170    DYNAMIC     Fa0/8
   1    001a.e2ce.ff40    DYNAMIC     Po1
   1    0050.0f02.642a    DYNAMIC     Fa0/3
   1    aaaa.bbbb.cccc    STATIC      Fa0/5
Total Mac Addresses for this criterion: 31
S1(config)#
```

Bir MAC adresinin, Fa0/5'e kalıcı olarak atandığını ve onun, aynı zamanda sadece VLAN 1'e atandığını görebilirsiniz.

## *show spanning-tree*

`Show spanning-tree` komutunu bilmeniz önemlidir. Onunla, kimin root bridge olduğunu ve her VLAN için hangi priority'lerin ayarlandığını görebilirsiniz.

Cisco switch'lerin, basit olarak, her VLAN'in, kendi STP protokol örneğini çalıştırdığı anlamına gelen, Per-VLAN Spanning Tree (PVST) çalıştırdığını bilin. Şayet `show spanning-tree` yazarsak, VLAN 1'den başlayarak her VLAN için bilgi alırız. Birçok VLAN'e sahipsek ve VLAN 2'de ne olduğunu görmek istiyorsak, `show spanning-tree vlan 2` komutunu kullanırız.

Aşağıda, S1 switch'inden, `show spanning-tree` komutu çıktısını bulabilirsiniz. Sadece VLAN 1'i kullandığımız için, komuta, VLAN numarasını eklememize ihtiyacımız yok:

```
S1#sh spanning-tree
VLAN0001
Spanning tree enabled protocol ieee
Root ID     Priority    32769
            Address     000d.29bd.4b80
            Cost        3012
            Port        56 (Port-channel1)
            Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

Bridge ID   Priority    49153  (priority 49152 sys-id-ext 1)
            Address     001b.2b55.7500
           Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
           Aging Time 15
Uplinkfast enabled

Interface    Role   Sts  Cost      Prio.Nbr    Type
---------    ----   ---  ----      --------    ---------
Fa0/3        Desg   FWD  3100      128.3       Edge Shr
Fa0/4        Desg   FWD  3019      128.4       Edge P2p
Fa0/8        Desg   FWD  3019      128.8       P2p
Po1          Root   FWD  3012      128.56      P2p
```

Sadece yapılandırılmış VLAN 1'imiz olduğundan, bu komut için, daha fazla çıktı olmayacaktır. Şayet olsaydı, switch'te yapılandırılmış her VLAN için başka çıktılar olacaktı. Varsayılan priority, 32768'dir, fakat system ID extension (sys-id-ext) denilen VLAN tanımlayıcı vardır. Bridge ID priority'si, bu VLAN'ın numarası ile artar. Ve sadece VLAN 1'e sahip olduğumuzdan, bir artırarak, 32769'a ulaşırız. Fakat varsayılan olarak, BackboneFast'in, bu bridge'in root olmasını engellemek için varsayılan priority'i 49152'ye artırdığını bilin.

Çıktının yukarısı, bize root bridge'in kim olduğunu gösterir:

```
VLAN0001
  Root ID  Priority    32769
           Address     000d.29bd.4b80
           Cost        3012
           Port        56 (Port-channel1)
           Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
```

EtherChannel Port 1, root bridge'e seçilen yolumuz anlamına gelen root port'umuzdur ve 000d.29bd.4b80 adresine sahiptir. Bu sadece Core switch ya da S2 olabilir ve kısa sürede hangisi olduğunu anlayacağız.

Komuttaki son çıktı, STP çalışan ve diğer cihaza bağlantısı olan port'ları gösterir. EtherChannel çalıştırdığımızdan, bloklanmış port'larımız yoktur. Bridge'inizin root olup olmadığını anlamanın tek yolu, Altn BLK port'ları (alternatif, bloklanmış port'lar anlamında) olup olmadığını anlamak için bakmaktır. Bir root bridge, herhangi bir interface'inde, bloklanmış port'a sahip olmayacaktır, fakat S1'deki tüm port'larımız, EtherChannel konfigürasyonumuzdan dolayı, forwarding (FWD) gösterir.

## Root Bridge'imizi Belirlemek

Root bridge'imizi belirlemek için `show spanning-tree` komutunu kullanırız. Gelin, diğer iki switch'imize bakalım ve varsayılan root bridge'in hangi switch olduğunu anlayalım. Hem Bridge ID MAC adresi hem de S1 switch'inin priority'sini not alın. S2 çıktısı şöyledir:

```
S2#sh spanning-tree

VLAN0001
  Spanning tree enabled protocol ieee
  Root ID  Priority    32769
           Address     000d.29bd.4b80
           Cost        3019
           Port        2 (FastEthernet0/1)
           Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID  Priority    49153  (priority 49152 sys-id-ext 1)
             Address     001a.e2ce.ff00
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
             Aging Time 300
  Uplinkfast enabled

  Interface    Role   Sts   Cost      Prio.Nbr    Type
  ---------    ----   ----  ----      --------    ----
  Fa0/1        Root   FWD   3019      128.2       P2p
  Fa0/2        Altn   BLK   3019      128.3       P2p
  Fa0/3        Desg   FWD   3100      128.4       Edge Shr
  Fa0/4        Desg   FWD   3019      128.5       Edge P2p
  S2#
```

Fa0/2'nin bloklanmış olduğunu görebiliriz, bu nedenle bizim root bridge'imiz olamaz. Root bridge, bloklanmış port'lara sahip olamaz. Bridge MAC ID adresine ve priority'e dikkat edin. Core switch çıktısı şöyledir:

```
Core#sh spanning-tree
VLAN0001
  Spanning tree enabled protocol rstp
  Root ID    Priority    32769
             Address     000d.29bd.4b80
             This bridge is the root
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID  Priority    32769  (priority 32768 sys-id-ext 1)
             Address     000d.29bd.4b80
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
             Aging Time 300
```

```
Interface    Role    Sts   Cost    Prio.Nbr  Type
---------    ------  ----  -----   --------  -------------
Fa0/5        Desg    FWD   19      128.5      P2p Peer(STP)
Fa0/6        Desg    FWD   19      128.6      P2p Peer(STP)
Po1          Desg    FWD   12      128.66     P2p Peer(STP)
```

İşte buldunuz, Bu bridge root'tur.

Fakat şunu düşünün, Core switch, 32768 varsayılan priority'e sahipken, diğer switch'ler neden 49152'dir? Çünkü o, STP 802.1w versiyonu çalışıyor ve BackBoneFast, varsayılan olarak pasiftir.

Şimdi, her switch'in bridge MAC adresine bakalım:

- *S1 address:* 001b.2b55.7500
- *S2 address:* 001a.e2ce.ff00
- *Core address:* 000d.29bd.4b80

MAC adresini kontrol ederek ve tüm switch'ler, varsayılan priority'e sahipse, hangi switch'in root bridge olacağını düşünürsünüz? MAC adresini, soldan başlayarak, sağa doğru okumaya başlayın. Core, açıkça en düşük MAC adresidir ve `show spanning-tree` komutuna bakarak, onun gerçekten bizim root bridge'imiz olduğunu görebiliriz (tüm switch'ler aynı priority'e sahip olsa da). MAC adreslerini karşılaştırarak root bridge'i anlamak güzel bir pratiktir.

## Root Bridge'imizi Ayarlamak

Varsayılan olarak Core'un bizim root bridge'imiz olması uygundur. Fakat sırf eğlence için onu değiştirelim. Bunu şöyle yaparız:

```
S1#config t
S1(config)#spanning-tree vlan 1 priority ?
  <0-61440>  bridge priority in increments of 4096
S1(config)#spanning-tree vlan 1 priority 16384
S1(config)#do show spanning-tree
VLAN0001
Spanning tree enabled protocol ieee
 Root ID  Priority    16385
          Address     001b.2b55.7500
          This bridge is the root
          Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

 Bridge ID  Priority    16385  (priority 16384 sys-id-ext 1)
            Address     001b.2b55.7500
            Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
           Aging Time 300
Interface    Role    Sts  Cost    Prio.Nbr      Type
---------    -----   ---- ----    --------     ----------
Fa0/3        Desg    FWD   100     128.3        Edge Shr
Fa0/4        Desg    FWD   19      128.4        Edge P2p
Fa0/8        Desg    FWD   19      128.8        P2p
Po1          Desg    FWD   12      128.56       P2p
```

S1 priority'sini, 16384'e düşürdüğünüzde S1 switch'i, hemen root bridge olacaktır. Priority'lerinizi, 0'da 61440'a kadar ayarlayabilirsiniz. Sıfırın (0) anlamı, switch'in daima root olacağı, 61440'ın anlamı, switch'in asla root olmayacağıdır.

Şayet tüm bu root bridge konfigürasyonlarını ve doğrulamayı geçmek isterseniz, size bahsetmek istediğim son bir komut var. Şayet, Cisco sınavlarını geçmek istiyorsanız, bunların hiçbirini atlamayın. Root bridge olarak ayarlamak için bir switch'te çalıştırabileceğiniz basit komut aşağıdadır:

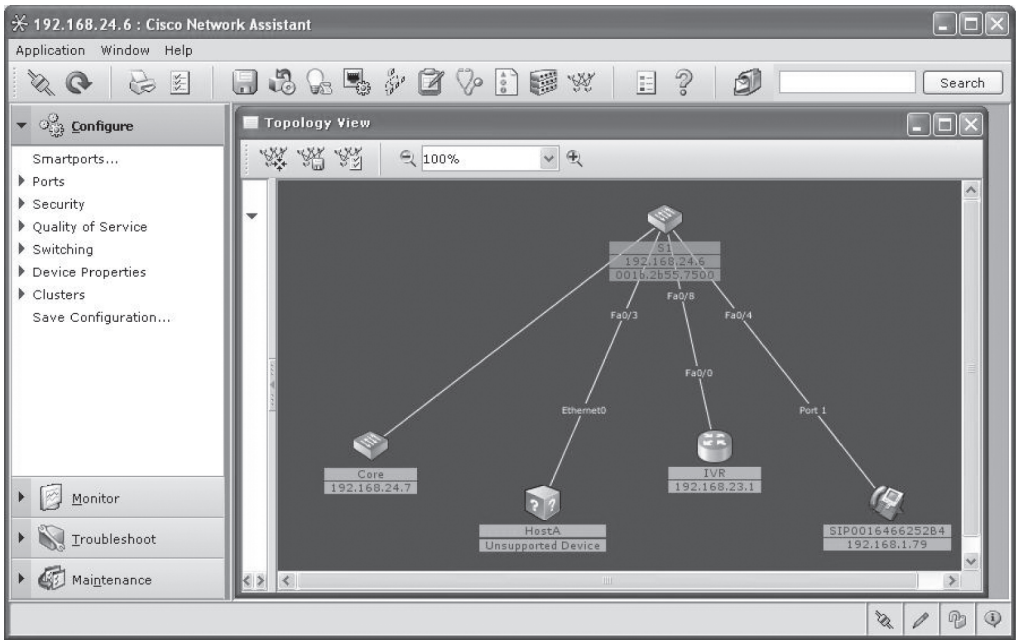
```
S1(config)#spanning-tree vlan 1 root ?
  primary    Configure this switch as primary root for this
spanning tree
  secondary  Configure switch as secondary root
S1(config)#spanning-tree vlan 1 root primary
```

Bunun düşük priority'li switch'i iptal etmeyeceğini bilin. Bu komut sadece tüm switch'leriniz aynı priority'e ayarlandıysa çalışır. Bunu, VLAN başına ayarlamanız gerektiğini ve birincil ile ikincil switch'leri, root olarak ayarlayabileceğinizi de belirttim mi? Evet, yapabilirsiniz ve bunu yapmak, bu modülde yaptığımızdan tamamıyla daha kolaydır. Fakat bu CCNA sınavına hazırlanmak için ilk ve gelişmiş bir kılavuzdur. Zor yöntemle olduğu halde onun nasıl yapıldığını bildiğinizden emin olun.

# Cisco Network Assistant

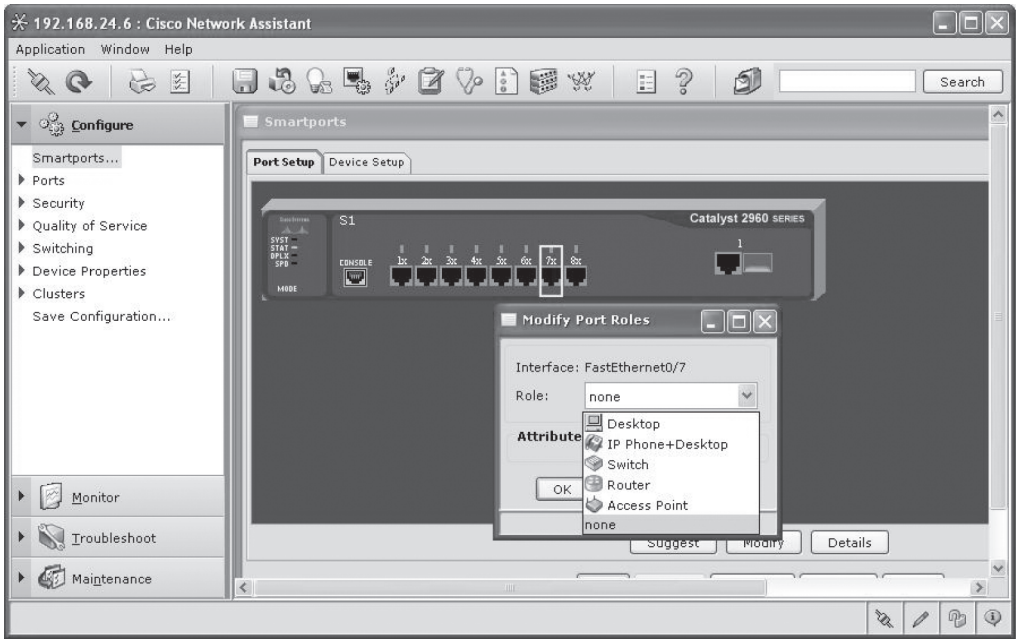
Cisco Network Assistant (CNA), SDM'le olduğu gibi, switch'lerinizin konfigürasyonunu kolayca yapmanızı sağlayabilir. Bu hem iyi hem de kötüdür. Daha zor konfigürasyonları daha kolay yapmamızı sağladığından dolayı iyidir ve herkesin kolayca yapmasını sağladığından kötüdür. Fakat başlangıçta hala akıllıca olabilir, bu nedenle onu indirin ve CNA'ye mümkün olduğu kadar aşina olun.

CNA kullanarak bir switch'e bağlanınca şöyle bir ekran gelir.




Bağlı tüm cihazlarımızın, güzel bir topolojik görünüşüne sahibiz. Benim IP telefonumu, Core switch'i ve 2811 router'ımın hostname'i olan inter-VLAN router (IVR)'ı görebiliyoruz. Fakat bu çıktıda sadece direkt bağlı cihazları göreceğimizi bilmeniz gerekmektedir. Yani Core'un diğer tarafındaki cihazları göremeyiz.

CNA ekranının sol tarafındaki en faydalı menü seçeneklerinden birisi, muhtemelen Smartports'tur. Smartports'a tıkladığınızda, switch'in dış görünüşü gelir.




Tüm yapmanız gereken, bir port'u ya da hepsini belirlemek ve sonra, sağ-tuş yapıp port ya da port grubunun bağlanacağı cihaz tipini seçmektir. Güzel özellik! Bununla beraber Smartports'un ilgi çeken durumu, onu tamamen yapılandırmasıdır. Örneğin S1 switch'imin 6.port'unu, bir desktop'a (PC'ye) bağlamak için yapılandırmak amacıyla Smartports'u kullandım. Switch'imde çalıştırmış olduğum komutlar aşağıdaki gibidir:

```
!
interface FastEthernet0/6
 switchport mode access
 switchport port-security
 switchport port-security aging time 2
 switchport port-security violation restrict
 switchport port-security aging type inactivity
 macro description cisco-desktop
 spanning-tree portfast
 spanning-tree bpduguard enable
!
```

Bunu kontrol edin. Switch'imi yapılandırmak için, bu modülde tartıştığım özelliklerin hiçbirini anlamaya dahi ihtiyacım olmadı. Onun yerine, sadece interface'e bağlı cihaz tipini seçtim ve switch, bu port tipi için akıllı seçimler olduğu düşünülen komutlarla, otomatik olarak yapılandırıldı.

CNA ile devam etmeden önce, interface altında yer alan konfigürasyondan biraz bahsedelim. Bu bölümde, port-security, spanning-tree portfast ve bpguard hakkında zaten bahsettik. Fakat ayrıca ayarlanmış, `macro` ve `port-security aging` komutlarına ne dersiniz? Macro'lar, bir Smartport ayarlandığında çalışan, yeni Cisco switch'lerde bulunan, varsayılan programlardır. Port-security aging komutu, port'un yaşlanma değerini ayarlar. Aging komutuyla kullanabileceğiniz iki seçenek vardır: absolute ve inactivity. Absolute seçeneği, 0 ile 1440 dakika arasında belirlenen bir zamandan sonra, bir port'taki gizli adresleri silecektir. Inactivity seçeneğinin anlamı, port'taki

adreslerin, belirli bir süre, etkin olmaması durumunda, silinmesidir. Bu süre de, 0 ile 1440 dakika arasında ayarlanmaktadır. Bu nedenle, `switchport port-security aging time 2` komutunu, `switchport port-security aging type inactivity` komutu ile kullandığınızda, port'la ilgili tüm MAC adresleri, iki dakika sonra silinecektir. Ve sadece bu değil, `switchport port-security violation restrict` komutunun anlamı, bir trap'in, SNMP sunucusuna ya da Network Management Staition'a (NMS) gönderileceğidir. Bu biraz, gereğinden fazla görünse de, bana göre hala güzel bir komuttur.

Bir interface'deki port güvenliğine şu komutla göz atabilirsiniz:

```
S1#sh port-security interface f0/6
Port Security              : Enabled
Port Status                : Secure-down
Violation Mode             : Restrict
Aging Time                 : 2 mins
Aging Type                 : Inactivity
SecureStatic Address Aging : Disabled
Maximum MAC Addresses      : 1
Total MAC Addresses        : 0
Configured MAC Addresses   : 0
Sticky MAC Addresses       : 0
Last Source Address:Vlan   : 0000.0000.0000:0
Security Violation Count   : 0
```

Burada, `macro` komutuna ne oldu? Endişelenmeyin. O sadece macro'ları oluşturmanızı ve çalıştırmanızı sağlayan bir switch komutudur. Ve Cisco'nun yeni switch'leri, önceden yapılandırılmış altı tane macro'ya sahiptir. Garip bir şekilde, onları çalışan konfigürasyonda göremezsiniz. Onları sadece, show parser komutuyla görebilirsiniz. F0/6 port'unda çalışan macro aşağıdaki gibidir:

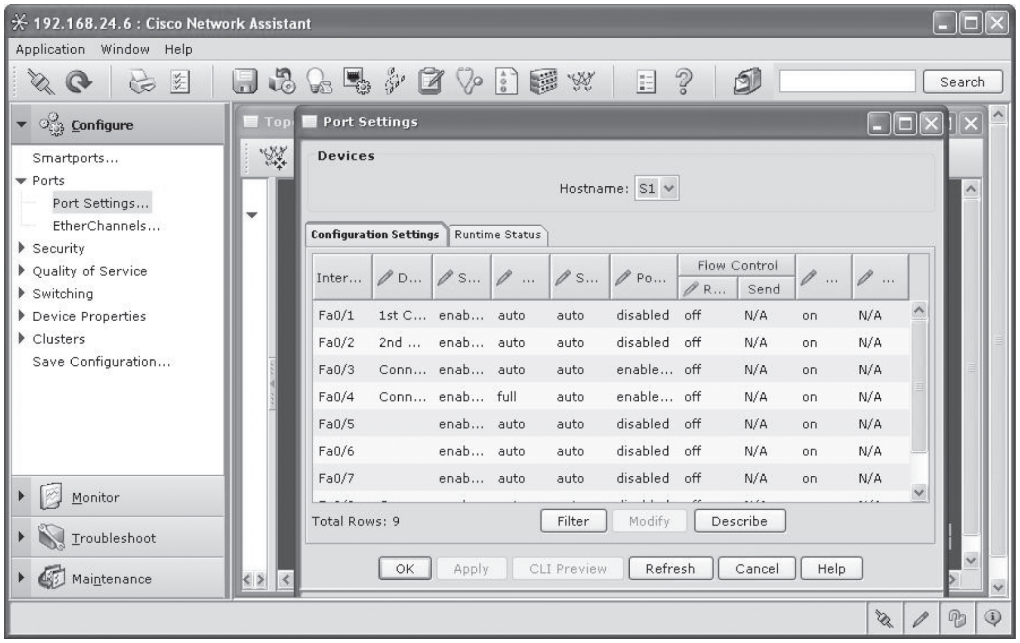
```
S1#sh parser macro
Total number of macros = 6
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Macro name : cisco-desktop
Macro type : default interface
# macro keywords $access_vlan
# Basic interface - Enable data VLAN only
# Recommended value for access vlan should not be 1
switchport access vlan $access_vlan
switchport mode access

# Enable port security limiting port to a single
# MAC address — that of desktop
switchport port-security
switchport port-security maximum 1

# Ensure port-security age is greater than one minute
# and use inactivity timer
switchport port-security violation restrict
```

```
switchport port-security aging time 2
switchport port-security aging type inactivity

# Configure port as an edge network port
spanning-tree portfast
spanning-tree bpduguard enable
[output cut]
```

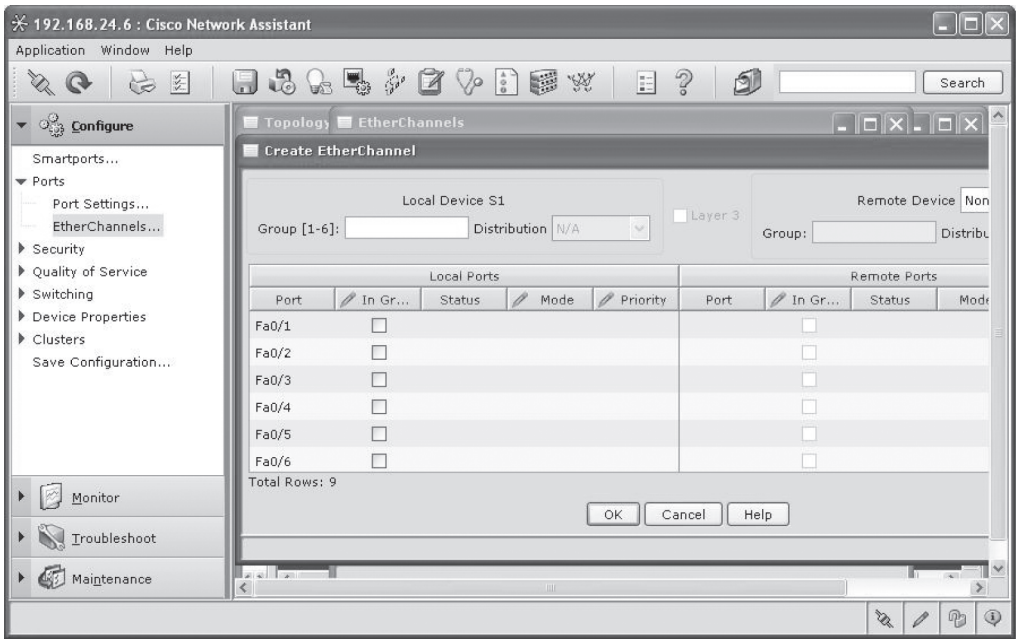
Tamam her şey güzel. En azından, bu komutların, switch'imizde nasıl yapılandırıldığını biliyoruz. Smartports'un altındaki tüm cihazlar için macro vardır. Show `parser macro brief` komutuyla gösterilmektedir:

```
S1#sh parser macro brief
     default global   : cisco-global
     default interface: cisco-desktop
     default interface: cisco-phone
     default interface: cisco-switch
     default interface: cisco-router
     default interface: cisco-wireless
```

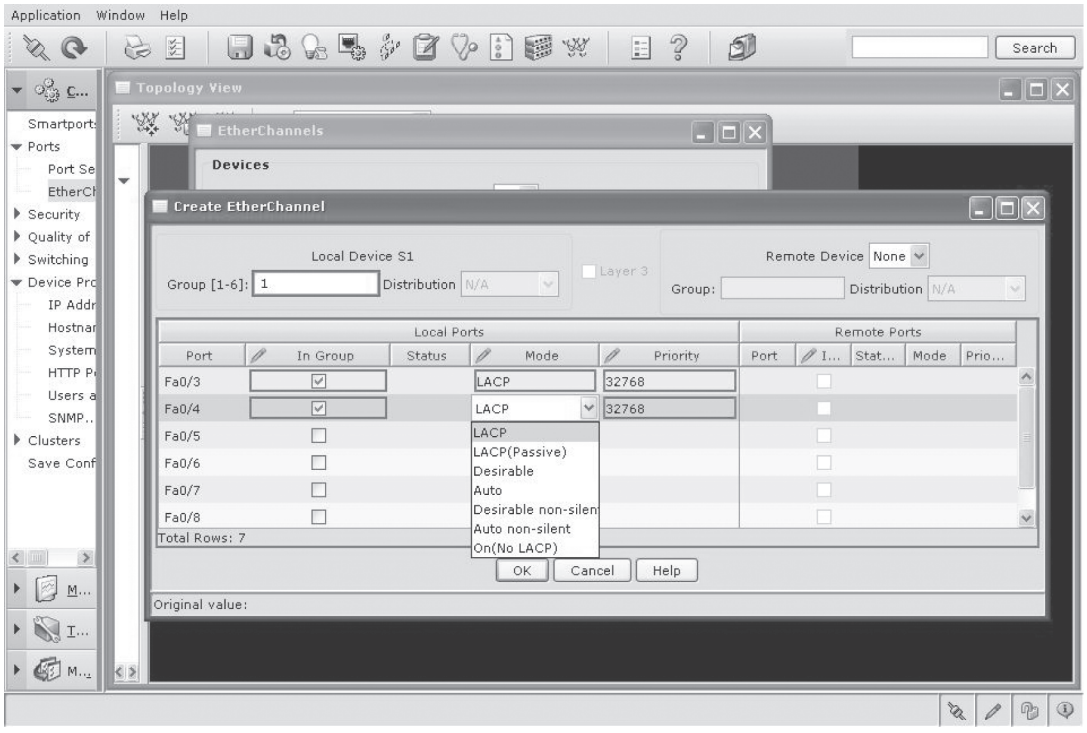
Herneyse, CNA'ye dönelim, Ports altında, port Settings'i bulacaksınız.



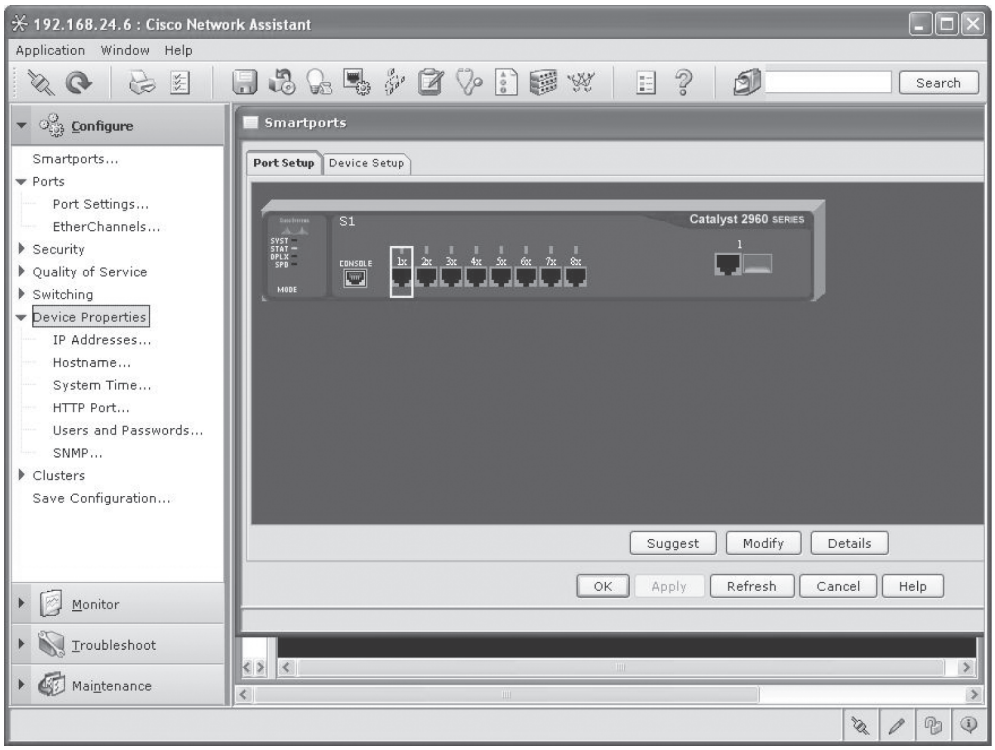

Burası, hem bakıp hem de değiştirebileceğiniz, basit konfigürasyon bilgileriyle tüm port'ları bulacağınız yerdir. Ports'ların altında EtherChannels'ı da bulacaksınız.



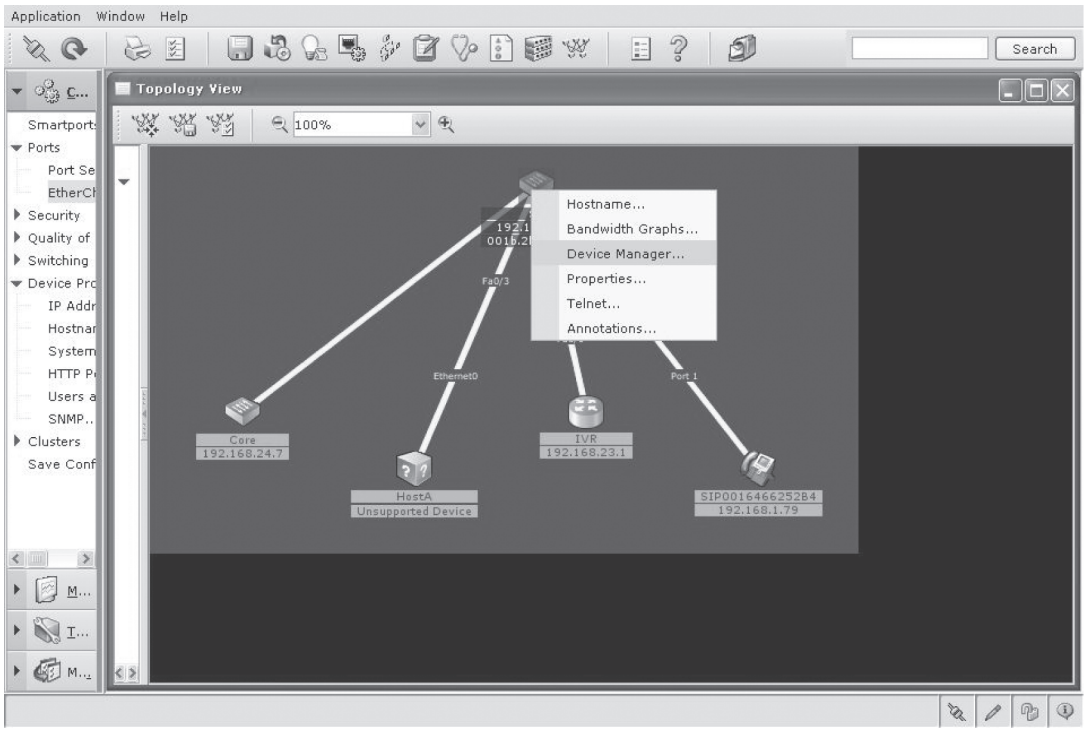

Bu, bazı port'ları birleştirmek için CLI kullanmaktan daha kolay bir yol olabilir. EtherChannel'ı tıkladıktan sonra grubunu seçin ve birleştirmek istediğiniz port'ları ve kullanacağınız protokolleri işaretleyin. Hepsi bu!



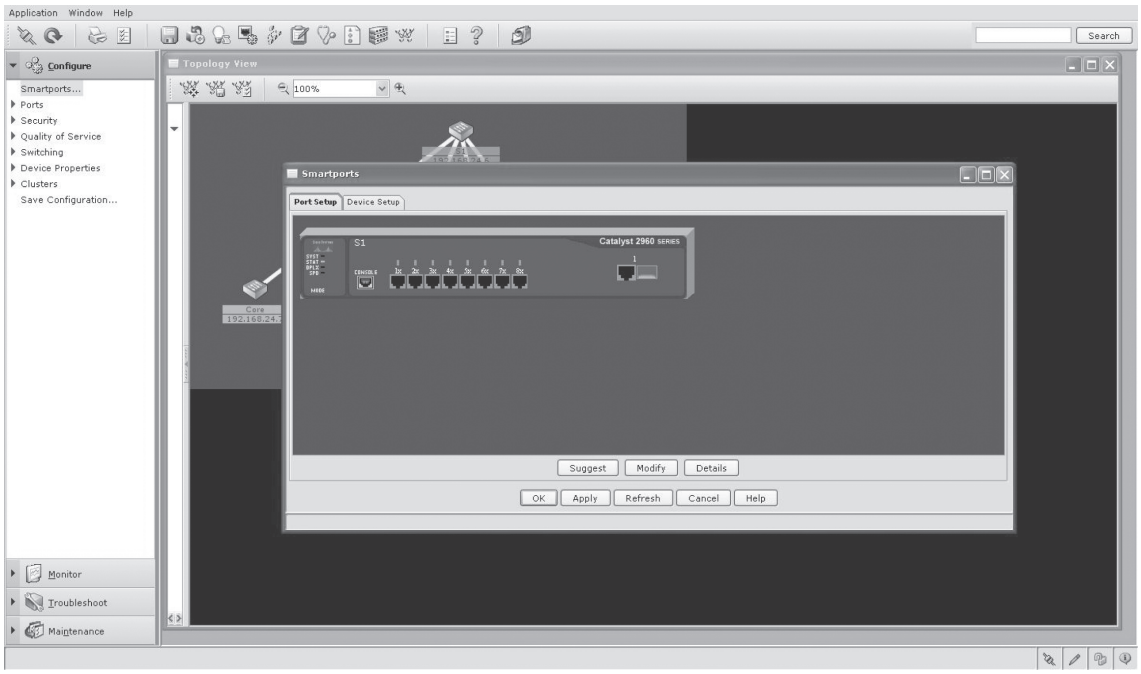

Bu, çok güzel! Device Properties'e tıklayarak, buradan switch'inizin birçok özelliğini ayarlayabilirsiniz: IP adresleri, hostname, kullanıcı adı ve şifreler ve daha da fazlası.




Topology View ekranından bir cihaza sağ-tuş yapabilir ve birçok seçeneği tıklayabilirsiniz. Fakat Device Manager ilginçtir.




Device Manager'i seçtiğinizde varsayılan HTTP browser'ınız açılacaktır ve switch'inizi, bir HTTP browser'dan yapılandırmaya ve doğrulamaya başlama zamanıdır. Bununla beraber CNA ile olandan daha yavaş bir cevap süresine sahiptir, bu nedenle kişisel olarak onu kullanmam.




CNA'de daha da fazlası vardır ve onu indirmek ve kurcalamak için mümkün olduğu kadar zaman ayırmanızı tavsiye ederim.

Şimdi, şunu kabul etmelisiniz ki büyük bir bölüm olduğu halde, bu bölümde gerçekten çok şey öğrendiniz, belki biraz eğlenmiş bile olabilirsiniz. Şimdi, tüm switch'leri yapılandırdınız ve doğruladınız, port security'i ayarladınız ve hem STP eklentilerini incelediniz hem de root bridge'inizi ayarladınız. Yani Virtual LAN'larla ilgili herşeyi öğrenmeye hazırsınız! Tüm switch konfigürasyonlarımızı kaydedeceğim, böylece "Virtual LAN'lar (VLAN'lar)" başlıklı bölüm 9'a buradan başlayacağım.

# Özet

Bu bölümde, bridge ve switch'ler arasındaki farklılıklardan, her ikisinin de katman2'de nasıl çalıştığından ve bir frame'i iletmek ya da yayınlama kararını vermesi için, bir MAC adres forward/filter tablosu oluşturmaktan bahsettim.

Ayrıca bridge'ler (switch'ler) arasında çoklu linklere sahip olduğunuzda oluşabilecek problemleri ve Spanning Tree Protokolünü (STP) kullanarak, bu problemlerin nasıl çözüleceğini tartıştım.

Aynı zamanda, konfigürasyonları doğrulamayı, STP eklentilerini ayarlamayı ve bir bridge priority ayarlayarak, root bridge'i değiştirmeyi içeren, Cisco Catalyst switch'lerin konfigürasyonlarını detaylı olarak işledim.

Son olarak switch konfigürasyonlarınızda, size büyük yardımı olacak, Cisco Network Assistant'ı gözden geçirdim.

# Sınav Gereklilikleri

**Üç switch fonksiyonunu hatırlamak:** Adres öğrenme, forward/filter kararları ve kısır döngüden kaçınma, bir switch'in fonksiyonlarıdır.

**show mac address-table komutunu hatırlamak:** `show mac address-table` komutu, LAN switch'inde kullanılan, forward/filter tablosunu gösterecektir.

**Bir switch LAN'ındaki Spanning Tree Protokolünün asıl amacını anlamak:** STP'nin asıl amacı, yedek switch yollarına sahip bir ağdaki switching döngülerini engellemektir.

**STP durumlarını hatırlamak:** Blocking durumunun amacı, döngüdeki yolların kullanımını engellemektir. Listening durumundaki bir port, MAC adres tablosuna yerleştirmeden, veri frame'lerini iletmek için hazırdır. Learning durumundaki bir port, MAC adres tablosu oluşturur, fakat veri frame'lerini iletmez. Forwarding durumundaki bir port, bridge port'undaki tüm veri frame'lerini alır ve gönderir. Son olarak, disabled durumundaki bir port, nerdeyse işlevsel değildir.

**show spanning-tree komutunu hatırlamak:** `show spanning-tree` komutuna ve kimin root bridge olduğunun nasıl belirleneceğine aşina olmalısınız.

# Yazılı Lab 8

Aşağıdaki soruların cevaplarını yazın:

1. Hangi komut size forward/filter tablosunu gösterecektir?
2. MAC adresi forward/filter tablosunda değilse, switch, bu frame'i ne yapacaktır?
3. Katman2'deki üç switch fonksiyonu nedir?
4. Switch port'undan bir frame alınırsa ve kaynak MAC adresi, forward/filter tablosunda değilse, switch ne yapacaktır?
5. Switching döngüsünü engellemek için katman2'de ne kullanılır?
6. 802.1w, aynı zamanda ne olarak belirtilmektedir?
7. STP'nin ne zaman converge olması düşünülür?
8. Switch'ler, _____ domain'lerini ayırır.
9. Yedek switch yollarına sahip bir ağda, switching döngülerini engellemek için ne kullanılır?
10. Hangi Cisco 802.1d eklentisi, BPDU'ların bir port'tan aktarılmasını durdurur?

*(Yazılı lab8'nin cevapları, bu bölüm için gözden geçirme sorularına cevaptan sonra bulunabilir.)*

# Gözden Geçirme Soruları

NOT *Aşağıdaki sorular, bu bölümün materyallerini anladığınızı test etmek için tasarlanmıştır. İlave sorulara nasıl sahip olacağınızla ilgili daha fazla bilgi için, lütfen bu kitabın Giriş bölümüne bakın.*

1. Aşağıdakilerden hangisi, döngü olmayan bir network sağlamak için kullanılan bir katman2 protokolüdür?
   A. VTP
   B. STP
   C. RIP
   D. CDP
2. Hangi komut, forward/filter tablosunu görüntüler?
   A. `show mac filter`
   B. `show run`
   C. `show mac address-table`
   D. `show mac filter-table`
3. Bir ağı, bir bridge (switch) ile segment'lerine ayırmanın sonucu nedir? (İki şık seçin.)
   A. Collision domain sayısını artırır.
   B. Collision domain sayısını azaltır.
   C. Broadcast domain domain sayısını artırır.
   D. Broadcast domain domain sayısını azaltır.
   E. Collision domain'leri küçültür.
   F. Collision domain'leri büyültür.
4. Converge olan bir spanning-tree ağını hangi ifade açıklar?
   A. Tüm switch ve bridge port'ları forwarding durumdadır.
   B. Tüm switch ve bridge port'ları, root ya da designated port olarak atanmışlardır.
   C. Tüm switch ve bridge port'ları, ya forwarding ya da blocking durumdadır.
   D. Tüm switch ve bridge port'ları, ya forwarding ya da looping durumdadır.
5. Bir switch LAN'ında Spanning Tree Protokolünün amacı nedir?
   A. Switch ortamında, ağı görüntülemek için bir mekanizma sağlamak.
   B. Yedek yollara sahip ağlarda, switching döngülerini engellemek.
   C. Yedek switch yollarına sahip ağlarda, switching döngülerini engellemek.
   D. Birçok switch boyunca, VLAN veritabanını yönetmek.
   E. Collision domain'leri oluşturmak.
6. Ağdaki kullanılabilir bantgenişliğini artıran, katman2'deki üç fonksiyon nedir? (Üç şık seçin.)
   A. Adres öğrenme
   B. Routing
   C. Forwarding ve filtering
   D. Network döngüleri yaratma

E. Loop'tan kaçınma

F. IP adresleme

7. Switch'inizin, yeşil ve amber arasında değişen LED durumu olan bir port'u vardır. Bunun sebebi ne olabilir?

   A. Port'ta problemler vardır.

   B. Port kapalıdır.

   C. Port, STP blocking moddadır.

   D. Hiçbir şey, herşey normaldir.

8. Aşağıdaki ifadelerden hangisi doğrudur?

   A. Bir switch, tek bir collision domain ve tek bir broadcast domain oluşturur. Bir router, tek bir collision domain'i oluşturur.

   B. Bir switch, ayrı collision domain'leri ve ayrı broadcast domain'leri oluşturur. Bir router, ayrı bir broadcast domain'i sağlar.

   C. Bir switch, tek bir collision domain'i ve ayrı bir broadcast domain'i oluşturur. Bir router da ayrı bir broadcast domain'i sağlar..

   D. Bir switch ayrı bir collision domain ve ayrı bir broadcast domain oluşturur. Bir router, ayrı collision domain'leri sağlar.

9. Uzak olarak yönetebilmek için, bir Catalyst switch yapılandırmanız gerekmektedir. Aşağıdakilerden hangisini, bunu gerçekleştirmek için kullanırsınız?

   A.
```
Switch(configs)#int fa0/1
Switch(configs-if)#ip address 192.168.10.252 255.255.255.0
Switch(configs-if)#no shut
```

   B.
```
Switch(configs)#int vlan 1
Switch(configs-if)#ip address 192.168.10.252 255.255.255.0
Switch(configs-if)#ip default-gateway 192.168.10.254
255.255.255.0
```

   C.
```
Switch(configs)#ip default-gateway 192.168.10.254
Switch(configs)#int vlan 1
Switch(configs-if)#ip address 192.168.10.252 255.255.255.0
Switch(configs-if)#no shut
```

   D.
```
Switch(configs)#ip default-network 192.168.10.254
Switch(configs)#int vlan 1
Switch(configs-if)#ip address 192.168.10.252 255.255.255.0
Switch(configs-if)#no shut
```

10. Bir interface'den bir frame alındığında ve hedef donanım adresi, bilinmediğinde veya filter tablosunda olmadığında switch ne yapar?

    A. İlk uygun link için, switch'e iletir.

    B. Frame'i atar.

    C. Cihazı bulması için, frame'i yayınlar.

    D. Yeni bir isim çözümlemesi için, ilk gönderen istasyona bir mesaj gönderir.

11. Şayet switch bir frame alırsa ve kaynak MAC adresi, MAC adres tablosunda değilse, fakat hedef adresi tabloda ise switch, frame'i ne yapacaktır?

    A. Onu atar ve ilk gönderen makinaya bir hata mesajı gönderir.

    B. Frame'i ağda yayınlar.

    C. MAC adres tablosuna, kaynak adresini ve port'unu ekler ve frame'i hedef port'una gönderir.

    D. MAC adres tablosuna hedefi ekler ve sonra frame'i iletir.

12. Switch'inizde, yeni 802.1w çalıştırmak istiyorsunuz. Aşağıdakilerden hangisi bu protokolü etkinleştirir?

    A. `Switch(config)#spanning-tree mode rapid-pvst`

    B. `Switch#spanning-tree mode rapid-pvst`

    C. `Switch(config)#spanning-tree mode 802.1w`

    D. `Switch#spanning-tree mode 802.1w`

13. Hangi durumda bir switch LAN'da aktarılan, aynı unicast frame'in çok sayıda kopyası olur?

    A. Yüksek trafik periyotları sırasında

    B. Arızalı linklerin tekrar kurulmasından sonra

    C. Üst-katman protokolleri, yüksek güvenlik gerektirdiğinde

    D. Hatalı kurulmuş yedek topoloji'lerde

14. Hangi komut, aşağıdaki çıktıyı üretmek için kullanılmıştır:

```
Vlan    Mac Address       Type       Ports
----    -------------   --------    -----
  1    0005.dccb.d74b    DYNAMIC     Fa0/1
  1    000a.f467.9e80    DYNAMIC     Fa0/3
  1    000a.f467.9e8b    DYNAMIC     Fa0/4
  1    000a.f467.9e8c    DYNAMIC     Fa0/3
  1    0010.7b7f.c2b0    DYNAMIC     Fa0/3
  1    0030.80dc.460b    DYNAMIC     Fa0/3
```

    A. `show vlan`

    B. `show ip route`

    C. `show mac address-table`

    D. `show mac address-filter`

15. Bir sunucuya bağlı port'ta STP'yi etkisiz kılmak isterseniz, hangi komutu kullanırsınız?

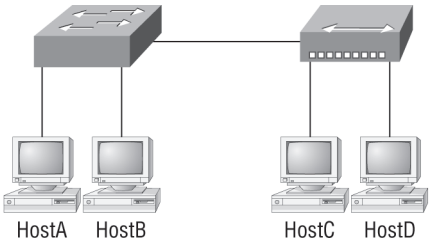
    A. `disable spanning-tree`

    B. `spanning-tree off`

    C. `spanning-tree security`

    D. `spanning-tree portfast`

16. Grafiğe referans alın. Switch, switch adres tablosundaki FastEthernet 0/1 port'una neden iki MAC adresi atanmıştır?




| MAC Address | Type | Ports |
|---|---|---|
| 0005.dccb.d74b | DYNAMIC | Fa0/1 |
| 000a.f467.9e80 | DYNAMIC | Fa0/1 |
| 000a.f467.9e8b | DYNAMIC | Fa0/4 |
| 000a.f467.9e8c | DYNAMIC | Fa0/3 |

A. HostC ve HostD'den gelen veri, FastEthernet 0/1 switch port'uyla alınmıştır..

B. Switch'e bağlı iki cihazdan gelen veri, HostD'ye iletilmiştir.

C. HostC ve HostD, yerleştirilmiş NIC'lerine sahiptir.

D. HostC ve HostD, farklı VLAN'lerdedir.

17. Katman2 switching, aşağıdakilerden hangisini sağlar? (Dört şık seçin.)

A. Donanım tabanlı bridging (ASIC)

B. Wire speed

C. Düşük latency

D. Düşük cost

E. Routing

F. WAN servisleri

18. **show mac address-table** yazdınız ve aşağıdaki çıktıyı aldınız:

```
Switch#sh mac address-table
Vlan    Mac Address       Type        Ports
-----   -------------   ---------   ----------
   1    0005.dccb.d74b   DYNAMIC     Fa0/1
   1    000a.f467.9e80   DYNAMIC     Fa0/3
   1    000a.f467.9e8b   DYNAMIC     Fa0/4
   1    000a.f467.9e8c   DYNAMIC     Fa0/3
   1    0010.7b7f.c2b0   DYNAMIC     Fa0/3
   1    0030.80dc.460b   DYNAMIC     Fa0/3
```

Yukardaki switch'in, aşağıdaki MAC adresleriyle bir frame aldığı düşünülmektedir:

- Kaynak MAC: 0005.dccb.d74b
- Hedef MAC: 000a.f467.9e8c

O ne yapacaktır?

A. Frame'i atacaktır.

B. Frame'i sadece, Fa0/3 port'undan atacaktır.

C. Onu sadece, Fa0/1 port'undan atacaktır.

D. Onu, Fa0/1 port2u hariç tüm port'lardan atacaktır.

19. Her switch interface'ine dinamik olarak, bir host'un eklenmesinin kabul edilmesini sağlamanız gerekmektedir. Bu politikayı uygulamak için, Catalyst switch'inizde hangi iki komutu yapılandırmalısınız? (İki şık seçin.)

    A. `Switch(config-if)#ip access-group 10`

    B. `Switch(config-if)#switchport port-security maximum 1`

    C. `Switch(config)#access-list 10 permit ip host 1`

    D. `Switch(config-if)#switchport port-security violation shutdown`

    E. `Switch(config)#mac-address-table secure`

20. Yedekleme için iki çapraz kabloyla birbirine bağlı iki switch'iniz var ve STP, pasif durumdadır. İki switch arasında, aşağıdakilerden hangisi olacaktır?

    A. Switch'lerdeki routing tabloları güncellenmeyecektir.

    B. Switch'lerdeki MAC forward/filter tablosu güncellenmeyecektir.

    C. Switch ağında, broadcast fırtınası olacaktır.

    D. Switch'ler, iki link arasında, otomatik olarak yük-dengelemesi yapacaktır.

# Gözden Geçirme Sorularının Cevapları

1. B Spanning Tree Protokolü, yedek yolları olan bir switch ağında, switching döngülerini durdurmak için kullanılmaktadır.
2. C `show mac address-table`, switch'teki forward/filter tablosunu gösterir.
3. A, E Bridge'ler, collision domain'lerini ayırır. Bu, bir ağdaki collision domain sayısını artıracak ve daha küçük collision domain'ler yapacaktır.
4. C Switch ya da bridge'lerin tüm port'ları, forwardin ya da blocking durumuna geçtiğinde convergence olur. Convergence tamamlanıncaya kadar, hiçbir veri iletilmez.Verinin tekrar iletilmesinden önce, tüm cihazların güncellenmesi gerekir.
5. E Spanning Tree Protokolü (STP), katman2 döngülerini durdurmak için tasarlanmıştır. Tüm Cisco switch'leri, varsayılan olarak, STP'ye sahiptir.
6. A,C,E Katman2 özellikleri adres öğrenme, ağın iletilmesi ve filtrelenmesi ile döngüden kaçınmayı içermektedir.
7. A Bir switch portuna bağlandığınızda, link ışıkları ilk olarak, turuncu/amber'dır ve sonra, normal operasyonu işaret eden yeşile dönerler. Şayet link ışığı yanıp sönüyorsa bir problem var demektir.
8. B Switch'ler collision domain'leri; router'lar, broadcast domain'lerini ayırır.
9. C Bir switch'i uzaktan yönetmek için varsayılan olarak interface vlan 1 olan, yönetim VLAN altında, bir IP adresi ayarlamalısınız. Sonra, global configuration moddan, `ip default-gateway` komutu ile varsayılan ağgeçidi ayarlamalısınız.
10. C Switch'ler tüm frame'leri, bilinmeyen bir hedef adresi ile yayınlar. Şayet bir cihaz, frame'e cevap verirse switch, cihazın lokasyonunu netice olarak vermek için MAC adres tablosu'nu güncelleyecektir.
11. C Kaynak MAC adresi, MAC adresi tablosunda olmadığından switch, kaynak adresini ve bağlı olduğu port'u, MAC adres tablosuna ekler ve sonra frame'i çıkış port'undan iletir.
12. A 802.1w, aynı zamanda Rapid Spanning Tree Protokolü olarak adlandırılmaktadır. Cisco switch'lerde, varsayılan olarak etkin değildir. Fakat Cisco eklentilerinin, 802.1d ile sağladığı tüm düzeltmelere sahip olduğundan çalışmasıSTP'den daha iyidir.
13. D Şayet Spanning Tree Protokolü switch'lerinizde çalışmıyorsa ve onları, yedek linklerle birbirine bağladıysanız, broadcast fırtınalarına ve birçok frame kopyalarına sahip olacaksınız.
14. C `show mac address-table` komutu, bir switch'teki, aynı zamanda MAC tablosu da denilen, forward/filter tablosunu görüntüler.
15. D Şayet, switch'inize bağlı bir sunucu ya da başka cihazlar varsa, STP, pasif olduğu halde, bir switching döngüsü oluşturmadığınıza tamamen eminseniz, bu port'larda portfast kullanabilirsiniz. Onu kullanmak, STP converge olurken, port'un gelmesi için olağan 50 saniye harcamayacağı anlamına gelir.
16. A Bir switch, birden çok MAC adresine sahip bir port'a sahip olabilir. Grafikte, iki host bağlı bir hub, Fa0/1 port'una bağlıdır.
17. A,B,C,D Bridge'lerin aksine, Switch'ler, donanım tabanlıdır. Cisco, switch'lerinin, wire speed olduğunu ve düşük latency sağladığını söyler. 1990'lardaki fiyatlarıyla karşılaştırıldığında, onların düşük maliyetli olduğunu düşünüyorum.
18. B Hedef MAC adresi, MAC adres tablosu'nda (forward/filter tablosu) olduğundan, onu sadece Fa0/3 port'undan gönderecektir.

19. B, D `switchport port-security`, önemli bir komuttur ve CNA ile süper kolaydır. Bununla beraber CLI'dan port için kabul edilen maksimum sayıda MAC adresini ayarlayabilirsiniz ve sonra maksimum sayı geçilirse, ceza uygulayabilirsiniz.
20. C Şayet bir switch'te spanning tree etkin değilse ve başka switch'e yedek linkleriniz varsa diğer olası problemler arasında, broadcast fırtınaları olacaktır.

# Yazılı Lab 8.1'in Cevapları

1. `Show mac address-table`
2. Frame'i, aldığı dışındaki tüm port'lardan gönderir.
3. Adres öğrenme, forwar/filter kararları ve döngüden kaçınma.
4. Kaynak MAC adresini, forward/filter tablosuna ekleyecektir ve frame'i aldığı port ile ilişkilendirecektir.
5. Spanning Tree Protokolü (STP)
6. Rapid Spanning Tree Protokolü (RSTP)
7. Tüm port'lar, blocking ya da forwarding modda olduğunda.
8. Collision
9. Spanning Tree Protokolü (STP)
10. PortFast

# 9 Virtual LAN'lar (VLAN'lar)

# 9 Virtual LAN'lar (VLAN'lar)

- VLAN Temelleri
- VLAN Üyelikleri
- VLAN'ları Tespit Etmek
- VLAN Trunking Protokolü (VTP)
- VLAN'lar Arası Routing
- VLAN'ları yapılandırmak
- VTP'yi Yapılandırmak
- Telephony: Voice VLAN'ları Yapılandırmak
- VLAN'ları ve VLAN'lar Arası Routing'i Yapılandırmak İçin CNA Kullanmak
- Özet
- Sınav Gereklilikleri
- Yazılı Lab 9
- Gözden Geçirme Soruları
- Gözden Geçirme Sorularının Cevapları
- Yazılı Lab 9'un Cevapları

# Virtual LAN'lar (VLAN'lar)

Bunu size sürekli söylediğimi biliyorum, fakat unutmadığınızdan emin olmalıyım. Son kez hatırlatıyorum ki; varsayılan olarak switch'ler collision domain'lerini , router'lar ise broadcast domain'lerini ayırır. Tamam, şimdi kendimi daha iyi hissediyorum! Artık devam edebiliriz.

Zayıf omurga temeline dayalı geçmişteki ağların tersine, bugünün network tasarımı, daha düz bir mimariyle tanımlanmaktadır (switch'lere teşekkürler). Sadece switch'lerden oluşan bir ağ topluluğunda, broadcast domain'lerini nasıl ayırırız? Virtual local area network (VLAN) oluşturarak. Bir VLAN, bir switch'te yönetimsel olarak tanımlı portlara bağlı kaynakların ve network kullanıcılarının mantıksal bir gruplandırmasıdır. VLAN'lar oluşturduğunuzda, switch'teki farklı portları, farklı alt network'lere atayarak, bir katman2 switch ağ topluluğunda daha küçük broadcast domain'leri oluşturma kabiliyetine sahip olursunuz. Her VLAN, kendi subnet'i ya da broadcast domain'inde gibi davranmaktadır. Yani ağa broadcast edilen frame'ler, sadece aynı VLAN'daki mantıksal gruplanmış portlar arasında anahtarlanırlar.

Bu, artık router'lara ihtiyacımız olmayacak mı demektir? Belki evet, belki de hayır. Bu tamamen ne istediğinize ve ihtiyaçlarınızın ne olduğuna bağlıdır. Varsayılan olarak, belirli bir VLAN'daki host'lar, başka bir VLAN'a üye olan host'larla haberleşemezler. Şayet VLAN'lar arası iletişim isterseniz, cevap, sizin hala bir router'a ihtiyacınız olduğudur.

Bu bölümde, bir VLAN'ın ne olduğunu ve VLAN üyeliklerinin, bir switch ağında nasıl kullanıldığını detaylı bir şekilde öğreneceksiniz. Ayrıca, VLAN Trunk Protocol'ün (VTP), VLAN bilgisiyle switch veritabanlarını güncellemek için nasıl kullanıldığını ve trunking'in, tüm VLAN bilgilerini tek bir link boyunca göndermek için nasıl kullanıldığını anlatacağım. Bir router'ı, switch ağıyla tanıştırarak oluşan VLAN'lar arası iletişimi nasıl yapabileceğinizi göstererek bitireceğim.

Tabi ki, switch ağımızı, VLAN'larla ve VLAN'lar arası routing ile yapılandıracağız. Switch'lerimizdeki VLAN'ları yapılandırmak için, Cisco Network Asistant (CNA) kullanarak bölümü tamamlayacağız.

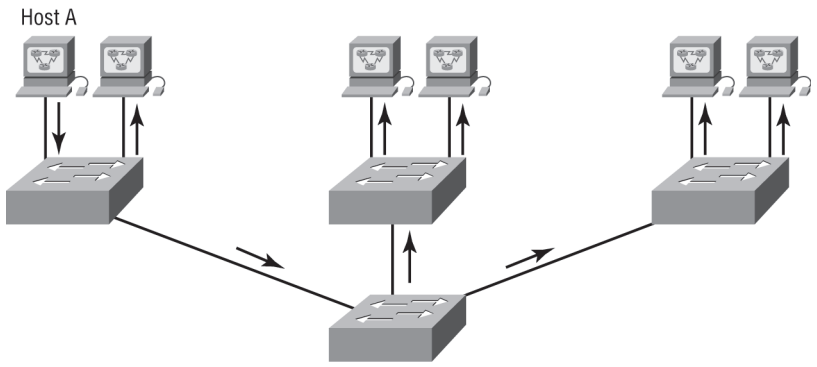
*Bu bölüm ile ilgili son güncellemeler için lütfen, www.lammle.com ve/veya www.sybex.com adreslerine bakınız.*

**NOT**

## VLAN Temelleri

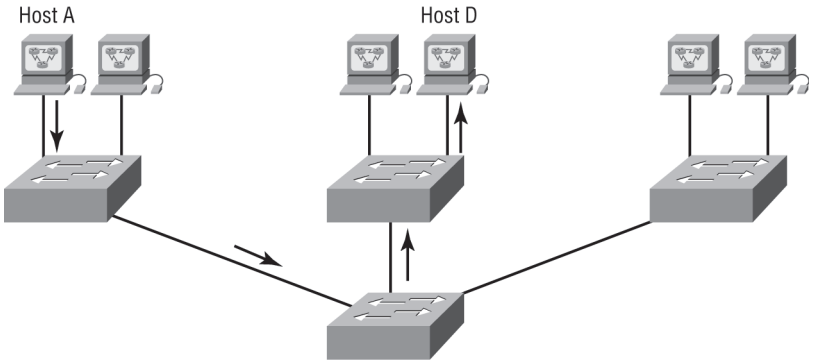
Şekil 9.1, katman2 switch network'lerinin, düz ağlar olarak genellikle nasıl tasarlandıklarını göstermektedir. Bu yapılandırma ile aktarılan her broadcast paketi, cihazın bu paketi almaya ihtiyacı olup olmadığına bakmaksızın, network'teki tüm cihazlar tarafından görülür.

Switch'ler, broadcast'leri tüm segment'lere iletirlerken, router'lar, varsayılan olarak broadcast'lerin sadece başlatılan ağda olmasına izin verir. Her neyse, onun düz network olarak isimlendirilmesinin sebebi, gerçek tasarımın fiziksel olarak düz olması değildir, onun tek broadcast domain olmasındandır. Şekil 9.1'de, Host A'nın, bir broadcast gönderdiğini ve orijinal olarak alınan port dışındaki tüm switch portlarının onu ilettiğini görüyoruz.




***Şekil 9.1:*** *Düz network yapısı.*

Şimdi, Şekil 9.2'yi inceleyin. Bir switch ağını resmeder ve Host A'nın, hedef olarak Host D'ye bir frame gönderdiğini göstermektedir. Görebileceğiniz gibi, önemli olan bu frame'in sadece Host D'nin bulunduğu porta iletilmesidir. Bu, varsayılan olarak tek collision domain'e sahip olmayı gerçekten istemeniz dışında (muhtemelen istemezsiniz!), eski hub network'leri karşısında büyük bir gelişmedir.




***Şekil 9.2:*** *Bir switch ağının avantajı.*

Şimdi, zaten bildiğiniz gibi, katman2 switch ağına sahip olarak elde edeceğiniz en büyük kar, onun switch'teki her porta bağlı cihazlar için ayrı collision domain segment'leri oluşturmasıdır. Bu senaryo bizi, Ethernet mesafe kısıtlamalarından korur. Böylece, daha geniş ağlar oluşturulabilir. Fakat çoğu kez, yeni gelişmeler yeni sorunlarla gelir. Örnek olarak, kullanıcı ve cihazlar ne kadar artarsa, her switch'in işlemesi gereken daha fazla broadcast ve paket olacaktır.

Başka bir sorun ise güvenliktir. Tipik bir katman2 ağ topluluğunda, tüm kullanıcıların, varsayılan olarak tüm cihazları görebilmesi gerçek bir sıkıntıdır. Cihazların, broadcast yapmasını durduramazsınız, artı kullanıcıların broadcast'lere cevap vermeye çalışmalarını engelleyemezsiniz. Yani, güvenlik seçenekleriniz, sunucu ve diğer cihazlarınıza şifre koymakla sınırlıdır.

Fakat durun, bir umut var! Bu, bir sanal LAN (VLAN) oluşturmanızdır. Katman2 switching ile ilgili birçok problemi, birazdan göreceğiniz gibi VLAN'larla çözebilirsiniz.

VLAN'ların network yönetimini kolaylaştırma yöntemlerinin kısa bir listesini aşağıda bulabilirsiniz:

- Network eklemek, taşımak ve değiştirmek, uygun VLAN'a bir port yapılandırarak kolayca yapılmaktadır.
- Genel olarak en yüksek güvenlik seviyesine ihtiyaç duyan bir kullanıcı grubu, kendilerine özel bir VLAN'a konabilir. Böylece bu VLAN dışındaki kullanıcılar, onlarla haberleşemezler.
- Kullanıcıların, işlevleriyle mantıksal gruplanması gibi, VLAN'lar, kullanıcıların fiziksel ve coğrafi lokasyonlarından bağımsız düşünülebilmektedir.
- VLAN'lar, network güvenliğini fazlasıyla artırır.
- VLAN'lar, boyutlarını küçültürken, broadcast domain sayısını arttırır.

Şimdi, switch özelliklerinden bahsedeceğim ve günümüz ağlarında, switch'lerin, hub'lardan daha iyi network servislerini nasıl sağladığını detaylı bir şekilde açıklayacağım.

## Broadcast Kontrolü

Broadcast'ler her protokol'de olurlar, fakat hangi sıklıkta olduğu, üç şeye bağlıdır:

- Protokol tipi
- Ağ topluluğundaki uygulama(lar)
- Bu servislerin nasıl kullanıldığı

Bazı eski uygulamalar, bant genişliği isteklerini azaltmak için tekrar yazılır. Fakat buldukça tüketecek, inanılmaz bant genişliği ihtiyacı olan yeni kuşak uygulamalar vardır. Bu bant genişliği oburları, hem broadcast hem de multicast kullanan multimedya uygulamalarıdır. Hatalı ekipmanlar, yetersiz segment'lere ayırma ve başarısız tasarlanmış firewall'lar, bu broadcast yoğunluklu uygulamaların oluşturulmasıyla ilgili problemlerin etkisini artırır. Bunların tamamı, network tasarımına yeni bir boyut ekler ve bir yönetici için yeni bir uğraşı yumağı sunar. Ağınızın düzgün bir şekilde subnet'lere ayrıldığından emin olarak, tek bir segment problemini çabuk bir şekilde izole edip, tüm ağ topluluğunuza dağıtılmasını engellemek zorunludur. Bunu yapmanın en etkili yöntemi, stratejik switching ve routing'dir.

Son zamanlarda, switch'lerin daha kolay satın alınabilir olmasından dolayı, birçok şirket, düz hub ağlarını tamamen switch'lerden oluşan ağlar ve VLAN ortamlarıyla değiştirmektedir. Bir VLAN'daki tüm cihazlar, aynı broadcast domain'inin üyesidir ve tüm broadcast'i alırlar. Varsayılan olarak, bu broadcast'ler, aynı VLAN üyesi olmayan bir switch'teki tüm portlarda filtrelenirler. Tüm kullanıcılarınızın aynı broadcast domain'inde olmasıyla karşılaşacağınız problemleri izole ettiğinden, bu çok iyi bir çözümdür.

## Güvenlik

Tamam, biliyorum. Daima anlaşılması gereken bir şeyler vardır, değil mi? Bu güvenlik problemlerine geri dönme zamanı geldi. Düz bir ağ topluluğunun güvenliğinin üstesinden, hub ve switch'lerin birbirine router'larla bağlanmasıyla gelinirdi. Basit olarak router'ların görevi, güvenliği sağlamaktı. Bu düzenleme, bazı nedenlerden dolayı oldukça etkisizdi. İlk olarak, fiziksel ağa bağlanan herhangi birisi, bu fiziksel LAN'da yer alan network kaynaklarına erişebiliyordu. İkincisi, bu ağda olan tüm trafiği izlemek isteyen birinin yapması gereken şey, basitçe bir network analizör'ünü hub'a bağlamaktı. Bu hoş olmayan durumun benzeri olarak, kullanıcıların, mevcut hub'a workstation'larını bağlayarak bir workgroup'a katılabilmeleriydi. Bu, bir ayı için üstü açık bir bal fıçısının güvenli olması gibi bir şeydir.

Fakat VLAN'ları bu kadar iyi yapan tamamıyla budur. Şayet onları oluşturur ve çoklu broadcast grupları oluşturursanız, her port ve kullanıcı için eksiksiz bir kontrole sahip olursunuz. Herhangi birinin, workstation'larını bir switch port'una taktığı ve network kaynaklarına erişim sağladığı günler geçmişte kalmıştır. Çünkü artık her port'u, artı port'ların erişebileceği tüm kaynakları kontrol edebilirsiniz. Daha da fazlası, yeni 2960/3560 switch'lerle, bunun otomatik olmasıdır.

Ve kontrol burada bitmez dostlarım. Çünkü VLAN'lar, belirli bir kullanıcının istediği network kaynaklarına uygun olarak oluşturulabilir, artı switch'ler, bir network yönetim istasyonunun, network kaynaklarına yetkisiz bir erişimi bildirmesi için yapılandırılabilirler. Şayet VLAN'lar arası haberleşmeye ihtiyacınız varsa, bunu yapmak için kullanacağınız bir router'da da kısıtlamalar yapabilirsiniz. Ayrıca donanım adresleri, protokoller ve uygulamalarda da kısıtlamalar yapabilirsiniz. Bu bahsettiğimiz güvenlik ile bal fıçısı, sert titanyum'dan yapıldı, kapaklandı ve dikenli telle çevrildi.

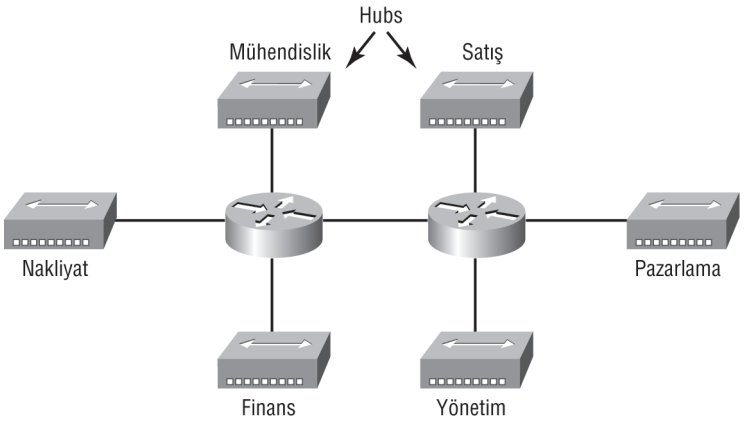
## Esneklik ve Ölçeklenebilirlik

Şayet şimdiye kadar okuduklarınıza dikkat ederseniz, katman2 switch'lerin, filtreleme için sadece frame'leri okuduğunu bilirsiniz. Onlar, Network katman protokol'üne bakmazlar. Ve switch'ler, varsayılan olarak tüm broadcast'leri iletirler. Fakat VLAN'lar oluşturur ve çalıştırırsanız, aslında, katman2'de daha küçük broadcast domain'ler oluşturursunuz.

Bunun anlamı, bir VLAN'daki bir düğümden gönderilen broadcast'lerin, farklı bir VLAN'a ait port'lara iletilmeyeceğidir. Böylece switch port'larını ve kullanıcıları, bir switch'teki VLAN gruplarına veya switch'lere bağlı gruba atayarak, fiziksel lokasyonlarına bakılmaksızın, olmasını istediğiniz broadcast domain'lerine kullanıcılar ekleyerek esneklik sağlayabilirsiniz. Bu kurulum ayrıca, hem hatalı network interface card'ların (NIC) sebep olduğu broadcast fırtınalarını durdurmak hem de aradaki bir cihazın, tüm ağ topluluğu boyunca broadcast fırtınalarını dağıtmasını engellemek için çalıştırılabilir. Bu sıkıntılar, problemlerin üretildiği VLAN'da hala olabilir. Fakat hastalık, rahatsızlığın olduğu bu VLAN ile izole edilecektir.

Diğer avantajı, bir VLAN çok büyüdüğünde, broadcast'lerin daha fazla bant genişliği harcamalarını engellemek için, daha fazla VLAN oluşturabilmenizdir. Bir VLAN'da ne kadar az kullanıcı olursa, broadcast'lerden o kadar az kullanıcı etkilenecektir. Bu tamamıyla güzel bir durumdur. Fakat network servislerini ciddi olarak aklınızda tutmanız ve VLAN oluşturduğunuzda, kullanıcıların bu servislere nasıl bağlanacaklarını bilmeniz gerekmektedir. Herkesin ihtiyaç duyduğu e-mail ve internet erişimi dışındaki tüm servislerin lokalde bulunmasını sağlamak iyi bir hareket olacaktır.

Bir VLAN'ın bir switch'e nasıl güvendiğini anlamak için, ilk olarak geleneksel bir ağa bakmak faydalı olacaktır. Şekil 9.3, fiziksel LAN'ları bir router'a bağlamak için hub'ları kullanarak bir ağın nasıl oluşturulduğunu göstermektedir.



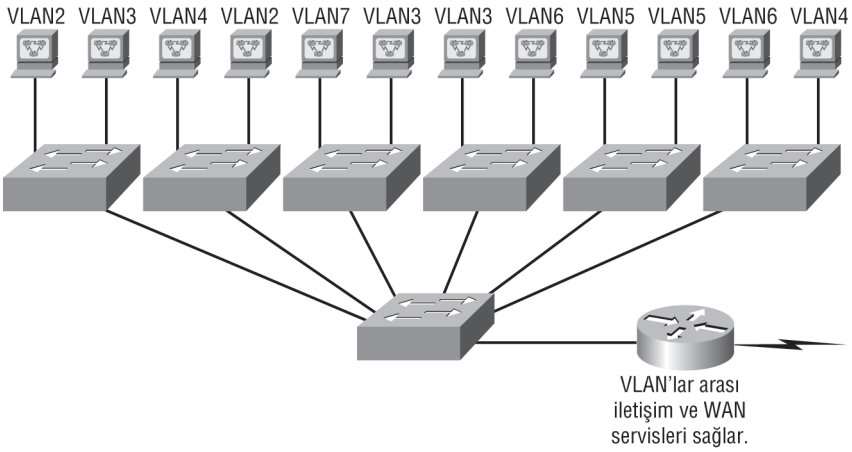

***Şekil 9.3:*** *Bir router'a bağlı fiziksel LAN.*

Burada, her ağın, bir hub port'u ile router'a bağlandığını görebilirsiniz. (Şekilde açıkça görünmese de, her segment ayrıca kendi mantıksal network numarasına sahiptir.) Belirli fiziksel ağa bağlı her düğüm, ağ topluluğunda haberleşebilmek için bu network numarasıyla eşleşmek zorundadır. Her bölümün kendi LAN'ına sahip olduğuna dikkat edin. Böylece, örneğin Satış bölümüne yeni kullanıcılar eklemeniz gerekirse, onları sadece Satış LAN'ının olduğu port'a bağlarsınız ve onlar artık Satış bölümünün collision ve broadcast domain'inin bir parçası olacaklardır. Bu tasarım, uzun yıllardır oldukça iyi çalıştı.

Fakat büyük bir çatlak vardı: Satışçılar için kullanılan hub dolduğunda ve Satış LAN'ına başka bir kullanıcı eklememiz gerektiğinde ne olur? Veya yeni çalışanın yerleştirileceği Satış takımında fiziksel bir yer yoksa ne yaparız? Binanın Finans bölümünde bol miktarda oda olduğunu farz edelim. Yeni Satış ekibi elemanı, Finans elemanlarının bulunduğu yerde oturmak zorunda kalacaktır. Yeni Satış çalışanı, Finans broadcast domain'i üyesi olacağından, yeni eleman, aynı sunucuları görebilecek ve Finans çalışanlarının kullandığı tüm network servislerine erişebilecektir. İkincisi, bu kullanıcılar işlerini yapmaları için gerekli Satış network servislerine erişmek ve Satış bölümü sunucusunda oturum açmak için router'u geçmek zorundadırlar. Bu çok randımanlı olmayacaktır!

Şimdi, switch'in bunu bizim için nasıl başardığına bakalım. Şekil 9.4, problemimizi çözmek için fiziksel sınırları kaldırarak switch'lerin imdada nasıl yetiştiğini göstermektedir. Ayrıca, her bölüm için bir broadcast domain oluşturmak için altı (2'den 7'ye kadar) VLAN'ın nasıl kullanıldığını da göstermektedir. Bundan sonra her switch portu, host'a ve onun yerleştirileceği broadcast domain'ine bağlı olarak, yönetimsel olarak bir VLAN üyeliğine atanacaktır.

Şimdi, Satış VLAN'ına başka bir kullanıcı eklememiz gerekirse, yeni Satış ekibi üyesinin fiziksel olarak nerede olduğuna bakılmaksızın, port'u sadece VLAN 7'ye atayabiliriz. Bu, eski collapsed omurga tasarımına karşı, ağınızı VLAN'larla tasarlamanızın avantajlarından birini göstermektedir. Şimdi, temiz ve basit bir şekilde, Satış VLAN'ında olması gereken her host, sadece VLAN 7'ye atanmaktadır. Önceden yüklü makro'lara sahip yeni switch'leri kullanarak, port'u, bir Desktop bağlantısı olarak yapılandırmak için CNA veya Smartports'u kullanabiliriz. Port yapılandırması, bizim için kolayca tamamlanmaktadır.




| | | |
|---|---|---|
| Pazarlama | VLAN2 | 172.16.20.0/24 |
| Nakliyat | VLAN3 | 172.16.30.0/24 |
| Mühendislik | VLAN4 | 172.16.40.0/24 |
| Muhasebe | VLAN5 | 172.16.50.0/24 |
| Yönetim | VLAN6 | 172.16.60.0/24 |
| Satış | VLAN7 | 172.16.70.0/24 |

***Şekil 9.4:*** *Fiziksel sınırları kaldıran switch'ler.*

VLAN'ları tanımlamaya, VLAN 2'den başladığıma dikkat edin. Numaranın önemi yoktur. Fakat VLAN 1'e ne olduğunu merak edebilirsiniz. Bu VLAN bir yönetim VLAN'ıdır ve bir workgroup için kullanılabilmesine rağmen Cisco, onu sadece yönetimsel amaçlarla kullanmanızı tavsiye eder. VLAN 1'in adını silemez veya değiştiremezsiniz ve varsayılan olarak bir switch'teki tüm port'lar, siz değiştirene kadar VLAN 1'in üyesidirler.

Her VLAN bir broadcast domain olarak kabul edildiğinden, onların kendi subnet numaralarına da sahip olmaları gerekmektedir. (Tekrar Şekil 9.4'ü referans alın.) Ayrıca IPv6 kullanıyorsanız, o zaman her VLAN'a kendi IPv6 network numarasının atanması gerekmektedir. Bu nedenle, kafanız karışmasın. Sadece, VLAN'ların ayrı subnet veya network olduklarını düşünmeye devam edin.

Şekil 9.4'teki host'ların, farklı VLAN'daki bir düğüm veya host'la haberleşmelerini sağlamak için hangi becerikli küçük aracı kullanırız? Tahmin ettiğiniz gibi, bir router! Bu düğüm'lerin, tıpkı ağ topluluğu iletişimi için yapılandırıldıklarındaki gibi (şekil 9.3'te gösterildiği şekilde), kesinlikle bir router'ı veya başka bir katman3 cihazı geçmeleri gerekir. Farklı fiziksel ağlara bağlanmaya çalışmamızla yanı şekilde çalışmaktadır. VLAN'lar arasındaki iletişimin, katman3 bir cihaz üzerinden geçmesi gerekmektedir. Bu nedenle, kısa bir zaman içinde, router'ların ortadan kaldırılmalarını beklemeyin.

*Bu modülün sonuna doğru, switch ağınızdaki VLAN'lar arası routing'i sağlamak için bir router ve 3560 switch'i beraber kullanacağız. 3560'ı, bir katman3 switch olarak, aynı bir router gibi kullanabiliriz.*

NOT

# VLAN Üyelikleri

Çoğu zaman, VLAN'lar, switch port'larını her bir VLAN'a atamaya başlayan bir sistem yöneticisi tarafından oluşturulur. Bu tip VLAN'lar, statik VLAN olarak bilinir. Bu işleme başladığınızda, biraz daha fazla çalışmaktan rahatsız olmazsanız, tüm host cihazlarının donanım adreslerini, bir veritabanına tanımlayın. Böylece switch'leriniz, bir host'u, bir switch'e bağladığınız zaman VLAN'ları dinamik olarak ataması için yapılandırılabilirler. "Açıkça" gibi şeyler söylemekten nefret ediyorum, fakat açıkça, bu tip VLAN'lar dinamik VLAN olarak bilinir. Önümüzdeki birkaç bölümde, hem statik hem de dinamik VLAN'lardan bahsedeceğim.

## Statik VLAN'lar

Statik VLAN'ları oluşturmak, VLAN oluşturmanın en yaygın yoludur ve statik VLAN'ların güvenilir olmasının sebeplerinden bir tanesidir. Port atamasını manuel olarak değiştirmediğiniz müddetçe bu güvenlik, bir VLAN ile ilgili olarak tanımladığımız bir switch port'unun, onu daima korumasından kaynaklanmaktadır.

Statik VLAN yapılandırmasının kurulması ve denetlenmesi oldukça basittir ve ağlardaki kullanıcı hareketlerinin denetlenmesinin gerektiği bir ağ kurulum ortamında oldukça iyi çalışırlar. Port'ların yapılandırılması için network yönetim yazılımı kullanılmasına yardımcı olabilir. Fakat istemezseniz, kullanmak zorunda değilsiniz.

Şekil 9.4'de, her switch port'u, host'un üye olması gereken VLAN bazında VLAN üyelikleriyle yapılandırılmıştır. Cihazın gerçek fiziksel lokasyonunun hiçbir önemi olmadığını hatırlayın. Host'larınızın hangi broadcast domain'e üye olacağı sadece size bağlıdır. Her host'un doğru IP adres bilgisine sahip olması gerektiğini de hatırlayın. Örneğin, VLAN 2'deki her host'u, bu VLAN'ın üyesi yapmak için 172.16.20.0/24 ağına yapılandırmanız gerekmektedir. Bir host'u bir switch'e bağladığınızda, bu port'un VLAN üyeliğini doğrulamak zorunda olduğunuzu akılda tutmanızda iyi bir fikirdir. Host'un ihtiyacı olandan farklı bir üyelik olursa, bu host, bir workgroup sunucusu gibi ihtiyaç duyduğu network servislerine erişmeyi başaramayacaktır.

NOT *Statik Access port'ları, ya bir VLAN'a manuel olarak atanırlar veya IEEE 802.1.x ile kullanmak için bir RADIUS sunucusu üzerinden tanımlanırlar.*

## Dinamik VLAN'lar

Diğer taraftan, dinamik bir VLAN, bir düğümün VLAN atamasını otomatik olarak belirler. Akıllı yönetim yazılımları kullanarak, donanım (MAC) adresleri, protokoller ve hatta dinamik VLAN'ları oluşturan uygulamalarda, VLAN bazlı atamalar yapabilirsiniz.

Örneğin, MAC adreslerinin, merkezi bir VLAN yönetim uygulamasına eriştiğini ve yeni bir düğümü merkeze bağladığınızı düşünelim. Şayet onu atanmamış bir switch port'una bağlarsanız, VLAN yönetim veritabanı, donanım adresini arayacaktır ve switch port'unu doğru VLAN için hem atayacak hem de yapılandıracaktır. Söylemeye gerek yok, bu yönetim ve yapılandırmayı kolaylaştıracaktır. Çünkü kullanıcılar yer değiştirirse, switch otomatik olarak onları doğru VLAN'a kolayca atayacaktır. Fakat anlamanız gereken bir konu var: Başlangıçta veritabanı kurulumunda daha fazla çalışmanız gerekmektedir. Yinede sıkıntıya katlanmaya değer, değil mi?

Bazı güzel haberler var: VLAN'larınızın dinamik adreslemesinde kullanılan MAC adres veritabanını kurmak için VLAN Management Policy Server (VMPS) servisini kullanabilirsiniz. VMPS veritabanı, MAC adreslerini IP adresleriyle otomatik olarak eşleştirir.

Bir dinamik Access port'u, bir VLAN'a ait olabilir (VLAN ID 1'den 4094'e kadar) ve söylediğim gibi, VMPS tarafından, dinamik olarak atanmaktadır. Catalyst 2960 switch, sadece bir VMPS kullanıcısı olabilir. Aynı switch'te, dinamik Access portlara ve trunk port'lara sahip olabilirsiniz. Fakat dinamik Access port'larını bir uç istasyona veya hub'a (başka bir switch'e değil) bağlamak zorundasınız.

# VLAN'ları Tespit Etmek

Switch port'larının, fiziksel bir portla ilgili katman2 interface'ler olduğunu bilin. Bir switch port'u, bir Access port ise sadece tek bir VLAN'a veya bir trunk port ise tüm VLAN'lara ait olabilir. Veya switchport mode ayarlamak için, Dynamic Trunking Protocol'ün (DTP) port bazlı çalışmasına izin verebilirsiniz. DTP bunu, linkin diğer ucundaki port ile görüşerek yapar.

Switch'ler oldukça meşgul cihazlardır. Frame'lerin, network boyunca anahtarlanmaları gibi tüm farklı tipleri takip edebilmek, artı donanım adresine bağlı olarak onları ne yapacaklarını bilmek zorundadırlar. Frame'lerin, üstünden geçtikleri link tiplerine göre farklı şekilde kullanıldığını hatırlayın.

Switch ortamında, iki farklı link tipi vardır:

**Access portları:** Bir Access portu sadece bir VLAN'a aittir ve trafiği sadece bir VLAN'a taşır. Trafik, VLAN etiketlemesi olmadan, doğal şekliyle gönderilip alınır. Bir Access portuna ulaşan herhangi bir şeyin, porta atanan VLAN'a ait olduğu düşünülür. Öyleyse, bir Access portunun, IEEE 802.1Q etiketli bir paket alması durumunda ne yapacağını düşünürsünüz? Doğru, bu paket

atılacaktır. Fakat neden? Access portu, kaynak adresine bakmadığından, etiketlenmiş trafik, sadece trunk portlarında alınıp, iletilecektir.

Bir Access link ile bu, portun yapılandırılmış VLAN'ı olarak belirtilebilir. Bir Access linke bağlanmış bir cihaz, VLAN üyeliğinden habersizdir. Cihaz, kendisinin aynı broadcast domain'inin parçası olduğunu düşünür. Fakat büyük resme sahip değildir, bu nedenle fiziksel network topolojisinden anlamaz.

Bilinmesi gereken diğer önemli bilgi, switch'lerin, bir Access link cihazına gönderilmeden önce, VLAN bilgisini frame'den çıkartmasıdır. Paket route edilmedikçe, Access link cihazlarının, VLAN'ları dışındaki cihazlarla haberleşemediğini hatırlayın. Ve siz bir switch portu, ya bir Access portu veya bir trunk port olarak oluşturabilirsiniz, ikisi birlikte olmaz. Bu nedenle, birini veya diğerini seçmek zorundasınız. Onu bir Access port yaparsanız, bu portun sadece bir VLAN'a atanabileceğini bilmelisiniz.

**Voice Access portları:** Kafanız karışmasın, fakat bir Access portunun sadece bir VLAN'a atanabileceğiyle ilgili söylediğim, aslında neredeyse doğrudur. Bugünlerde, birçok switch, bir switch porttaki Access portuna, ses trafiğiniz için ikinci bir VLAN daha eklemenize izin vermektedir. O, voice VLAN'ı olarak belirtilmektedir. Bu teknik olarak farklı bir link tipi olarak kabul edilse de, hala hem veri hem de ses VLAN'ı için yapılandırılabilen bir Access linkidir. Tek switch portuna hem PC hem de telefon bağlamanıza izin verir. Fakat her cihaz hala ayrı VLAN'larda olacaktır. "Telephony: Voice VLAN'larını Yapılandırmak" bölümünde, ses VLAN'larına detaylı gireceğim ve tüm bunları açıklığa kavuşturacağım.

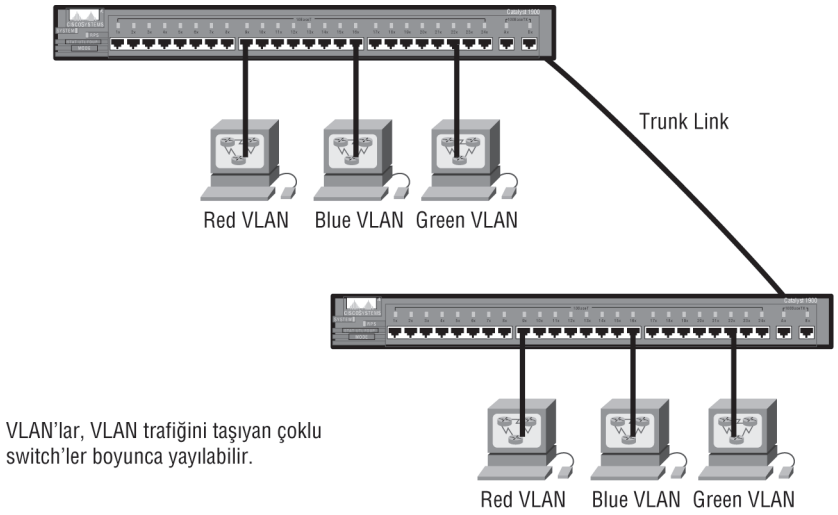
**Trunk portları:** İster inanın ister inanmayın, *trunk port* terimi, aynı anda birçok telefon görüşmesini taşıyan telefon sistemi trunk'larından esinlenmiştir. Bundan dolayı, trunk portları da benzer şekilde, aynı anda birçok VLAN'ı taşıyabilir.

Bir trunk hattı, iki switch, bir switch ile bir router, hatta bir switch ile sunucu arasındaki 100 veya 1000Mbps point-to-point linktir ve aynı anda 1'den 4096'ya kadar birçok VLAN trafiğini taşır. (genişletilmiş VLAN'lar kullanmadan, aslında sadece 1,005'e kadardır.)

Trunking gerçekten iyi bir avantajdır, çünkü onunla, aynı anda, farklı VLAN'ların tamamını tek bir portun parçası yapabilirsiniz. Bu, büyük bir özelliktir, çünkü portları, eşzamanlı iki ayrı broadcast domain'indeki bir sunucuya sahip olması için ayarlayabilirsiniz. Böylece kullanıcılarınız ona bağlanmak ve erişmek için, bir katman3 cihazını (router) geçmek zorunda kalmayacaktır. Trunking kullanmaya başlamanın diğer faydası, switch'lere bağlandığınız zamandır. Trunk linkleri, link boyunca muhtelif miktarlarda VLAN bilgisini taşıyabilir. Fakat varsayılan olarak, switch'leriniz arasındaki linkler trunk değilse, sadece yapılandırılmış VLAN'dan gelen bilgi, bu link boyunca gönderilecektir.

Her VLAN'ı elle silmediğiniz müddetçe, tüm VLAN'ların bilgiyi, bir trunk linkten gönderdiğini bilmek güzeldir. Endişelenmeyin, birazdan bir trunk'tan VLAN'ların nasıl silineceğini göstereceğim.

Şekil 9.5'i inceleyin. Bir switch ağında farklı linklerin nasıl kullanıldığını göstermektedir. Switch'lere bağlı tüm host'lar, aralarındaki trunk linkten dolayı VLAN'larındaki tüm portlarla haberleşebilirler. Switch'ler arasında access link kullandığımızda, bunun switch'ler arasında iletişim için sadece bir VLAN'a izin vereceğini hatırlayın. Görebileceğiniz gibi, bu host'lar, bir switch'e bağlanmak için Access linkleri kullanmaktadırlar. Bu nedenle, sadece tek VLAN'da haberleşmektedirler. Yani, bir router olmadan hiçbir host, VLAN dışıyla iletişim kuramaz. Fakat onlar, kendisiyle aynı VLAN'da yapılandırılmış başka switch'teki host'lara, trunk linkleri üzerinden veri gönderebilir.




***Şekil 9.5:*** *Bir switch ağındaki Access ve trunk linkleri.*

Son olarak, VLAN belirleme yöntemleri ve frame etiketlemeden bahsedeceğim.

## Frame Etiketleme (Tagging)

Şimdi bildiğiniz gibi, VLAN'larınızı birden fazla switch'e dağıtmak için ayarlayabilirsiniz. Çeşitli VLAN'lardaki host'ların bir switch demetine dağılmasını Şekil 9.4'te görebilirsiniz. Bu esnek, güçlü kapasite VLAN'lar oluşturmanın asıl avantajıdır.

Bir switch için bile, karmaşıklık olabilir. Tüm kullanıcı ve frame'lerin, switch fabric ve VLAN'lar boyunca dolaştıklarını takip etmek için bir yol olması gerekir. Switch fabric dediğimde, aynı VLAN bilgisini paylaşan bir switch grubuna işaret ediyorum. Ve bu sadece frame etiketlemesi sahneye çıktığında olur. Bu frame tespit yöntemi, her frame'e eşsiz olarak kullanıcı-tanımlı ID atar. Bazen insanlar bunu, VLAN ID veya renk olarak belirtirler.

Şöyle çalışmaktadır: frame'in ulaştığı her switch, ilk olarak frame etiketinden, VLAN ID'sini tespit etmelidir. Daha sonra, filtre tablosundaki bilgiye bakarak frame'i ne yapacağını belirler. Şayet frame, başka trunk linke sahip bir switch'e ulaşırsa, frame trunk link portuna iletilecektir.

Frame, forward/filter tablosu tarafından, frame'in VLAN ID'sini eşleştiren bir Access link olduğu belirlenen bir çıkışa ulaşınca, switch VLAN tanımlayıcıyı kaldırır. Böylece hedef cihaz, VLAN kimliklerini anlamak zorunda kalmadan, frame'leri alabilecektir.

Trunk portları hakkındaki diğer konu, etiketli ve etiketsiz trafiği eşzamanlı olarak destekleyecekleridir. (şayet, gelecek bölümde bahsedeceğim 802.1Q trunking kullanıyorsanız) Trunk portuna, tüm etiketsiz trafiğin seyahat edeceği bir VLAN için varsayılan port VLAN ID'si (PVID) atanır ve varsayılan olarak daima VLAN 1'dir (fakat başka bir VLAN numarası ile değiştirilebilir)

Benzer şekilde, bir NULL (atanmamış) VLAN ID'si ile etiketsiz veya etiketli trafiğin, varsayılan PVID'li VLAN'a (varsayılan olarak VLAN 1'e) ait olduğu düşünülür. Giden port varsayılan PVID'sine eşit VLAN ID'li bir paket, etiketsiz gönderilir ve sadece VLAN 1'deki host'lar veya cihazlarla haberleşebilir. Diğer tüm VLAN trafiği, bu etikete uygun belirli bir VLAN ile haberleşmek için bir VLAN ile etiketlenmelidir.

## VLAN Belirleme Yöntemleri

VLAN belirlemeyi, bir switch fabric'te dolaşan frame'lerin tamamını izlemek için switch'ler kullanırlar. Bu, hangi switch'in hangi VLAN'a ait olduğunu switch'lerin belirlemesidir ve birden fazla trunk yapma yöntemi vardır.

### Inter-Switch Link (ISL)

Inter-Switch Link (ISL), bir Ethernet frame'ine, VLAN bilgisini açıkça etiketlemenin bir yoludur. Bu etiketleme bilgisi, VLAN'ların, harici bir enkapsülasyon yöntemi (ISL) ile bir trunk link boyunca çoklanmasına izin verir. ISL, switch'in trunk link boyunca bir frame'in VLAN üyeliğini tespit etmesine olanak verir.

ISL çalıştırarak, birçok switch'i birbirine bağlayabilirsiniz ve trunk linklerde trafik switch'ler arasında akarken, VLAN bilgisini hala sağlayabilirsiniz. ISL, bir veri frame'ini, yeni bir başlık ve cyclic redundancy check (CRC) ile enkapsüle ederek katman2'de çalışır.

Şu önemlidir ki, ISL, Cisco switch'lere özgüdür ve sadece FastEthernet ve Gigabit Ethernet linkleri için kullanılmaktadır. *ISL routing*, çok yönlüdür ve bir switch portunda, router interface'lerinde ve bir sunucuya trunk bağlanan sunucu interface kartlarında kullanılabilmektedir.

### IEEE 802.1Q

Standart bir frame etiketleme yöntemi olarak IEEE tarafından oluşturulan IEEE 802.1Q, VLAN'ı tanımlamak için frame'e bir alan ekler. Şayet, Cisco switch link ile farklı bir marka switch arasında trunk yapıyorsanız, trunk için 802.1Q kullanmalısınız.

Şöyle çalışır: İlk olarak, 802.1Q enkapsülasyon ile trunk yapılacak her portu tanımlayın. Portlara, haberleşmeleri için, onları native VLAN yapan belirli bir VLAN ID'si atanmalıdır. Aynı trunk'a yerleştirilen portlar, bu native VLAN ile bir grup oluştururlar ve her port, varsayılan VLAN 1 olan bir kimlik numarası ile etiketlenir. Native VLAN, trunk'ların, herhangi bir VLAN kimliği veya frame etiketi olmadan, alınan bilgiyi taşımasına izin verir.

2960'lar sadece IEEE 802.1Q trunking protokolünü destekler. Fakat 3560'lar, hem ISL hem de IEEE yöntemlerini destekler.

*ISL ve 802.1Q frame-etiketleme yöntemlerinin asıl amacı, switch'ler arası VLAN iletişimini sağlamaktır. Ayrıca, bir frame, Access linkten iletilirse, ISL veya 802.1Q frame etiketlemesinin kaldırılacağını hatırlayın. Etiketleme sadece, trunk linklerde kullanılmaktadır.*

NOT

## VLAN Trunking Protokolü (VTP)

Cisco, bunu da oluşturdu. VLAN Trunking Protokol'ün (VTP) temel hedefi, bir switch ağ topluluğu boyunca oluşturulan tüm VLAN'ları yönetmektir ve VTP'nin VLAN'ları ekleme, silme ve isimlerini değiştirmeye izin verdiği ağ boyunca tutarlılığı sağlamaktır. Bundan sonra bilgi, VTP domain'indeki diğer tüm switch'lere gönderilir.

VTP'nin sunduğu bazı güzel özellikler şunlardır:

- Ağdaki tüm switch'lerdeki VLAN yapılandırmasının tutarlılığı
- Ethernet ile ATM LANE veya hatta FDDI gibi karışık ağlarda VLAN trunk yapılması.
- VLAN'ların doğru şekilde izlenmesi ve görüntülenmesi
- VTP domain'indeki tüm switch'lere eklenen VLAN'ların dinamik olarak rapor edilmesi
- Tak ve Çalıştır VLAN eklenmesi

Çok güzel, fakat ağ boyunca VLAN'larınızı yönetmek için VTP'ye sahip olmadan önce, bir VTP sunucusu oluşturmalısınız. VLAN bilgisini paylaşması gereken tüm sunucular, aynı domain adını kullanmak zorundadır ve bir switch aynı anda sadece bir domain'de olabilir. Yani basit olarak, bir switch, aynı VTP domain'i için yapılandırıldılarsa, VTP domain bilgisini diğer switch'lerle paylaşabilir. Bir ağda, birden fazla bağlı switch'e sahipseniz, bir VTP domain'i kullanabilirsiniz. Fakat switch'leriniz sadece bir VLAN'da ise VTP kullanmanıza gerek yoktur. Şunu aklınızda tutun ki, VTP bilgisi sadece switch'ler arasında bir trunk port yardımıyla gönderilir.

Switch'ler, hem VTP yönetim domain bilgisini hem de bir konfigürasyon revizyon numarasını ve herhangi bir parametreyle, bilinen tüm VLAN'ları yayınlar. Ayrıca, VTP transparent mod olarak tanımlanan bir mod ayarı vardır. Bu modda, trunk portlar boyunca VTP bilgisini iletmesi için switch'leri yapılandırabilirsiniz. Fakat bilgi güncellemesi veya VTP veritabanlarının güncellenmesi kabul edilmeyecektir.

Şayet, arkanızdan VTP domain'inize switch'lerini ekleyen sinsi kullanıcılarınız varsa, şifre kullanabilirsiniz. Her switch'te aynı şifrenin kullanılması gerektiğini unutmayın. Hayal edebileceğiniz gibi, bu küçük problem yönetimsel olarak gerçek bir bela olabilir.

Switch'ler, bir VTP yayınında, eklenen VLAN'ları algılarlar ve sonra beraberinde yeni olarak tanımlanan trunk portlarındaki bilgiyi göndermek için hazırlarlar. Güncellemeler, uyarının bir fazlasından oluşan revizyon numarası olarak gönderilirler. Bir switch, daha yüksek numaraya sahip bir revizyon numarası gördüğünde, aldığı bilginin daha güncel olduğunu bilir. Bundan dolayı, en son bilgi, mevcut veritabanının üzerine yazılır.

Switch'ler arasındaki VLAN bilgisinin haberleşmesinde, VTP için şu üç şartı bilmelisiniz:

- Switch'lerin VTP yönetim domain ismi aynı olmalıdır.
- Switch'lerden birisi, VTP sunucusu olarak yapılandırılmalıdır.
- Router'a gerek yoktur.

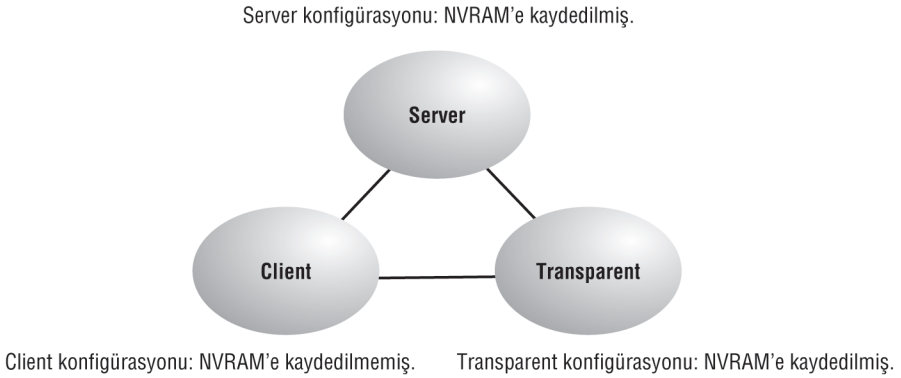
Şimdi bunu iyice öğrenince, VTP dünyasını, VTP modları ve VTP pruning ile derinlemesine araştıracağız.

## VTP Operasyon Modları

Şekil 9.6, bir VTP domain'deki üç farklı operasyon modunu göstermektedir:

**Server:** Bu, Catalyst switch'ler için varsayılan moddur. VTP domain'inizde, bu domain boyunca VLAN bilgisini yaymak için en az bir sunucuya ihtiyacınız vardır. Switch, bir VTP domain'indeki VLAN'ları oluşturabilmek, ekleyebilmek ve silebilmek için server modda olmalıdır. VTP bilgisi, server modda değiştirilmelidir ve server modda bir switch'te yapılan değişiklik, tüm VTP domain'inde yayınlanacaktır. VTP server modda, VLAN konfigürasyonu, NVRAM'e kaydedilir.

**Client:** Client modda, switch'ler VTP sunucularından bilgi alırlar. Fakat onlar ayrıca güncellemeleri alır ve gönderirler. Bu yolla, VTP server'lar gibi davranırlar. Farklılık, VLAN oluşturamaması, değişiklik yapamaması veya silememesidir. Artı, VTP server'ın, client switch'i yeni VLAN için uyarmasından önce, client switch'teki portların hiç biri, yeni bir VLAN'a eklenemez. Ayrıca bilmekte fayda var ki, VTP server'dan alınan VLAN bilgisi, NVRAM'de tutulmaz. Yani, switch yeniden başlatılır veya reload edilirse, VLAN bilgileri silinecektir. İşte bir ipucu: Şayet bir switch'in server olmasını istiyorsanız, ilk olarak onu client yapın, böylece tüm doğru VLAN bilgisini alacaktır. Sonra onu server olarak değiştirin. Böylesi daha kolaydır.




***Şekil 9.6:*** *VTP modları.*

VTP client moddaki bir switch, VTP özet yayınlarını iletecektir ve onları işlemden geçirecektir. Bu switch, bilgileri öğrenecektir, fakat çalışan konfigürasyondaki VTP yapılandırmasını kaydetmeyecektir. Onu, NVRAM'e kaydetmeyecektir. VTP client moddaki switch'ler, sadece öğrenecektir ve VTP bilgisini geçirecektir. Hepsi bu!

### VTP Kullanmayı Ne Zaman Düşünmem Gerekir?

*İşte size senaryo. San Francisco'daki Acme Corporation'da uzman ağ yöneticisi olarak çalışan Bob, hepsi birbirine bağlı 25 switch'e sahiptir ve VLAN'ları broadcast domain'lere ayırmak için yapılandırmak istemektedir. Ne zaman VTP kullanmaya başlaması gerektiğini düşünürsünüz?*

*Eğer, birden fazla switch'i ve çoklu VLAN'ları olduğu an VTP kullanmalıydı şeklinde cevaplarsanız, haklısınız. Şayet sadece bir switch'iniz varsa, VTP gereksizdir. Ağınızda VLAN'ları yapılandırmıyorsanız, ona gerek yoktur.*

*Switch ağ'ınızı ilk olarak ayağa kaldırdığınızda, ana switch'in VTP server ve diğer tüm switch'lerin VTP client olduğunu doğrulayın. Ana VTP server'da VLAN'ları oluşturduğunuzda, tüm switch'ler VLAN veritabanını alacaktır.*

*Şayet mevcut bir switch ağınız varsa ve yeni bir switch eklemek istiyorsanız, yüklemeden önce, onu VTP client olarak yapılandırdığınızdan emin olun. Şayet olmazsanız, oldukça yüksek ihtimaldir, küçük switch'iniz, diğer tüm switch'lere yeni VTP veritabanı gönderecektir. Mevcut tüm VLAN'larınızı, nükleer bir patlama gibi harap edecektir.*

**Transparent:** Transparent moddaki switch'ler, VTP domain'ine katılmazlar veya VLAN veritabanını paylaşmazlar. Fakat hala VTP yayınlarını, yapılandırılmış tüm trunk linkleri boyunca ileteceklerdir. Onlar, diğer switch'lerden gizledikleri bir veritabanı tuttuklarından, VLAN'lar oluşturabilir, değiştirebilir ve silebilirler. NVRAM'de tutulmasına rağmen, transparent moddaki VLAN veritabanı sadece lokal olarak önemlidir. Transparent modun tüm amacı, uzak switch'lerin, aynı VLAN atamalarında yer almayan bir switch'i kullanarak VTP server'dan VLAN veritabanını almasını sağlamaktır. VTP sadece VLAN ID 1'den 1005'e kadar olan normal VLAN aralığındaki VLAN'lar hakkında bilgi öğrenir. 1005'ten büyük ID'li VLAN'lar genişletilmiş VLAN aralığı olarak tanımlanırlar ve VLAN veritabanında tutulmazlar. 1006'dan 4094'e kadar VLAN ID'si oluşturacağınız zaman,switch, transparent modda olmalıdır. Bu VLAN'ları kullanmanız çok seyrektir. Diğer önemli bir şey:1 ve 1002 ile 1005 arası VLAN IDleri, tüm switch'lerde otomatik olarak oluşturulur ve silinemezler.

## VTP Pruning

VTP size broadcast, multicast ve unicast paketlerinin miktarını düşürmesi için yapılandırarak bant genişliğini koruma olanağı verir. Bunun adı pruning'dir. VTP pruning'in etkin olduğu switch'ler, broadast'leri sadece bilgiye gerçekten sahip olması gereken trunk linklerine gönderir.

Bunun anlamı şudur: SwitchA'nın VLAN 5 için ayarlanmış bir portu yoktur ve bir broadcast, VLAN 5'e gönderilmektedir. Bu broadcast, SwitchA'ya giden trunk linkten geçmeyecektir. Varsayılan olarak, VTP pruning tüm switch'lerde pasif durumdadır.

Bir VTP server'da pruning'i etkinleştirirseniz, onu tüm domain'de etkinleştirmiş olursunuz. Varsayılan olarak, 2'den 1001'e kadarki VLAN'larda pruning seçilebilir. VLAN 1'de asla seçemezsiniz, çünkü o bir yönetim VLAN'ıdır. VTP pruning, VTP versiyon 1 ve 2 ile desteklenmektedir.

`Show inteface trunk` komutunu kullanarak, varsayılan olarak bir trunk link boyunca kabul edilen tüm VLAN'ları görebilirsiniz:

```
S1#sh int trunk

Port        Mode         Encapsulation  Status        Native vlan
Fa0/1       auto         802.1q         trunking      1
Fa0/2       auto         802.1q         trunking      1

Port        Vlans allowed on trunk
Fa0/1       1-4094
Fa0/2       1-4094

Port        Vlans allowed and active in management domain
Fa0/1       1
Fa0/2       1

Port        Vlans in spanning tree forwarding state and not
pruned
Fa0/1       1
Fa0/2       none
S1#
```

Yukarıdaki çıktıya bakarak, VTP pruning'in varsayılan olarak pasif olduğunu görebilirsiniz. Sadece tek bir komut gerektirir ve tüm switch ağınızda, listelenen VLAN'lar için etkinleşmiş olur. Gelin ne olduğuna bakalım:

```
S1#config t
S1(config)#int f0/1
S1(config-if)#switchport trunk ?
  allowed  Set allowed VLAN characteristics when interface is
  in trunking mode
  native   Set trunking native characteristics when interface
  is in trunking mode
  pruning  Set pruning VLAN characteristics when interface is
  in trunking mode
S1(config-if)#switchport trunk pruning ?
  vlan  Set VLANs enabled for pruning when interface is in
  trunking mode
S1(config-if)#switchport trunk pruning vlan 3-4
```

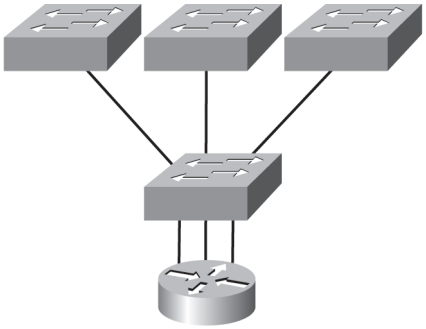
Pruning uygulanabilecek geçerli VLAN'lar, 2 ile 1001 arasıdır. Genişletilmiş VLAN aralığında pruning yapılmaz ve pruning'e katılmayan bu VLAN'lar, tüm trafiği alabilmektedir.

# VLAN'lar Arası Routing

Bir VLAN'daki host'lar, kendi broadcast domain'inde yaşar ve serbestçe haberleşebilirler. VLAN'lar, ağı kısımlara ayırır ve OSI'nin katman2'sinde trafiği ayırırlar. Switch'lere hala neden ihtiyacımız olduğundan bahsettiğim zaman söylediğim gibi, host'ların veya diğer IP-adreslenebilir bir cihazın, VLAN'lar arası haberleşmesini istiyorsanız, bir katman3 cihaza sahip olmanız gerekmektedir.

Bunun için her VLAN için bir interface'e sahip bir router veya ISL ya da 802.1Q routing'i destekleyen bir router kullanabilirsiniz. ISL ve 802.1Q routing'i destekleyen en ucuz router, 2600 serisi router'lardır. (Üretimleri bittiğinden, onları ikinci-el satıcılardan satın almak zorundasınız.) 1600, 1700 ve 2500 serisi router'lar, ISL ve 802.1Q routing'i desteklememektedir. Sadece 802.1Q'yu destekleyen, en az boş bir 2800 öneririm. Cisco gerçekten ISL'i bırakmaktadır, bu nedenle muhtemelen sadece 802.1Q kullanmalısınız. (Ben hiç görmedim ama 2800'deki bazı IOS'lar, hem ISL hem de 802.1Q'yu destekleyebilir.)

Şekil 9.7'de görüldüğü gibi, az sayıda VLAN'a sahipseniz (iki veya üç), iki ya da üç Fast Ethernet bağlantısına sahip bir router alabilirsiniz. Ve ev kullanıcıları için 10BaseT yeterlidir. Bunu sadece ev kullanıcıları için söylüyorum. Bunun dışında, FastEthernet ve Gigabit interface'leri öneririm.



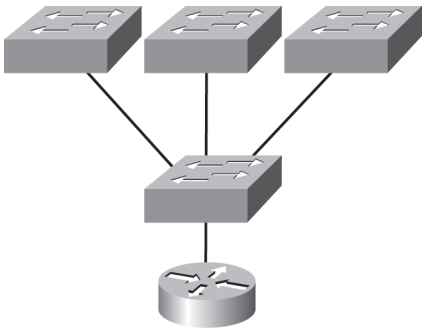
Bir router, inter-VLAN bağlantısı için üç VLAN'ı birbirine bağlıyor. Her VLAN için bir interface.

***Şekil 9.7:*** *Router ve farklı VLAN bağlantıları.*

Şekil 9.7'de her router interface'inin bir Access linke bağlandığını görürüz. Yani, router interface'lerinin IP adreslerinin her biri, her VLAN'daki tüm kullanıcılar için varsayılan ağ geçidi adresi olacaktır.

Eğer router interface'inden daha çok VLAN'a sahipseniz, tek Fast Ethernet interface'inde trunking ayarlayabilirsiniz veya Cisco 3560 ya da yüksek kapasiteli 6500 gibi katman3 switch satın alabilirsiniz.

Her VLAN için bir router interface'i kullanmak yerine, tek Fast Ethernet interface'i kullanabilir ve ISL ya da 802.1Q çalıştırabilirsiniz. Şekil 9.8, ISL veya 802.1Q ile yapılandırıldığında, router'daki bir Fast Ethernet interface'inin nasıl görüneceğini göstermektedir. Bu, tüm VLAN'ların tek interface'den haberleşmelerine izin verir. Cisco buna router on a stick der.



Router, inter-VLAN bağlantısı sağlamak için sadece bir router interface'i kullanarak bütün VLAN'ları birbirine bağlıyor (router on a stick).

***Şekil 9.8:*** *Router on a stick.*

Bunun hem bir darboğaz hem de tek arıza noktası oluşturduğunu belirtmem gerekir. Bu nedenle, host/VLAN sayınız sınırlıdır. Ne kadardır? Bu sizin trafik seviyenize bağlıdır. Bir şeyleri gerçekten doğru yapmak için, yüksek kapasiteli switch ve backplain'de routing kullanmanız daha iyi olacaktır. Şayet sadece bir router'a sahipseniz, bu yöntemi kullanmak çok ucuza gelir, değil mi?

# VLAN'ları yapılandırmak

Sizi şaşırtabilir, fakat VLAN'ları yapılandırmak oldukça basittir. Her VLAN'da olmasını istediğiniz kullanıcıları bilmeye gerek yoktur, o büyük zaman kaybıdır. Her birine dâhil olmasını istediğiniz kullanıcılar için, oluşturmak ve kurmak istediğiniz VLAN sayısına karar verdiğinizde, sizin ilk VLAN'ınızı dünyaya getirme zamanınız gelmiştir.

Bir Cisco Catalyst switch'te VLAN'ları yapılandırmak için, `vlan` global komutunu kullanın. Aşağıdaki örnekte, üç farklı bölüm için üç VLAN oluşturarak, S1 switch'inde VLAN'ların nasıl yapılandırılacağını göstereceğim. VLAN 1'in varsayılan olarak native ve yönetim VLAN'ı olduğunu hatırlayın:

```
S1#config t
S1(config)#vlan ?
  WORD      ISL VLAN IDs 1-4094
  internal  internal VLAN
S1(config)#vlan 2
```

```
S1(config-vlan)#name Sales
S1(config-vlan)#vlan 3
S1(config-vlan)#name Marketing
S1(config-vlan)#vlan 4
S1(config-vlan)#name Accounting
S1(config-vlan)#^Z
S1#
```

Yukarıdaki çıktıdan, 2'den 4094'ya kadar VLAN oluşturabileceğinizi görebilirsiniz. Bu genelde doğrudur. Söylediğim gibi, VLAN'lar gerçekte, sadece 1005'e kadar oluşturulabilmektedir ve 1 ve 1002 ile 1005 arasındaki VLAN'ları kullanamaz, ismini değiştiremez ve silemezsiniz. Çünkü onlar rezerve edilmişlerdir. Bu numaraların üzerindeki VLAN'lar, genişletilmiş VLAN'lar olarak belirtilirler ve switch'iniz VTP transparent moda ayarlanmadıkça, veritabanına kaydedilmezler. Bu VLAN'ların gerçek ağlarda çok sık kullanıldığını görmezsiniz. VTP server moda ayarlandığında, S1 switch'imin VLAN 4000'e ayarlanmasıyla ilgili örnek şöyledir:

```
S1#config t
S1(config)#vlan 4000
S1(config-vlan)#^Z
% Failed to create VLANs 4000
Extended VLAN(s) not allowed in current VTP mode.
%Failed to commit extended VLAN(s) changes.
```

İstediğiniz VLAN'ları oluşturduktan sonra, onları kontrol etmek için, `show vlan` komutunu kullanabilirsiniz. Fakat switch'in tüm portlarının VLAN 1'de olduğuna dikkat edin. Bir port ile ilgili VLAN'ı değiştirmek için, her interface'e gitmek ve hangi VLAN'ın parçası olacağını söylemelisiniz.

**NOT** *Bir switch portu veya portlarına atanana kadar, oluşturulan VLAN'ların kullanım dışı olduğunu ve başka bir ayarlama olmadan, tüm portların VLAN 1'e atandığını hatırlayın.*

VLAN'lar oluşturulduğunda, `show vlan` (kısaca `sh vlan`) komutu ile konfigürasyonunuzun doğruluğunu kontrol edin:

```
S1#sh vlan
VLAN Name                    Status    Ports
---- ------------------------------    -
1    default                  active    Fa0/3, Fa0/4, Fa0/5, Fa0/6
                                        Fa0/7, Fa0/8, Gi0/1
2    Sales                    active
3    Marketing                active
4    Accounting               active
 [output cut]
```

Bu tekrar olarak görülebilir, fakat önemlidir ve hatırlamanızı istiyorum: varsayılan VLAN olduğundan, VLAN 1'i değiştiremez, silemez ve ismini değiştiremezsiniz. Ve bu durumu değiştiremezsiniz. O, tüm switch'ler için varsayılan olarak native VLAN'dır ve Cisco, onu yönetim VLAN'ınız olarak kullanmanızı tavsiye eder. Özel olarak farklı bir VLAN'a atanmamış paketler, native VLAN'a gönderilecektir.

Yukarıdaki S1 çıktısında, Fa0/3'ten Fa0/8'e kadar olan ve Gi0/1 uplink portlarının tamamının, VLAN 1'de olduğunu görebilirsiniz. Peki, port 1 ve 2 nerededir? Trunk ve EtherChannel birleştirmesi yaptığım bir önceki modülden bunu hatırlayın. Trunk olan bir port, VLAN veritabanında

görünmeyecektir. Trunk portlarınızı görmek için, `show interface trunk` komutunu kullanmalısınız.

Şimdi, oluşturulan VLAN'ları görebiliriz. Switch portlarını bunlardan birine atayabiliriz. Voice Access portları istisna olmak üzere, her port sadece bir VLAN'ın parçası olabilir. Daha önce incelediğimiz trunking ile bir portu tüm VLAN trafiği için kullanmaya elverişli hale getirebilirsiniz. Bunu sonra açıklayacağım.

## Switch Portlarını VLAN'lara Atamak

Portun taşıyacağı trafik türünü, artı ait olabileceği VLAN'ların numarasını belirleyen bir üyelik modu atayarak VLAN'a dâhil olan bir portu yapılandırabilirsiniz. `switchport` interface komutunu kullanarak, bir switch'teki her portu, belirli bir VLAN'da (access port) olması için yapılandırabilirsiniz. Ayrıca birçok portu aynı anda, Modül5'te bahsettiğimiz `interface range` komutu ile yapılandırabilirsiniz.

Bir portta, statik üyelik veya dinamik üyelikler yapılandırabileceğinizi hatırlayın. Bu kitabın amacı için, sadece statik olanı işleyeceğim. Aşağıdaki örnekte, fa0:3 interface'ini, VLAN 3'e ayarlayacağım. Bu, S1 switch'inden, Host A cihazına bağlantıdır:

```
S1#config t
S1(config)#int fa0/3
S1(config-if)#switchport ?
  access         Set access mode characteristics of the interface
  backup         Set backup for the interface
  block          Disable forwarding of unknown uni/multi cast
addresses
  host           Set port host
  mode           Set trunking mode of the interface
  nonegotiate    Device will not engage in negotiation protocol
on this
                 interface
  port-security  Security related command
  priority       Set appliance 802.1p priority
  protected      Configure an interface to be a protected port
  trunk          Set trunking characteristics of the interface
  voice          Voice appliance attributes
```

Yukarıdaki çıktıda görünen bazı yeni komutlar var. Zaten işlediğim bazı komutları da görebiliriz. Fakat endişelenmeyin, `access`, `mode`, `nonegotiate`, `trunk` ve `voice` komutlarını bu bölümde en kısa sürede işleyeceğim. Gelin, S1'deki bir Access portu ayarlayarak başlayalım. VLAN'lara sahip gerçek switch'lerde, muhtemelen en yaygın şekilde kullanılan port tipleridir:

```
S1(config-if)#switchport mode ?
  access   Set trunking mode to ACCESS unconditionally
  dynamic  Set trunking mode to dynamically negotiate access or
  trunk mode
  trunk    Set trunking mode to TRUNK unconditionally

S1(config-if)#switchport mode access
S1(config-if)#switchport access vlan 3
```

`Switchport mode access` komutu ile başlayarak, switch'e bunun bir katman2 portu olduğunu söylüyorsunuz. Daha sonra, `switchport access` komutu ile bir VLAN'ı porta atayabilirsiniz. `interface range` komutunu kullanırsanız, aynı anda birçok portu seçebileceğinizi hatırlayın. `Dynamic` ve `trunk` komutları, sadece trunk portları için kullanılmaktadır.

Nerdeyse hepsi bu. Cihazları, her bir VLAN portuna bağlarsanız, onlar sadece, aynı VLAN'daki diğer cihazlarla konuşabileceklerdir. VLAN'lar arası haberleşmeyi sağlamak istiyoruz ve bunu yapacağız. Fakat ilk olarak, trunking hakkında biraz daha bilgi almanız gerekir.

## Trunk Portlarını Yapılandırmak

2960 switch sadece IEEE 802.1Q enkapsülasyon yöntemini kullanır. Bir Fast Ethernet portunda trunking ayarlamak için `trunk [parameter]` interface komutunu kullanın. 3560 switch'te çok küçük farklılık vardır ve gelecek bölümde bunu göstereceğim.

Aşağıdaki switch çıktısı, trunk'ı aktif olarak ayarlamak için, fa0/8 interface'indeki trunk yapılandırmasını göstermektedir:

```
S1#config t
S1(config)#int fa0/8
S1(config-if)#switchport mode trunk
```

Aşağıdaki liste, bir switch interface'i yapılandırıldığında, farklı seçenekleri listelemektedir:

**`Switchport mode access`:** Önceki bölümde bundan bahsetmiştim, fakat bu, interface'i (Access portunu) kalıcı nontrunking moduna koyar ve linki, trunk olmayan linke dönüştürmeyi görüşür. Komşu interface'inin trunk bir interface olduğuna bakmaksızın, interface, trunk olmayan bir interface olur. Port, atanmış bir katman2 portu olacaktır.

**`Switchport mode dynamic auto`:** Bu mod, interface'in linki, bir trunk linke dönüştürmesini mümkün kılar. Şayet komşu interface, trunk veya desirable moda ayarlanırsa interface, trunk bir interface olur. Bu, yeni Cisco switch'lerdeki Ethernet interface'leri için varsayılan switchport modudur.

**`Switchport mode dynamic desirable`:** Bu, interface'in linki trunk bir linke dönüştürmesi için aktif olarak girişimde bulunmasını mümkün kılar. Şayet komşu interface, trunk, desirable veya auto moda ayarlanırsa, interface, trunk bir interface olur. Bazı eski switch'lerde bu modu varsayılan olarak gördüm, fakat artık öyle değil. Varsayılan, şimdi dynamic auto'dur.

**`Switchport mode trunk`:** Interface'i, kalıcı trunking moda koyar ve komşu linki trunk bir linke dönüştürmeyi görüşür. Komşu interface, trunk bir interface olmasa bile, interface, trunk interface olur.

**`Switchport nonegotiate`:** Interface'in DTP frame'leri üretmesini engeller. Bu komutu sadece, interface switchport modu, Access veya trunk olduğunda kullanabilirsiniz. Komşu interface'i, trunk bir link kurmak için manuel olarak yapılandırmalısınız.

NOT *Dynamic Trunking Protocol (DTP), hem iki cihaz arasındaki bir linkte trunking görüşmesi hem de 802.1Q veya ISL enkapsülasyon görüşmesi için kullanılmaktadır. Nonegotiate komutunu, atanmış trunk portlarının soru sormasını istemediğimde kullanırım.*

Bir interface'de trunking'in pasif olması için portu tekrar atanmış bir katman2 portuna dönüştürecek, `switchport mode access` komutunu kullanın.

### Cisco Catalyst 3560 Switch ile Trunking

Gelin, Cisco Catalyst 3560 switch'e bir bakalım. 3560'ın katman3 servisleri sağlaması dışında, yapılandırma, 2960 ile neredeyse aynıdır. Artı, 3560 hem ISL hem de IEEE 802.1Q trunking enkapsülasyon yöntemi çalıştırabilir. 2960 sadece 802.1Q çalışır. Bunların hepsini düşünerek, 3560 switch ile ilgili VLAN enkapsülasyon farkına kısaca bir göz atalım.

3560, 2960'da olmayan, `encapsulation` komutuna sahiptir:

```
Core(config-if)#switchport trunk encapsulation ?
  dot1q      Interface uses only 802.1q trunking encapsulation
 when trunking
  isl        Interface uses only ISL trunking encapsulation
 when trunking
  negotiate  Device will negotiate trunking encapsulation with
peer on
             interface
Core(config-if)#switchport trunk encapsulation dot1q
Core(config-if)#switchport mode trunk
```

Görebileceğiniz gibi, 3560 switch'e, 802.1Q (`dot1q`) enkapsülasyon veya ISL enkapsülasyon ekleme seçeneğine sahibiz. Enkapsülasyon'u ayarladıktan sonra, hala interface modunu trunk olarak ayarlamak zorundasınız. Gerçekten, ISL enkapsülasyon yöntemini kullanmaya devam etmeniz oldukça seyrektir. Cisco, ISL'den uzaklaşmaktadır. Onun yeni router'ları artık onu desteklememektedir.

## Bir Trunk'ta Allowed VLAN'ları Tanımlamak

Belirttiğim gibi, trunk portlar, varsayılan olarak tüm VLAN'lardan bilgileri alır ve gönderir. Bir frame etiketsizse, o yönetim VLAN'ına gönderilecektir. Bu genişletilmiş VLAN'lara da atanır.

Fakat trunk linkte mevcut VLAN'ların dolaşmasından trafiği korumak için VLAN'ları izin verilenler listesinden çıkartabiliriz. Bunu şöyle yaparız:

```
S1#config t
S1(config)#int f0/1
S1(config-if)#switchport trunk allowed vlan ?
  WORD    VLAN IDs of the allowed VLANs when this port is in
  trunking mode
  add     add VLANs to the current list
  all     all VLANs
  except  all VLANs except the following
  none    no VLANs
  remove  remove VLANs from the current list
S1(config-if)#switchport trunk allowed vlan remove ?
  WORD  VLAN IDs of disallowed VLANS when this port is in
trunking mode
S1(config-if)#switchport trunk allowed vlan remove 4
```

Yukarıdaki komut, S1 f0/1 portunda ayarlanmış trunk linkinde VLAN 4 için alınan ve gönderilen tüm trafiğin iptal edilmesine sebep olmuştur. Bir trunk linkten VLAN 1'i kaldırmaya çalışabilirsiniz. Fakat o hala CDP, PAgP, LACP, DTP ve VTP gibi yönetimsel bilgileri alacak ve gönderecektir.

Belli bir VLAN aralığını kaldırmak için sadece tire (-) kullanın:

```
S1(config-if)#switchport trunk allowed vlan remove 4-8
```

Şayet birileri kazayla bazı VLAN'ları, trunk linkten silerse ve trunk'ın tekrar varsayılana ayarlanmasını isterseniz, şu komutu kullanın:

```
S1(config-if)#switchport trunk allowed vlan all
```

Veya şu komutta aynı şeyi yapacaktır:

```
S1(config-if)#no switchport trunk allowed vlan
```

VLAN'lar arası routing'e başlamadan önce, VLAN'lar için pruning'in nasıl yapılandırılacağını göstermek istiyorum.

## Trunk Native VLAN'ı Değiştirmek veya Modifiye Etmek

Siz aslında trunk port native VLAN'ı, VLAN 1'den değiştirmeyi istemezsiniz. Fakat bunu yapabilirsiniz ve bazı insanlar güvenlik nedenleriyle bunu yaparlar. Native VLAN'ı değiştirmek için aşağıdaki komutu kullanın:

```
S1#config t
S1(config)#int f0/1
S1(config-if)#switchport trunk ?
  allowed  Set allowed VLAN characteristics when interface is
  in trunking mode
  native   Set trunking native characteristics when interface
  is in trunking mode
  pruning  Set pruning VLAN characteristics when interface is
  in trunking mode
S1(config-if)#switchport trunk native ?
  vlan  Set native VLAN when interface is in trunking mode
S1(config-if)#switchport trunk native vlan ?
  <1-4094>  VLAN ID of the native VLAN when this port is in
  trunking mode
S1(config-if)#switchport trunk native vlan 40
S1(config-if)#^Z
```

Böylece trunk linkimizdeki native VLAN'ımızı 40 olarak değiştirdik. Ve `show running-config` komutunu kullanarak, trunk link altındaki konfigürasyonu görebilirim:

```
!
interface FastEthernet0/1
 switchport trunk native vlan 40
 switchport trunk allowed vlan 1-3,9-4094
 switchport trunk pruning vlan 3,4
!
```

Durun bir dakika, bunun kolay olduğunu ve çalışmaya başlayacağını düşünmediniz, değil mi? Şayet tüm switch'ler, trunk linklerde ayarlı aynı native VLAN'a sahip değillerse, şu hatayı alırız:

```
19:23:29: %CDP-4-NATIVE_VLAN_MISMATCH: Native VLAN mismatch
discovered on FastEthernet0/1 (40), with Core FastEthernet0/7
(1).
19:24:29: %CDP-4-NATIVE_VLAN_MISMATCH: Native VLAN mismatch
discovered on FastEthernet0/1 (40), with Core FastEthernet0/7
(1).
```

Aslında bu mesaj iyi, anlaşılması kolay bir hatadır. Böylece ya diğer uçtaki trunk link(ler)imize gidip, native VLAN'ı değiştiririz veya varsayılan native VLAN'ı tekrar ayarlarız. Bunu şöyle yaparız:

```
S1(config-if)#no switchport trunk native vlan
```

Şimdi trunk linkimiz, native VLAN olarak VLAN 1'i kullanıyor. Tüm switch'lerin aynı native VLAN'ı kullanmak zorunda olduğunu yoksa ciddi problemlerinizin olacağını hatırlayın. Router'u switch ağımıza bağlayarak ve VLAN'lar arası iletişimi yapılandırarak onu karıştıralım.

## VLAN'lar Arası Routing'i Yapılandırmak

Varsayılan olarak, aynı VLAN'a üye host'lar haberleşebilirler. Bunu değiştirmek ve VLAN'lar arası iletişime izin vermek için, bir router veya katman3 switch'e ihtiyacınız var. Ben, router kullanmak ile başlayacağım.

Bir Fast Ethernet interface'inde ISL veya 802.1Q routing'i desteklemek için, router'un interface'i, her biri bir VLAN için olmak üzere mantıksal interface'lere bölünmektedir. Bunlar, subinterface olarak belirtilmektedir. Bir Fast Ethernet veya Gigabit interface'ten, interface'i trunk yapmak için `encapsulation` komutunu kullanabiliriz:

```
ISR#config t
ISR(config)#int f0/0.1
ISR(config-subif)#encapsulation ?
  dot1Q  IEEE 802.1Q Virtual LAN
ISR(config-subif)#encapsulation dot1Q ?
  <1-4094>  IEEE 802.1Q VLAN ID
```

İsmi ISR olan 2811 router'umun sadece 802.1Q'yu desteklediğine dikkat edin. ISL enkapsülasyon çalıştırmak için eski-model bir router'a ihtiyacımız olabilir. Fakat sıkıntıya ne gerek var?

Subinterface numarası sadece lokal olarak önemlidir, bu nedenle router'da hangi subinterface numarası kullanıldığının bir önemi yoktur. Çoğu zaman bir interface'i route etmek istediğim VLAN ile aynı numarayla yapılandıracağım. Subinterface numaralarının sadece yönetimsel amaçlar için kullanılmasından dolayı bunun hatırlanması kolaydır.

Her VLAN'ın ayrı bir subnet olduğunu anlamanız gerçekten önemlidir. Mecbur değilsiniz, fakat VLAN'larınızı ayrı subnet'ler olarak yapılandırmanız iyi bir fikirdir. Bu nedenle, sadece böyle yapın.

Şimdi, hem VLAN'lar arası routing'i yapılandırmaya hem de bir switch VLAN ortamına bağlı host'ların port IP adreslerini belirlemeye tamamen hazır olduğunuza emin olmam gerekiyor. Her zamanki gibi, ortaya çıkabilecek her problemi çözebilmek de iyi bir hedeftir.

İlk olarak, şekil 9.9'a bakarak başlayalım ve ondaki router ile switch yapılandırmasına bakalım. Kitaptaki bu noktada, VLAN'lardaki host'ların her birinin IP adresini, maskını ve varsayılan ağ geçidini belirleyebilmelisiniz.

Bundan sonraki basamak, hangi subnet'lerin kullanıldığını anlamaktır. Şekildeki router yapılandırmasına bakarak, VLAN 1 ile 192.168.1.64/26 ve VLAN 10 ile 192.168.1.128'i kullandığımızı görebilirsiniz. Ve routing konfigürasyonuna bakarak, port 2 ve 3'ün VLAN 1'de ve port 4'ün VLAN 10'da olduğunu görebilirsiniz. Yani, HostA ve HostB, VLAN 1'de ve HostC, VLAN 10'dadır.

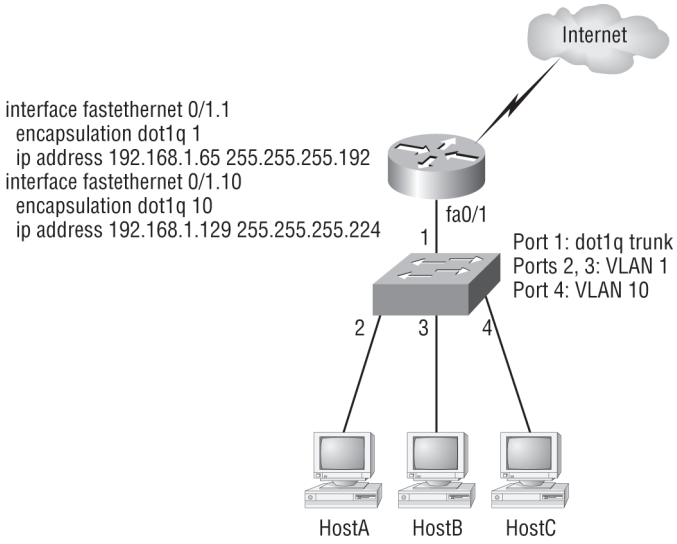
Host'ların IP adreslerinin ne olması gerektiği aşağıdaki gibidir:

**HostA:** 192.168.1.66, 255.255.255.192, default gateway 192.168.1.65

**HostB:** 192.168.1.67, 255.255.255.192, default gateway 192.168.1.65

**HostC:** 192.168.1.130, 255.255.255.224, default gateway 192.168.1.129

Host'lar, aralıktaki herhangi bir adrese sahip olabilirdi. Ben, varsayılan ağ geçidinden sonraki geçerli IP adresini seçtim. Bu çok zor değil, değil mi?




***Şekil 9.9:*** *VLAN'lar arası yapılandırma örneği 1.*

Şimdi şekil 9.9'u kullanarak, router ile bir link kurmak ve enkapsülasyon için IEEE versiyonunu kullanarak VLAN'lar arası iletişim sağlamak için gerekli komutları gözden geçirelim. Komutların, kullandığınız switch tipine göre çok az değişeceğini aklınızda tutun.

Bir 2960 switch için, aşağıdakini kullanın:

```
2960#config t
2960(config)#interface fa0/1
2960(config-if)#switchport mode trunk
```

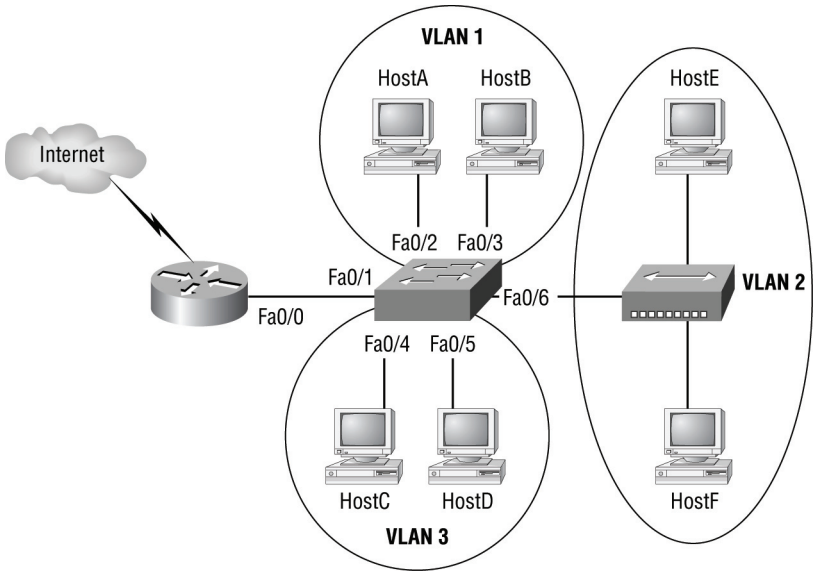
Zaten bildiğiniz gibi, 2960 switch'lerde sadece 802.1Q enkapsülasyon çalışır, bu nedenle onu belirtmeye gerek yok. Zaten yapamazsınız! 3560 için, esasen aynıdır, fakat hem ISL hem de 802.1Q çalışabileceğinden, kullanacağınız trunking protokolünü belirtmek zorundasınız.

NOT *Bir trunk link oluşturduğunuzda, varsayılan olarak tüm VLAN'ların veriyi geçirmesine izin verileceğini hatırlayın.*

Gelin Şekil 9.10'a bakalım ve ondan ne öğrenebileceğimizi anlayalım. Bu şekil, her birinde iki host olan üç VLAN'ı göstermektedir.

Şekil 9.10'daki router, fa0/1 switch porta bağlıdır ve VLAN 2, port f0/6'da yapılandırılmıştır. Diyagrama bakarak, Cisco'nun bilmenizi beklediği şeyler şunlardır:

- Router, subinterface'ler kullanarak switch'e bağlanmıştır.
- Router'a bağlanan switch portu bir trunk portudur.
- İstemcilere ve hub'a bağlı switch portları, Access porttur, trunk değildir.




***Şekil 9.10:*** *VLAN'lar arası yapılandırma örneği 2.*

Switch'in yapılandırması şöyle görünecektir:

```
2960#config t
2960(config)#int f0/1
2960(config-if)#switchport mode trunk
2960(config-if)#int f0/2
2960(config-if)#switchport access vlan 1
2960(config-if)#int f0/3
2960(config-if)#switchport access vlan 1
2960(config-if)#int f0/4
2960(config-if)#switchport access vlan 3
2960(config-if)#int f0/5
2960(config-if)#switchport access vlan 3
2960(config-if)#int f0/6
2960(config-if)#switchport access vlan 2
```

Router'u yapılandırmadan önce, mantıksal ağımızı yapılandırmamız gerekir:

**VLAN 1:** 192.168.10.16/28

**VLAN 2:** 192.168.10.32/28

**VLAN 3:** 192.168.10.48/28

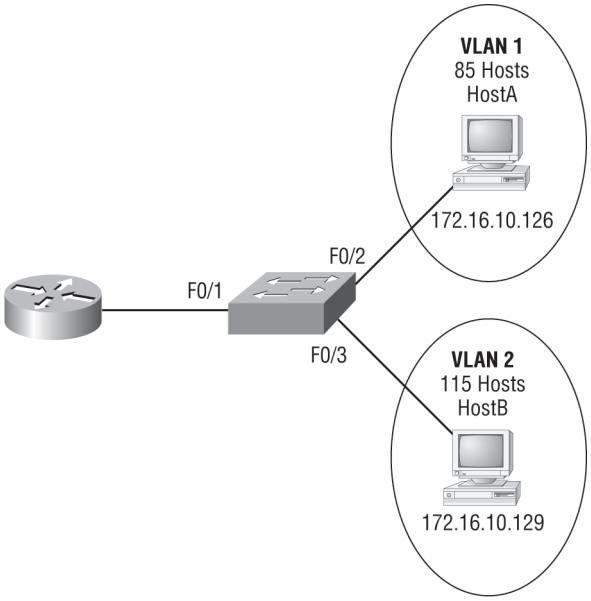
Router'un yapılandırması artık şöyle olur:

```
ISR#config t
ISR(config)#int f0/0
ISR(config-if)#no ip address
ISR(config-if)#no shutdown
ISR(config-if)#int f0/0.1
ISR(config-subif)#encapsulation dot1q 1
ISR(config-subif)#ip address 192.168.10.17 255.255.255.240
ISR(config-subif)#int f0/0.2
```

```
ISR(config-subif)#encapsulation dot1q 2
ISR(config-subif)#ip address 192.168.10.33 255.255.255.240
ISR(config-subif)#int f0/0.3
ISR(config-subif)#encapsulation dot1q 3
ISR(config-subif)#ip address 192.168.10.49 255.255.255.240
```

Her VLAN'daki host'lara, kendi adres aralığından bir adres atanabilir ve varsayılan ağ geçidi, bu VLAN'daki router'a atanan IP adresi olmalıdır.

Şimdi, diğer bir şekle bakalım ve cevaplara bakmadan (hile yok) switch ve router konfigürasyonlarını belirleyip belirleyemeyeceğinizi anlayalım. Şekil 9.11, iki VLAN'a sahip switch'e bağlı bir router'u göstermektedir. Her VLAN'daki bir host'a bir IP adresi atanmaktadır. Bu IP adreslerine bağlı olarak router ve switch konfigürasyonlarınız nasıldır?




***Şekil 9.11:*** *VLAN'lar arası yapılandırma örneği 3.*

Host'lar, bir subnet mask'ı listelemediğinden, blok boyutunu anlamak için, her VLAN'da kullanılan host'ların sayısına bakmak zorundasınız. VLAN 1, 85 host'a ve VLAN 2, 115 host'a sahiptir. Bunların hepsi, /25 mask (225.255.255.128) ile 128'lik bloğa sığmaktadır.

Şimdi subnet'lerin 0 ve 128 olduğunu bilmelisiniz. 0 subnet'i (VLAN 1), 1-126 host aralığına, 128 subnet'i (VLAN 2), 129-254 aralığına sahiptir. HostA, 126 IP adresine sahip olduğundan, HostA ile HostB neredeyse aynı subnet'te gibi görünüyor, az daha şaşıracaktınız. Fakat şaşırmadınız ve şu ana kadar öğrendikleriniz sayesinde, bu gibi şeylere kanmayacak kadar zekisiniz.

Switch konfigürasyonu şöyledir:

```
2960#config t
2960(config)#int f0/1
2960(config-if)#switchport mode trunk
2960(config-if)#int f0/2
2960(config-if)#switchport access vlan 1
2960(config-if)#int f0/3
2960(config-if)#switchport access vlan 2
Here is the router configuration:
ISR#config t
```

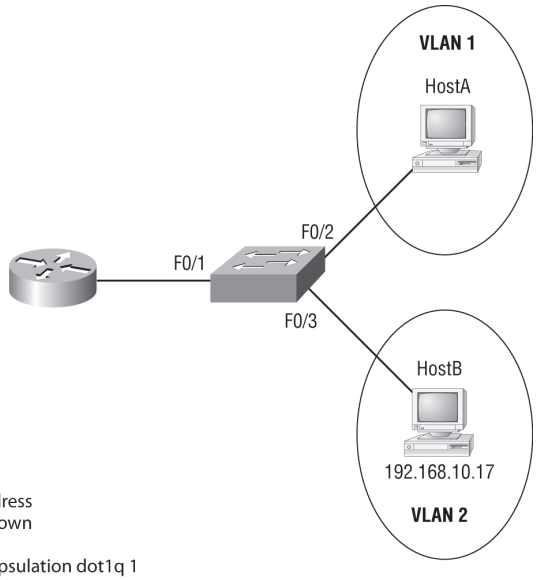
```
ISR(config)#int f0/0
ISR(config-if)#no ip address
ISR(config-if)#no shutdown
ISR(config-if)#int f0/0.1
ISR(config-subif)#encapsulation dot1q 1
ISR(config-subif)#ip address 172.16.10.1 255.255.255.128
ISR(config-subif)#int f0/0.2
ISR(config-subif)#encapsulation dot1q 2
ISR(config-subif)#ip address 172.16.10.254 255.255.255.128
```

VLAN 1 için host aralığındaki ilk adresi ve VLAN 2 için aralıktaki son adresi kullandım. Fakat aralıktaki herhangi bir adres de çalışırdı. Host'un varsayılan ağ geçidini, router'ın adresini kullanarak ayarlamalısınız.

Şimdi, sonraki örneğe geçmeden önce, switch'teki IP adresinin nasıl ayarlandığını bildiğinizden emin olmam gerekiyor. VLAN 1, genellikle yönetimsel bir VLAN olduğundan, bu adres havuzundan bir IP adresi kullanacağız. Switch'in IP adresi şöyle ayarlanır: (Sürekli dır dır etmiyorum, gerçekten bunu bilmelisiniz!)

```
2960#config t
2960(config)#int vlan 1
2960(config-if)#ip address 172.16.10.2 255.255.255.128
2960(config-if)#no shutdown
```

Bir tane daha örnek yapalım ve sonra kesinlikle kaçırmayı istemeyeceğiniz diğer önemli bir konu olan VTP'ye geçeceğiz. Şekil 9.12'de iki VLAN vardır. Router yapılandırmasına bakarak, HostA'nın IP adresi, mask'ı ve varsayılan ağ geçidi nedir? HostA'nın adresi olarak, aralıktaki son IP adresini kullanın:




```
Router#config t
Router(config)#int f0/0
Router(config-if)#no ip address
Router(config-if)#no shutdown
Router(config-if)#int f0/0.1
Router(config-subif)# encapsulation dot1q 1
Router(config-subif)# ip address 192.168.10.129 255.255.255.240
Router(config-subif)# int f0/0.2
Router(config-subif)# encapsulation dot1q 2
Router(config-subif)# ip address 192.168.10.46 255.255.255.240
```

***Şekil 9.12:*** *VLAN'lar arası yapılandırma örneği 4.*

```
Router#config t
Router(config)#int f0/0
Router(config-if)#no ip address
```

```
Router(config-if)#no shutdown
Router(config-if)#int f0/0.1
Router(config-subif)# encapsulation dotlq 1
Router(config-subif)# ip address 192.168.10.129 255.255.255.240
Router(config-subif)# int f0/0.2
Router(config-subif)# ancapsulation dotlq0.2
Router(config-subif)# ip address 192.168.10.46 255.255.255.240
```

Router konfigürasyonuna gerçekten dikkatle bakarsanız (bu şekilde hostname, sadece Router'dır), basit ve hızlı bir şekilde cevaplanabilir. İki subnet'te, 16 blok boyutunda, /28 (255.255.255.240) mask kullanıyor. VLAN 1 için router'un adresi, 128 subnet'indedir. Sonraki subnet, 144'tür, bu nedenle VLAN 1'in broadcast adresi 143'tür ve geçerli host aralığı, 129-142'dir. Öyleyse, host adresi şu olur:

**IP Address:** 192.168.10.142

**Mask:** 255.255.255.240

**Default Gateway:** 192.168.10.129

# VTP'yi Yapılandırmak

Tüm Cisco switch'ler, varsayılan olarak VTP server olarak ayarlanmıştır. VTP'yi ayarlamak için, ilk olarak kullanmak istediğiniz domain ismini belirlemelisiniz. Ve tabi ki, bir switch'teki VTP bilgisini yapılandırınca, onu doğrulamanız gerekir.

VTP domain'i oluşturduğunuzda, domain ismi, şifre, çalışma modu ve switch'in pruning kapasitesi ayarlarını içeren çok sayıda seçeneğe sahip olursunuz. Bu bilgilerin tamamını ayarlamak için, `vtp` global komutunu kullanın. Aşağıdaki örnekte, S1 switch'ini `vtp server` olarak, VTP domain'ini `Lammle` olarak ve VTP şifresini `todd` olarak ayarlayacağım:

```
S1#config t
S1#(config)#vtp mode server
Device mode already VTP SERVER.
S1(config)#vtp domain Lammle
Changing VTP domain name from null to Lammle
S1(config)#vtp password todd
Setting device VLAN database password to todd
S1(config)#do show vtp password
VTP Password: todd
S1(config)#do show vtp status
VTP Version                     : 2
Configuration Revision          : 0
Maximum VLANs supported locally : 255
Number of existing VLANs        : 8
VTP Operating Mode              : Server
VTP Domain Name                 : Lammle
VTP Pruning Mode                : Disabled
VTP V2 Mode                     : Disabled
VTP Traps Generation            : Disabled
```

```
MD5 digest                      : 0x15 0x54 0x88 0xF2 0x50 0xD9
0x03 0x07
Configuration last modified by 192.168.24.6 at 3-14-93 15:47:32
Local updater ID is 192.168.24.6 on interface Vl1 (lowest numbered
VLAN interface found)
```

Tüm switch'lerin varsayılan olarak server moda ayarlandığını ve bir switch'teki herhangi bir VLAN bilgisini değiştirmek isterseniz de, mutlaka VTP server modda olmanız gerektiğini hatırladığınızdan lütfen emin olun. VTP bilgisini yapılandırdıktan sonra, yukarıda verilen çıktıdaki gibi `show vtp` komutunu kullanarak, onun doğruluğunu kontrol edebilirsiniz. Yukarıdaki switch çıktısı VTP domain'ini, VTP şifresini ve switch'in modunu göstermektedir.

Core ve S2 switch'ini VTP bilgisiyle yapılandırmaya geçmeden önce, `show vtp status` çıktısının, lokal olarak desteklediği maksimum VLAN sayısını 255 olarak gösterdiğine bakalım. Bir switch'te 1,000'in üzerinde VLAN oluşturabileceğiniz halde, 255'den daha fazla switch'iniz varsa ve VTP kullanıyorsanız kesinlikle problem olacak gibi görünmektedir. Evet, bu problemdir. Bir switch'te 256. VLAN'ı yapılandırmaya çalışıyorsanız, yeterli donanım kaynağı olmadığını belirten küçük bir hata mesajı alacaksınız ve sonra VLAN'ı kapatacaktır. 256. VLAN, `show vlan` komutu çıktısında askıda görünecektir. Çok hoş bir durum değil!

Core ve S2 switch'lerine gidelim ve onları Lammle VTP domain'ine katalım. Domain adının küçük/büyük harf duyarlı olduğunu hatırlamak çok önemlidir. VTP hata affetmez, ufak bir hatada çalışmayacaktır.

```
Core#config t
Core(config)#vtp mode client
Setting device to VTP CLIENT mode.
Core(config)#vtp domain Lammle
Changing VTP domain name from null to Lammle
Core(config)#vtp password todd
Setting device VLAN database password to todd
Core(config)#do show vtp status
VTP Version                     : 2
Configuration Revision          : 0
Maximum VLANs supported locally : 1005
Number of existing VLANs        : 5
VTP Operating Mode              : Server
VTP Domain Name                 : Lammle
VTP Pruning Mode                : Disabled
VTP V2 Mode                     : Disabled
VTP Traps Generation            : Disabled
MD5 digest                      : 0x2A 0x6B 0x22 0x17 0x04 0x4F
0xB8 0xC2
Configuration last modified by 192.168.10.19 at 3-1-93 03:13:16
Local updater ID is 192.168.24.7 on interface Vl1 (first interface
found)
S2#config t
S2(config)#vtp mode client
Setting device to VTP CLIENT mode.
S2(config)#vtp domain Lammle
```

```
Changing VTP domain name from null to Lammle
S2(config)#vtp password todd
Setting device VLAN database password to todd
S2(config)#do show vtp status
VTP Version                     : 2
Configuration Revision          : 0
Maximum VLANs supported locally : 1005
Number of existing VLANs        : 5
VTP Operating Mode              : Client
VTP Domain Name                 : Lammle
VTP Pruning Mode                : Disabled
VTP V2 Mode                     : Disabled
VTP Traps Generation            : Disabled
MD5 digest                      : 0x02 0x11 0x18 0x4B 0x36 0xC5
0xF4 0x1F
Configuration last modified by 0.0.0.0 at 0-0-00 00:00:00
```

Güzel, şimdi tüm switch'lerimiz, aynı domain ve şifre ile yapılandırıldı. S1 switch'inde önceden ayarladığım VLAN'lar, Core ve S2 VTP client switch'lerine yayınlanmalıdır. Core ve S2 switch'inde `show vlan brief` komutunu kullanarak bir bakalım:

```
Core#sh vlan brief
VLAN Name                Status    Ports
---- ----------          ------    -----------
1    default             active    Fa0/1,Fa0/2,Fa0/3,Fa0/4
                                   Fa0/9,Fa0/10,Fa0/11,Fa0/12
                                   Fa0/13,Fa0/14,Fa0/15,
                                   Fa0/16,Fa0/17, Fa0/18, Fa0/19,
                                   Fa0/20,Fa0/21, Fa0/22, Fa0/23,
                                   Fa0/24, Gi0/1, Gi0/2
2    Sales                active
3    Marketing            active
4    Accounting           active
[output cut]

S2#sh vlan bri
VLAN Name                 Status    Ports
---- ----------          ------    -----------
1    default              active    Fa0/3, Fa0/4, Fa0/5, Fa0/6
                                    Fa0/7, Fa0/8, Gi0/1
2    Sales                active
3    Marketing            active
4    Accounting           active
[output cut]
```

Bu bölümde daha önce S1 (2960) switch'inde oluşturduğum VLAN veritabanı, VTP yayınlarıyla, Core ve S2 switch'ine yüklenmiştir. VTP, switch ağındaki VLAN isimlendirme tutarlılığını sağlamak

için çok güzel bir araçtır. Şimdi, Core ve S1 switch'lerindeki portlara VLAN atayabiliriz. Onlar, switch'ler arasındaki trunk portları boyunca, S1 switch'indeki aynı VLAN'lardaki host'larla haberleşeceklerdir.

*VTP domain ismi atayabilmeniz için, switch'i VTP server moda ayarlamanız ve bir VLAN oluşturmanız zorunludur.*

NOT

## VTP Hata Tespiti

Switch'lerinizi çapraz kabloyla bağlayın, ışıklar her iki uçta da yeşil olacaktır ve artık çalışıyor! Mükemmel bir dünya, değil mi? Keşke her şey bu kadar kolay olsaydı diyorsunuz değil mi? Aslında VLAN'lar olmadığı taktirde hemen hemen öyle. Fakat eğer VLAN'ları kullanıyorsanız, (kesinlikle kullanmalısınız) switch ağınızda çok sayıda VLAN'a sahipseniz, o zaman VTP kullanmalısınız.

Burada şu problem olabilir: şayet VTP doğru bir şekilde yapılandırılmazsa, çalışmayacaktır. Bu durumda, kesinlikle VTP hata tespiti yapabilmelisiniz. Birkaç konfigürasyona bakalım ve problemleri çözelim. Aşağıdaki iki switch'in çıktısında çalışalım:

```
SwitchA#sh vtp status
VTP Version                     : 2
Configuration Revision          : 0
Maximum VLANs supported locally : 64
Number of existing VLANs        : 7
VTP Operating Mode              : Server
VTP Domain Name                 : RouterSim
VTP Pruning Mode                : Disabled
VTP V2 Mode                     : Disabled
VTP Traps Generation            : Disabled

SwitchB#sh vtp status
VTP Version                     : 2
Configuration Revision          : 1
Maximum VLANs supported locally : 64
Number of existing VLANs        : 7
VTP Operating Mode              : Server
VTP Domain Name                 : GlobalNet
VTP Pruning Mode                : Disabled
VTP V2 Mode                     : Disabled
VTP Traps Generation            : Disabled
```

Bu switch'lere neler oluyor? Neden VLAN bilgisini paylaşmıyorlar? İlk bakışta, her iki sunucu da server moddadır. Fakat problem bu değildir. VTP server moddaki sunucular, VTP kullanarak VLAN bilgisini paylaşacaktır. Problem, onların iki farklı VTP domain'inde olmasıdır. SwitchA, RouterSim VTP domain'inde ve SwitchB, GlobalNet VTP domain'indedir. VTP domain isimleri farklı verildiğinden, VTP bilgilerini paylaşmayacaklardır.

Şimdi, switch'lerinizdeki yaygın VTP domain yapılandırma hatalarına nasıl bakacağınızı biliyorsunuz. Gelin başka bir switch konfigürasyonuna bakalım:

```
SwitchC#sh vtp status
VTP Version                     : 2
Configuration Revision          : 1
Maximum VLANs supported locally : 64
```

```
Number of existing VLANs        : 7
VTP Operating Mode              : Client
VTP Domain Name                 : Todd
VTP Pruning Mode                : Disabled
VTP V2 Mode                     : Disabled
VTP Traps Generation            : Disabled
```

Bekleyin, daha bitmedi. Bu iki switch çıktısına bakalım ve SwitchB'nin SwitchA'dan VLAN bilgisini neden almadığını anlayalım:

```
SwitchA#sh vtp status
VTP Version                     : 2
Configuration Revision          : 4
Maximum VLANs supported locally : 64
Number of existing VLANs        : 7
VTP Operating Mode              : Server
VTP Domain Name                 : GlobalNet
VTP Pruning Mode                : Disabled
VTP V2 Mode                     : Disabled
VTP Traps Generation            : Disabled

SwitchB#sh vtp status
VTP Version                     : 2
Configuration Revision          : 14
Maximum VLANs supported locally : 64
Number of existing VLANs        : 7
VTP Operating Mode              : Server
VTP Domain Name                 : GlobalNet
VTP Pruning Mode                : Disabled
VTP V2 Mode                     : Disabled
VTP Traps Generation            : Disabled
```

Her ikisi de VTP server olduğundan, yanılmış olabilirsiniz, fakat problem bu değildir. Tüm switch'leriniz server olabilir ve onlar hala VLAN bilgisini paylaşacaktır. Cisco aslında, tüm switch'lerin VTP server olarak kalmasını ve VTP VLAN bilgisini yayınlamak istediğiniz switch'in en yüksek revizyon numarasına sahip olduğuna emin olmanızı önermektedir. Tüm switch'ler VTP server ise switch'lerin hepsi, VLAN veritabanını kaydedecektir. SwitchA, SwitchB'den daha yüksek bir revizyon numarasına sahip olduğundan, SwitchB, SwitchA'dan VLAN bilgisini alamıyor. Bu problemin farkına varmanız çok önemlidir.

Bu sorunu çözmek için bazı çözümler vardır. Bunu çözebileceğiniz ilk şey, SwitchB'deki VTP domain adını başka bir isimle değiştirmek, sonra onu tekrar GlobalNet'e ayarlamaktır. Bu SwitchB'deki revizyon numarasını 0'a (sıfır) getirecektir. İkinci bir yaklaşım, SwitchB'deki revizyon numarasını geçene kadar, SwitchA'da VLAN'lar oluşturmak veya silmektir. İkinci yöntemin daha iyi olduğunu söyleyemem. Ben çözüm için diğer yolu seçtim!

# Telephony: Voice VLAN'ları Yapılandırmak

Şayet peş peşe sigara içiyorsanız veya streslenince aşırı yemek yiyorsanız, ufak bir ara verin ve stresinizi giderin. Çünkü dürüst olmam gerekirse bu konu, bu bölümün, hatta bu kitabın en kolay

bölümü değildir. Fakat söz veriyorum, bunu sizin için kolaylaştırmak için elimden gelenin en iyisini yapacağım.

Voice VLAN özelliği, bir IP telefondan IP voice trafiğini taşımak için access portlarını etkinleştirir. Bir switch, Cisco IP telefona bağlandığında, IP telefon, katman3 IP öncelikli voice trafiği ve katman2 class of service (CoS) değeri gönderir. Her ikisi de, voice trafiği için 5'e ayarlanır, diğer tüm trafik varsayılan olarak 0'a ayarlanmıştır.

Veri düzgün şekilde gönderilmezse, bir IP telefon aramasının ses kalitesi bozulabileceğinden, switch, IEEE 802.1p CoS tabanlı quality of service'i (QoS) destekler. (802.1p, MAC seviyesinde QoS uygulamak için bir mekanizma sağlamaktadır) 802.1p alanı, 802.1Q trunk başlığında taşınmaktadır. Şayet bir 802.1Q etiketindeki alanlara bakarsanız, 802.1p bilgisinin gittiği, priority alanı denilen bir alan göreceksiniz. QoS, switch'ten network trafiği göndermek için, organize, öngörülebilen bir biçimde, sınıflandırma ve zamanlama kullanır.

Cisco IP telefonu, yapılandırılabilir bir cihazdır ve onu, bir IEEE 802.1p önceliğiyle trafiği iletmesi için yapılandırabilirsiniz. Ayrıca switch'i, bir IP telefonu tarafından atanan trafik önceliğine güvenmesi veya onu geçersiz kılması için yapılandırabilirsiniz. Tam olarak bizim yapacağımız da budur. Cisco telefonu aslında üç switch portuna sahiptir: biri Cisco switch'e, birisi bir PC cihazına ve diğeri de içerde olan gerçek bir telefona bağlanması için.

Ayrıca, bir Cisco IP telefon bağlanmış access portu, voice trafiği için bir VLAN ve telefona bağlı PC gibi bir cihazdan veri trafiği için başka bir VLAN kullanması için yapılandırabilirsiniz. Switch'teki access portları, Cisco Discovery Protocol (CDP) paketleri göndermesi için yapılandırabilirsiniz. Bu IP telefonuna, switch'e şu yollardan biriyle voice trafiği göndermesi konusunda yol gösterir:

- Bir katman2 CoS priority değeri ile etiketli voice VLAN'ında
- Bir katman2 CoS priority değeri ile etiketli access VLAN'ında
- Etiketsiz (katman2 CoS priority değeri olmayan) access VLAN'ında

Switch ayrıca, Cisco IP telefondaki access portlara bağlı cihazdan gelen etiketli veri trafiğini (IEEE 802.1Q veya 802.1p frame tiplerindeki trafik) işlemden geçirebilir. Switch'teki katman2 access portlarını, CDP paketleri göndermesi için ayarlayabilirsiniz. Bu, ekli Cisco IP telefonuna, IP telefon access portunu şu modlardan biriyle yapılandırması konusunda yol gösterir:

- Trusted modda, Cisco IP telefonundaki access portundan alınan tüm trafik, değişmeden IP telefonundan geçer.
- Untrusted modda, IP telefonundaki access portundan alınan IEE 802.1Q veya 802.1p frame'lerindeki tüm trafik, düzenlenmiş bir katman2 CoS değeri alır. Varsayılan katman2 CoS değeri, 0 (sıfır)'dır. Untrusted mod, varsayılandır.

## Voice VLAN'ı Yapılandırmak

Varsayılan olarak, voice VLAN özelliği pasiftir. Onu, `switchport voice vlan` komutunu kullanarak etkinleştirebilirsiniz. Voice VLAN özelliği, etkinleştirildiğinde tüm etiketsiz trafik portun varsayılan CoS priority'sine göre gönderilir. CoS değeri, IEEE 802.1p veya IEEE 802i1Q etiketli trafik için güvenilir değildir.

Aşağıdakiler, voice VLAN yapılandırmasının ana hatlarıdır:

- Switch access portlarındaki voice VLAN'ı yapılandırmalısınız. Aslında onu ayarlayabilmenize rağmen, voice VLAN, trunk portlarda desteklenmez.
- Voice VLAN, üzerinde düzgün şekilde haberleşmek için IP telefon için switch'te bulunmalı ve aktif olmalıdır. VLAN'ın olup olmadığını anlamak için `show vlan privileged` EXEC komutunu kullanın. Şayet varsa, ekranda listelenecektir.

- Voice VLAN'I etkinleştirmeden önce, `mls qos` global komutunu girerek switch'te QoS'u etkinleştirmeniz ve `mls qos trust cos` interface komutunu kullanarak port güvenlik durumunu trust'a ayarlamanız önerilir.
- Konfigürasyonu göndermek için Cisco IP telefonuna bağlı switch portunda CDP'nin etkinleştirildiğinden emin olmanız gerekmektedir. Bu, varsayılan olarak açıktır. Onu pasifleştirmeniz müddetçe, bir problem yaşamamalısınız.
- PortFast özelliği, voice VLAN yapılandırıldığında, otomatik olarak açıktır. Fakat voice VLAN'ı pasif konuma getirdiğinizde, PortFast özelliği otomatik olarak pasif olmaz.
- Portu varsayılan ayarlarına getirmek için `no switchport voice vlan` interface komutunu kullanın.

## IP Telefon Voice Trafiği

Cisco IP telefona bağlı bir portu, telefonun voice trafiği gönderdiği yöntem ile yapılandırmak için telefona CDP paketleri gönderecek şekilde ayarlayabilirsiniz. Telefon, bir katman2 CoS değerine sahip belirli bir voice VLAN için IEEE 802.1Q frame'lerindeki voice trafiğini taşıyabilir. Hem voice trafiğine daha yüksek bir priority vermek hem de tüm voice trafiğini, native (access) VLAN boyunca iletmek için IEEE 802.1p priority etiketlemesi kullanabilir. Veya access VLAN'daki voice trafiğini göndermek için kendi yapılandırmasını kullanabilir. Tüm konfigürasyonlarda, voice trafiği, voice ayarı için genelde 5 olan bir katman 3 IP öncelik değeri taşır.

Bunu netleştirmek için bazı gerçek örnekler verme zamanı geldiğini düşünüyorum. Bu örnek, dört şeyi nasıl yapılandıracağınızı göstermektedir:

1. Gelen trafiği sınıflandırmak için CoS değeri kullanması için bir IP telefonuna bağlı portun nasıl yapılandırılacağı.
2. Voice trafiği için IEEE 802.1p priority etiketlemesi kullanması için bir portun nasıl yapılandırılacağı.
3. Tüm voice trafiğini taşıması için Voice VLAN (10)'u kullanması için onun nasıl yapılandırılacağı.
4. Ve son olarak, PC verisi taşıması için VLAN 3'ün nasıl yapılandırılacağı.

```
Switch#configure t
Switch(config)#mls qos
Switch(config)#interface f0/1
Switch(config-if)#switchport priority extend ?
  cos    Override 802.1p priority of devices on appliance
  trust  Trust 802.1p priorities of devices on appliance
Switch(config-if)#switchport priority extend trust
Switch(config-if)#mls qos trust cos
Switch(config-if)#switchport voice vlan dot1p
Switch(config-if)#switchport mode access
Switch(config-if)#switchport access vlan 3
Switch(config-if)#switchport voice vlan 10
```

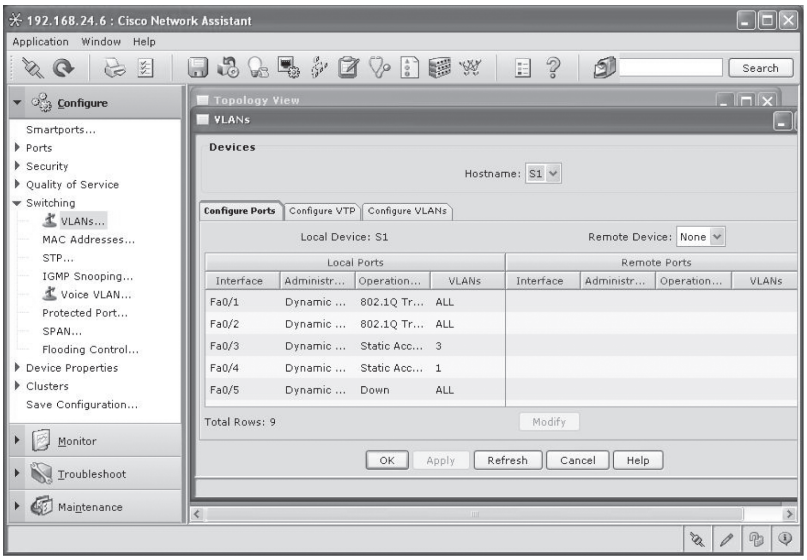
`mls qos trust cos` komutu, CoS paket değeri kullanarak, gelen trafik paketlerini sınıflandırması için interface'i yapılandıracaktır. Etiketsiz paketler için portun varsayılan CoS değeri kullanılacaktır. Fakat port trust durumunu ayarlamadan önce, `mls qos` global komutunu kullanarak, ilk olarak QoS'u global olarak etkinleştirmelisiniz.

İki access VLAN'ı aynı porta nasıl eklediğime dikkat ettiniz mi? Bunu sadece, veri VLAN'ı için bir tane ve voice VLAN'ı için başka bir taneye sahip olduğumda kullanabilirim.

Bu bölüm, muhtemelen tüm kitabın en zor kısmıdır ve açıkçası, onu anlamanız için kullanacağım en basit konfigürasyonu oluşturdum. Önceden düzenlenmiş makrolar ile CNA kullanmak, switch'inize bağlı telefonların yapılandırmasını oldukça kolaylaştırdığından, biraz rahatlayabilirsiniz. Bunun her iki yolla nasıl yapılacağını bilmelisiniz. Şimdiki bölümde, CNA kullanarak yapılan konfigürasyonu göstereceğim.

# VLAN'ları ve VLAN'lar Arası Routing'i Yapılandırmak İçin CNA Kullanmak

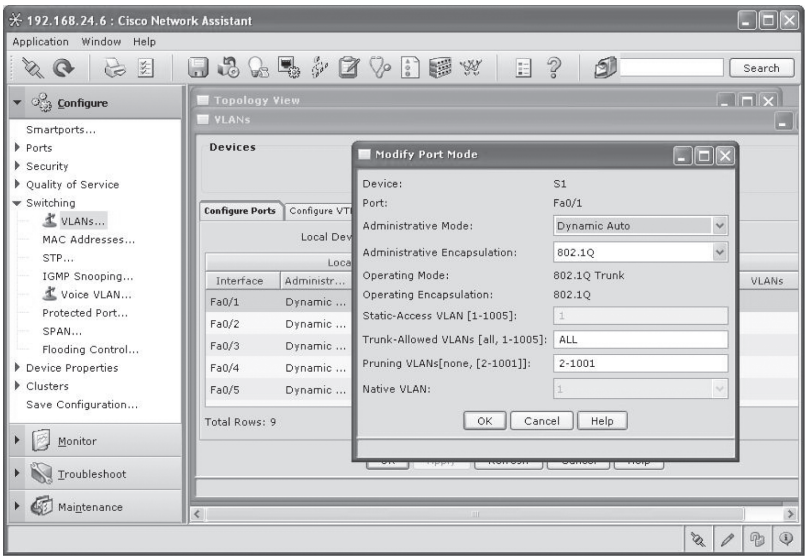
Bu bölüm ağırlıklı olarak gerçek bir router kullanımıyla, VLAN'lar arasında routing oluşturmaya odaklanmıştır. Fakat büyük bir şirket ortamında routing için daha çok, bir switch arkaplanı kullanırsınız. Şimdi, bir 2960'da VLAN'ların nasıl yapılandırıldığını ve bir 3560 switch'te VLAN'lar arasında routing'in nasıl çalıştırılacağını göstermek için command-line interface (CLI) kullanmak yerine, CNA'i kullanacağım. Bundan sonra, önceki bölümde yaptığımız tüm çalışmalarla zorlanmak yerine, yeni CNA'i kullanarak onu yapmanın ne kadar kolay olduğunu göstermek için telefon portlarına sahip switch portlarımızı, CNA Smartports'u kullanarak yapılandıracağım. Ve bana inanın, muhtemelen çoğu kez yaptıklarımıza benzemesine rağmen, biz aslında bu modüldeki telephony'ye ilk adımı atacağız.



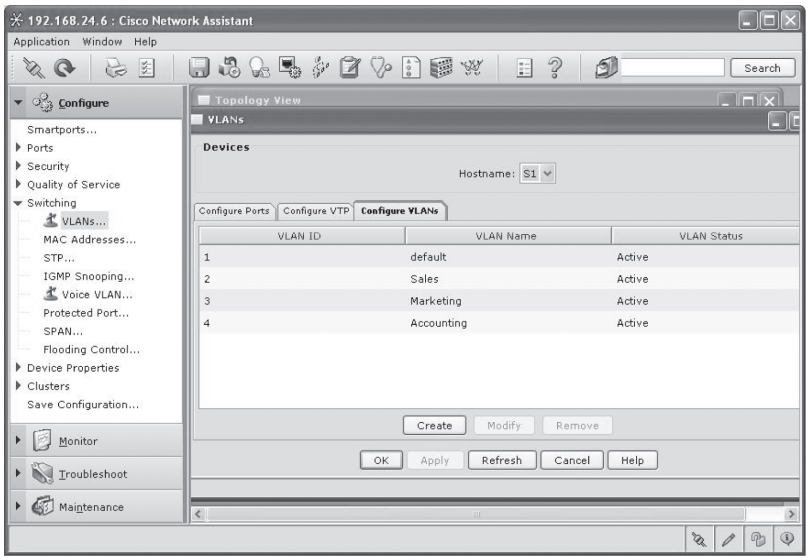

Bu modülde, daha önce yeni üç VLAN ile yapılandırılmış olan 2960 (S1) switch'ime bağlanarak başlayacağım. Sonra, yeni bir voice VLAN ekleyeceğim. İlk ekranda, Configure, Switching ve VLAN'leri tıkladım ve portlarımın durumunu gösteren yeni bir ekran geldi.

Şimdi, port 1 ve 2'nin dinamik olarak trunk olduğunu görebiliriz ve varsayılan olarak `dynamic auto` ile ayarlandıklarından, onların, Core switch bağlantıları otomatik olarak trunk link olacaktır. Bunun daha önce olması için switch'in konfigürasyonunu sildim ve reload ettim. Tabi ki EtherChannel yapılandırmamı da sildim. Fakat flash bellekte tutulan VLAN veritabanı hala oradadır. Port 3'ün VLAN 3'e üye olduğunu da görebilirsiniz (bu bölümde daha önce ayarladığım VLAN Access portu).

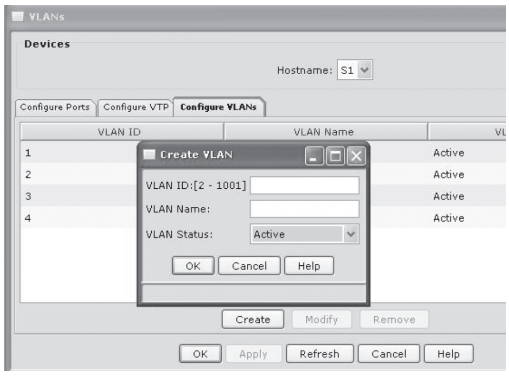
Bu ekrandaki güzel bir özellik, sağ altta bulacağınız Modify butonudur. Port 1'i belirttim, Modify butonuna tıkladım ve bu beni yeni bir ekrana getirdi.



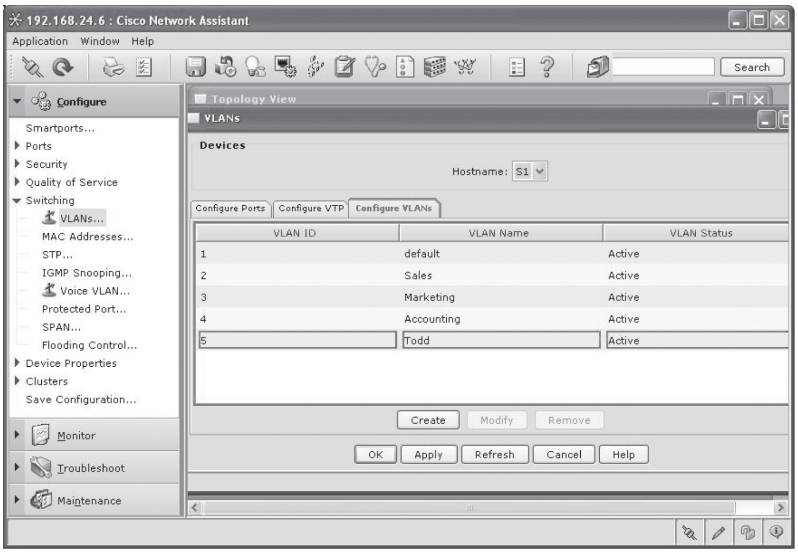

Bu ekranda, sağa doğru ilerleyerek, farklı yönetim modlarını ve enkapsülasyonu değiştirebiliyorum, artı hem trunk portta kabul edilen VLAN'ları hem de VTP pruning'imi ayarlayabiliyorum. Fakat en ilgimi çeken, VLANs ekranındaki Configure VLANs'dır.



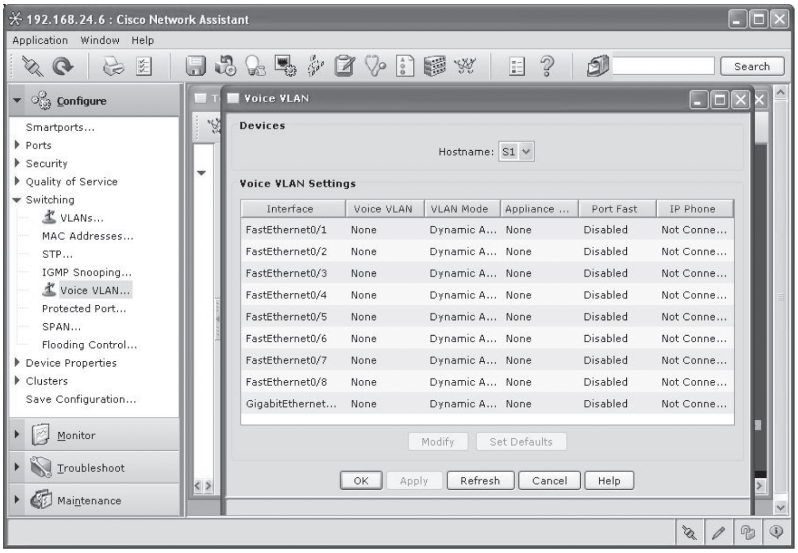

Buradan yapılandırdığım VLAN'ları görebiliyorum ve onları değiştirip, ekleyip silebiliyorum (bunu yapmak için VTP server olmanız gerektiğini hatırlayın). Create butonuna tıkladım ve şu ekranı aldım.



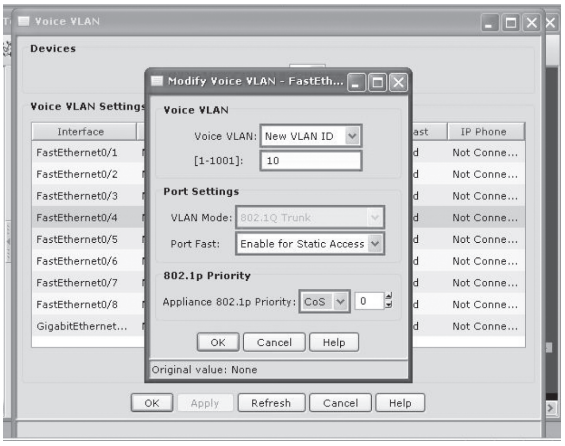

Sonra Create butonuna tıkladım, Todd isminde yeni bir VLAN ekledim ve OK butonuna tıkladım.




Şimdi, biraz eğlenelim ve bir voice VLAN oluşturalım. Configure altında Voice VLAN'a tıkladım ve şu ekran geldi.

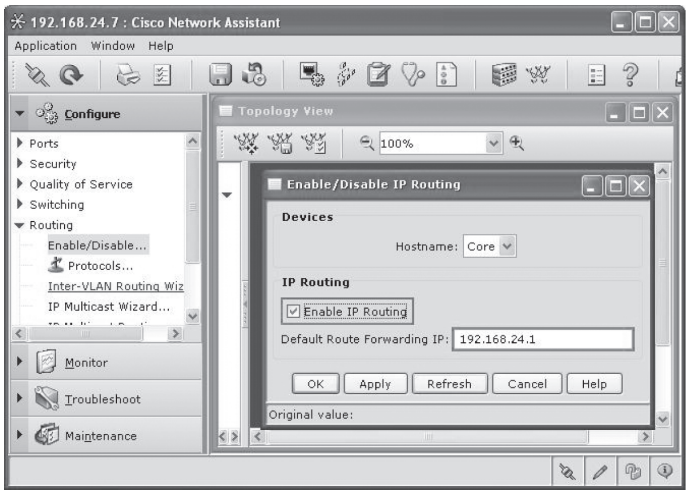



Sonra, telefonumun bağlı olduğu port 4'ü işaretledim ve Modify butonuna tıkladım. Yeni bir voice VLAN (Voice VLAN 10) oluşturdum ve OK butonuna tıkladım.

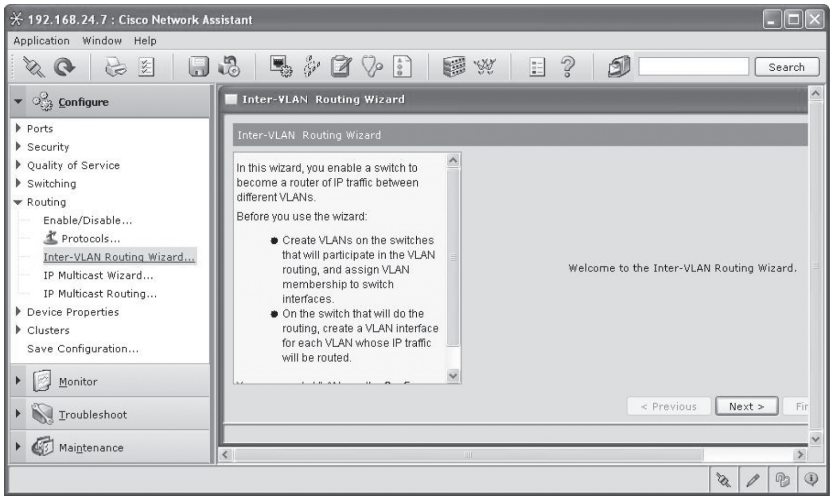


Her şey güzel gidiyor. Şimdi 3560 switch'e gitmek ve şimdiye kadar yaptığımız gibi, router kullanmak yerine switch kullanarak VLAN'lar arası routing'i yapılandırmak istiyorum.

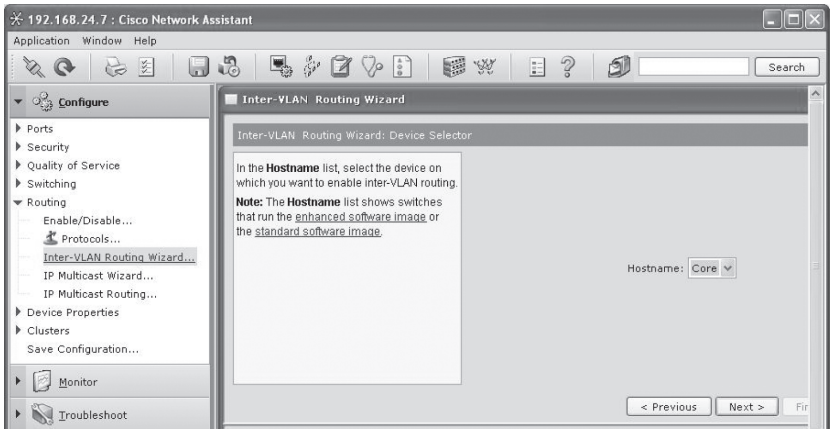
Bu nedenle Core (3560) switch'ime bağlandım, Configure altında Routing ve Enable/Disable seçeneğine tıkladım. Görünen ekrandan, Enable IP Routing seçeneğine tıkladım ve o, ayarladığım varsayılan ağ geçidini otomatik olarak ekledi.



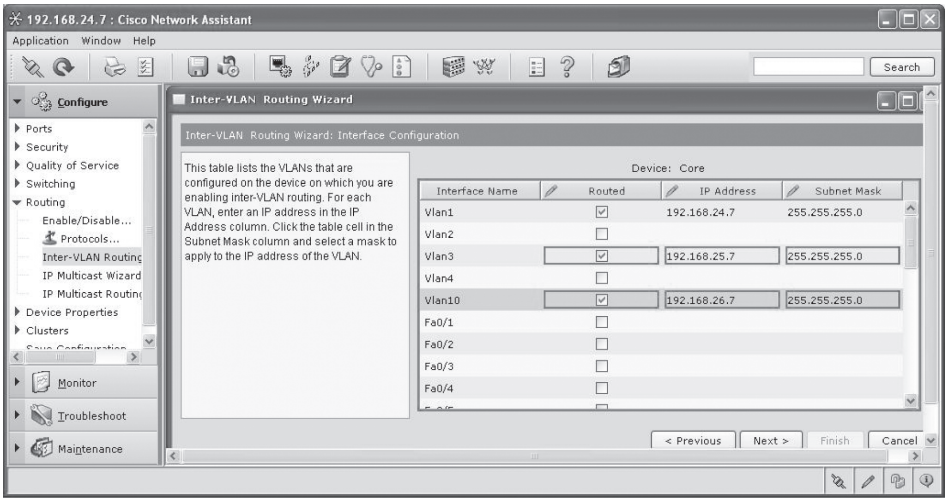

OK butonuna tıkladıktan sonra, Inter-VLAN Routing Wizard'ı tıkladım ve bu ekran göründü.



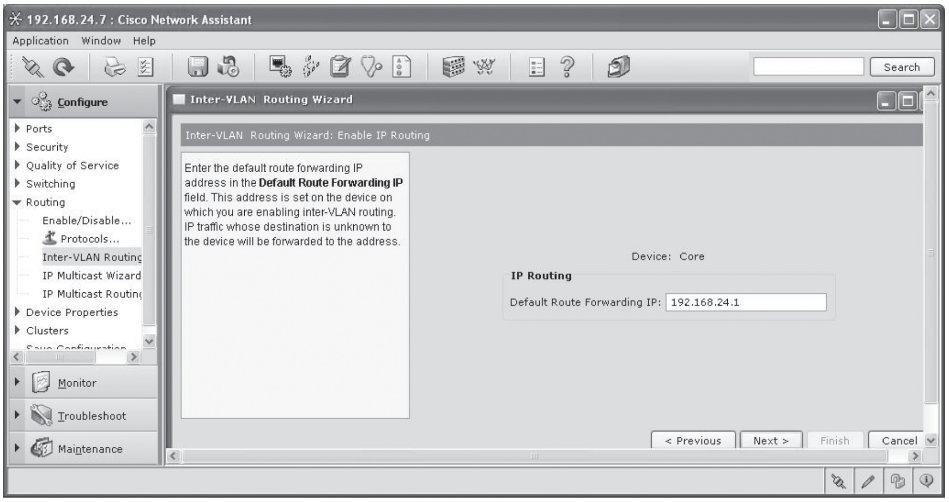

Parti neredeyse bitti sayılır. Bir CLI ile konfigürasyondan çok daha kolay olmayan bir noktaya geldik. RAM'de bitmiş yapılandırmayı gördüğünüzde, ne demek istediğimi anlayacaksınız. Next butonuna tıkladıktan sonra, şu ekrana sahip oldum.



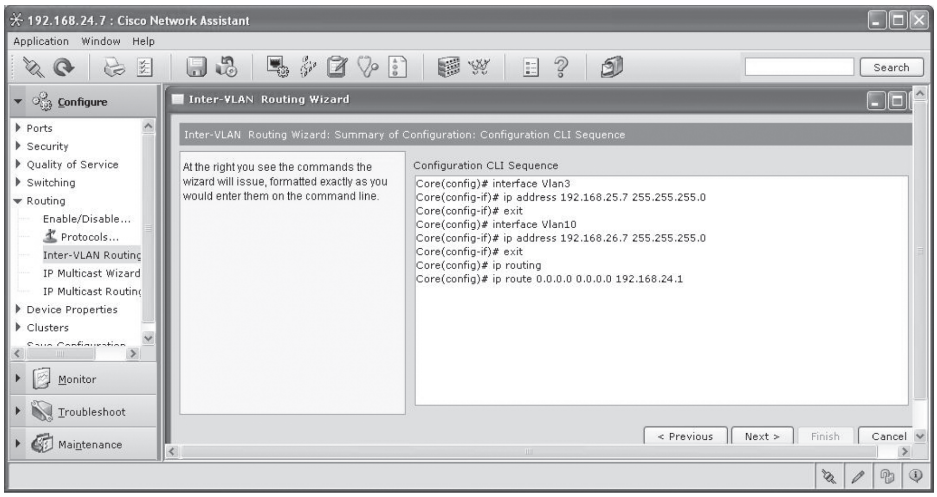
Burada yapılacak pek bir şey yok ama sonraki ekran için Next butonuna tıkladım.




Burası, VLAN'lar arası routing yapılandırmasını yapmak için kullanılan ekrandır. Aralarında VLAN haberleşmesi sağlamak istediğim VLAN'lara tıkladım, her ayrı VLAN için yeni subnet'ler ve subnet mask'ları ekledim ve sonra şu ekrana gitmek için Next butonuna tıkladım.




Bu ekranda zaten, switch'in varsayılan route'u olarak ayarlanan IP varsayılan ağ geçidi vardı. Tekrar Next butonuna tıkladım.

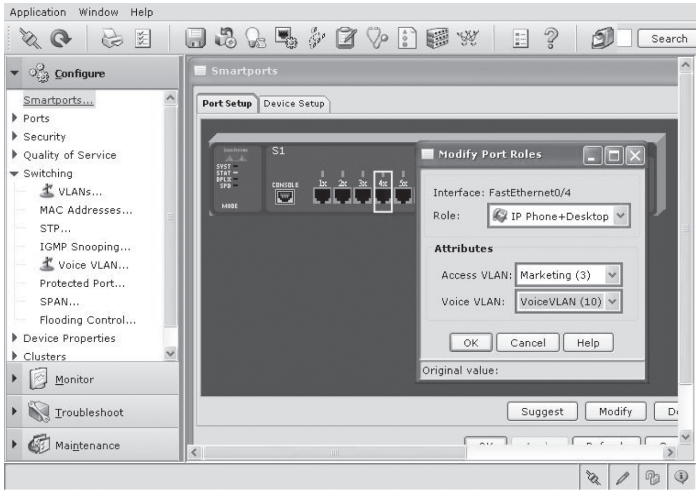


Burası partinin tekrar başladığı yerdir. Tekrar oturdum ve router'ın kendini otomatik yapılandırmasını izledim. Her biri için ayarladığım IP adresleri ile her VLAN için ayrı mantıksal bir interface olduğunu görebilirsiniz. IP routing aktif olur ve sonra varsayılan route ayarlanır. Dürüst olmak gerekirse, bunu CLI kullanarak daha hızlı yazabilirdim, fakat şimdi komutlara bakarak, bir switch'teki VLAN'lar arası routing'i yapılandırmanın her iki yolunu da biliyorsunuz. Tamamlamak için, tekrar Next butonuna tıkladım ve bundan sonra konfigürasyon, running-config'e yüklendi.

Ve running-config çıktısı şöyledir:

```
!
interface Vlan1
 ip address 192.168.24.7 255.255.255.0
!
interface Vlan3
 ip address 192.168.25.7 255.255.255.0
!
interface Vlan10
 ip address 192.168.26.7 255.255.255.0
!
```

Oldukça basit ve güzel görünüyor. Tüm host/telefonlarımız şimdi VLAN'lar arasında serbestçe haberleşebilmelidir. Bununla beraber, tüm VLAN'larınız arasında yönlendirme yapmanız gerektiğini söylemediğimi anlamanızı istiyorum. Fakat ispatlamak için bu yapılandırma çok iyi olacaktadır!




Modülü tamamlamadan önce, 2960'a bağlanmak ve telefonumun bağlı olduğu portu yapılandırmak için bir Smartports makro kullanmak istedim. Şimdi, 3560'daki routing yapılandırması ve CLI kullanmanın onu nasıl kolaylaştırdığı hakkında söylediğimin tersine, CLI yardımıyla telefon konfigürasyonu, Godzilla'dır. Öyleyse, gelin bunu kolay yolla yapalım.

2960 (S1) için CNA'yi açtım ve Smartports seçeneğine tıkladım. Sonra port 4'ü işaretledim, sağtuş yaptım ve IP Phone+Desktop'ı seçtim.

Sonra Access VLAN'da, PC'min kullandığı VLAN 3'ü ve daha önce oluşturduğum voice VLAN'ı seçtim. OK butonuna tıklayınca makro çalıştı ve konfigürasyon running-config'e yüklendi. Port 4'te cisco-telefon macro'su çalıştıktan sonraki running-config çıktısı şöyledir:

```
!
interface FastEthernet0/4
 switchport access vlan 3
```

```
 switchport mode access
 switchport voice vlan 10
 switchport port-security maximum 2
 switchport port-security
 switchport port-security aging time 2
 switchport port-security violation restrict
 switchport port-security aging type inactivity
 srr-queue bandwidth share 10 10 60 20
 srr-queue bandwidth shape  10  0  0  0
 mls qos trust device cisco-phone
 mls qos trust cos
 macro description cisco-phone
 auto qos voip cisco-phone
 spanning-tree portfast
 spanning-tree bpduguard enable
!
```

Desktop macro'su, router interface'ine birçok konfigürasyon eklemiş gibi geldi bana! Running-config'e 44 satırlı bir kuyruk eklemiş. Şayet biri konfigürasyonlarınızı görür ve CNA kullandığınızı bilmezse, kesinlikle dahi gibi görüneceksiniz.

Şimdi, hem PC türü cihaza hem de telefona aynı porttan bağlanabiliyorum. Onlar, ayrı VLAN'larda çalışacaktır.

Artık bu bölüm tamamlandı. En az 2960 switch'lerle pratik tecrübesi kazanmanız gerektiği konusunda sizi fazla zorlayamam. Switch'leri, hem CLI hem de CNA kullanarak yapılandırabilmelisiniz.

# Özet

Bu bölüm sizi virtual LAN dünyasıyla tanıştırdı ve Cisco switch'lerin onları nasıl kullanabildiğini açıkladı. Bir switch ağ topluluğunda, katman2 switch'lerin sadece collision domain'leri ayırması ve varsayılan olarak switch'lerin büyük bir broadcast domain'den oluşmasından dolayı, çok önemli ve gerekli olan VLAN'ların broadcast domain'lerini nasıl ayırdığından bahsettik. Ayrıca size Access linklerden bahsettim ve bir FastEthernet linkte trunk VLAN'ların nasıl çalıştığını inceledim.

Trunking, ayrı VLAN'larda çalışan birçok switch'ten oluşan bir ağ ile uğraştığınızda, iyi anlaşılması gereken bir teknolojidir. Ayrıca VLAN Trunk Protocol (VTP)'den detaylı bir şekilde bahsettim. Onun VLAN bilgisini trunk linke gönderdiğini, fakat kendi trunk yapılandırmasının, VTP'nin parçası olmadığını öğrendiniz.

Ve sonra, unutmak isteyebileceğiniz telephony konusu vardı. Fakat başarılı olmak istiyorsanız, tamamını tekrarlamanız gerekse bile, onu iyi öğrendiğinizden emin olmalısınız.

Modülü, hem VTP, trunking ve VLAN konfigürasyonu ile hata giderme hem de CNA kullanarak 2960 ve 3560 switch'lerin nasıl yapılandırıldığıyla ilgili örnekler vererek bitirdim.

# Sınav Gereklilikleri

**Frame etiketleme terimini anlamak:** Frame etiketleme, VLAN tanımlanmasına işaret eder. Bunu, bir switch fabric boyunca dolaşan tüm frame'leri izlemek için switch'ler kullanır. Switch'lerin, hangi frame'in hangi VLAN'a ait olduğunu tespit etmesidir.

**ISL VLAN belirleme yöntemini anlamak:** Inter-Switch Link (ISL), bir Ethernet frame'indeki VLAN bilgisini detaylı bir şekilde etiketleme yöntemidir. Bu etiketleme bilgisi VLAN'ların, switch'in link boyunca bir frame'in VLAN üyeliğini belirlemesini sağlayan harici bir enkapsülasyon yöntemiyle trunk bir link boyunca çoğaltılmalarına izin verir. ISL, sadece Cisco switch ve router'larla kullanılabilen, Cisco tescilli bir frame etiketleme yöntemidir.

**802.1Q VLAN belirleme yöntemini anlamak:** Bu, frame etiketlemenin tescilli olmayan bir IEEE yöntemidir. Şayet, bir Cisco switch linki ile farklı marka bir switch arasında trunk yapıyorsanız. Trunk'ın çalışması için 802.1Q kullanmalısınız.

**Bir 2960 switch'teki bir trunk portun nasıl ayarlanacağını hatırlamak:** Bir 2960'daki bir portta trunk'ı ayarlamak için `switchport mode trunk` komutunu kullanın.

**Yeni bir host'a bağlandığında, bir switch portunun VLAN atamasını kontrol etmeyi hatırlamak:** Şayet yeni bir host'u switch'e bağlarsanız, daha sonra bu portun VLAN üyeliğini doğrulamak zorundasınız. Şayet üyelik, host'un ihtiyacı olandan farklı olursa, host, workgroup server gibi, ihtiyacı olan ağ servislerine erişemeyecektir.

**VTP'nin amacını ve yapılandırmasını anlamak:** VTP, switch ağınız boyunca VLAN veritabanının yayılmasını sağlar. Tüm switch'lerin aynı VTP domain'inde olması gerekir.

**VLAN'lar arası iletişimi sağlamak için bir Cisco router on a stick'in nasıl oluşturulacağını hatırlamak:** VLAN'lar arası routing sağlamak için, bir Cisco Fast Ethernet veya Gigabit Ethernet interface'i kullanabilirsiniz. Router'a bağlı switch portu, bir trunk port olmalıdır. Bundan sonra, bağlanılan her VLAN için router portunda sanal interface'ler (subinterface'ler) oluşturmalısınız. Her VLAN'daki host, bu subinterface adresini, kendi varsayılan ağ geçidi olarak kullanacaktır.

# Yazılı Lab 9

Bu bölümde, aşağıdaki, soruların cevaplarını yazın:

1. Hangi VTP modu, sadece VLAN bilgisini kabul edebilir ve onu değiştiremez?
2. Hangi VLAN belirleme yöntemi, Cisco router'lara özgüdür?
3. VLAN'lar _____ domain'lerini ayırır.
4. Switch'ler, varsayılan olarak, sadece _____ domain'lerini ayırır.
5. Varsayılan VTP modu nedir?
6. Trunking ne sağlar?
7. Frame etiketleme nedir?
8. Doğru/Yanlış: ISL enkapsülasyonu, frame bir Access linkten iletildiğinde, frame'den atılmaktadır.
9. Hangi link tipi sadece bir VLAN'ın parçasıdır ve portun native VLAN'ı olarak belirtilmektedir?
10. Hangi Cisco etiketleme bilgisi türü, VLAN'ların, harici bir enkapsülasyon yöntemi ile trunk bir link boyunca çoklanmasına izin verir?

(Yazılı lab 9'un cevapları, bu bölüm için gözden geçirme sorularına cevaptan sonra bulunabilir.)

# Gözden Geçirme Soruları

*Aşağıdaki sorular, bu modülün materyallerini anladığınızı test etmek için tasarlanmıştır. İlave sorulara nasıl sahip olacağınızla ilgili daha fazla bilgi için kitabın Giriş bölümüne bakın.*

NOT

1. VLAN'lar hakkında aşağıdakilerden hangisi doğrudur?

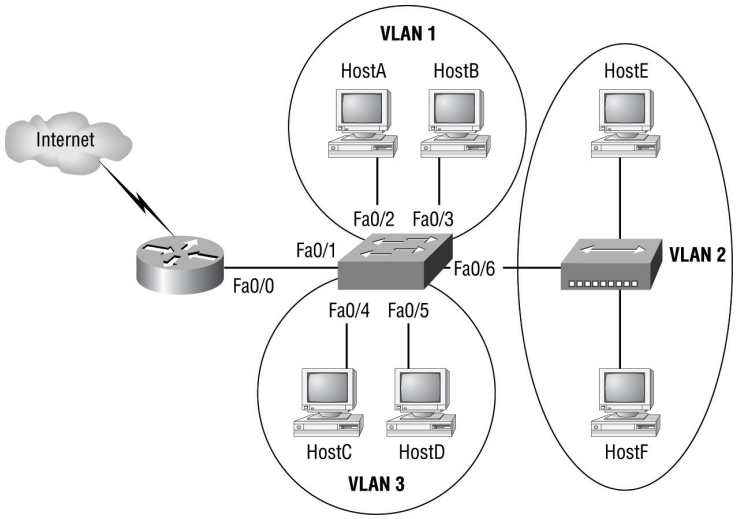
   A. Her Cisco switch ağında tanımlı en az iki VLAN'a sahip olmalısınız.

   B. Tüm VLAN'lar en hızlı switch'te yapılandırılır ve varsayılan olarak bu bilgiler, diğer tüm switch'lere dağıtılır.

   C. Aynı VTP domain'inde 10'dan fazla switch'e sahip olmamalısınız.

   D. VTP, VLAN bilgisinin, yapılandırılmış bir VTP domain'indeki switch'lere gönderilmesi için kullanılmaktadır.

2. Aşağıdaki diyagrama göre, topolojide gösterilen router ve switch port konfigürasyonunu hangisi açıklamaktadır?




   A. Router WAN portu, bir trunk port olarak yapılandırılır.

   B. Switch'e bağlı router portu, subinterface'ler kullanılarak yapılandırılır.

   C. Switch'e bağlı router portu, 10 Mbps hızda yapılandırılır.

   D. Hub'a bağlı switch portu, full duplex olarak yapılandırılır.

   E. Switch portlarına bağlı host'lar, Access portu olarak yapılandırılırlar.

3. Bir switch, üç farklı VLAN için yapılandırılmıştır: VLAN 2, VLAN 3 ve VLAN 4. VLAN'lar arası iletişimi sağlamak için bir router eklenmiştir. Router ve switch arasında sadece bir bağlantı yapılırsa, router'da hangi tip interface gerekmektedir?

   A. 10Mbps Ethernet

   B. 56Kbps Serial

   C. 100Mbps Ethernet

   D. 1Gbps Ethernet

4. Ağ performansını, hostlar için bant genişliğini artırmak ve broadcast domain boyutunu küçültürken sayısını çoğaltmak istiyorsunuz. Bunu aşağıdaki seçeneklerden hangisi gerçekleştirir?

   A. Yönetilebilir hub'lar

   B. Bridge'ler

C. Switch'ler

D. VLAN'larla yapılandırılmış switch'ler

5. Aşağıdaki protokollerden hangisi, bir switch'teki trunking'i yapılandırmak için kullanılmaktadır? (İki şık seçin)

   A. VLAN Trunk Protokol

   B. VLAN

   C. 802.1Q

   D. ISL

6. IOS tabanlı bir switch'te, yeni bir trunk link yapılandırıldığında, link boyunca hangi VLAN'lar kabul edilecektir?

   A. Varsayılan olarak tüm VLAN'ların trunk'ta olmasına izin verilir.

   B. Hiçbir VLAN'a izin verilmez, VLAN'lar elle yapılandırılmalıdır.

   C. Sadece yapılandırılmış VLAN'lara linkte izin verilir.

   D. Varsayılan olarak, sadece genişletilmiş VLAN'lara izin verilir.

7. Hangi switch teknolojisi, bir broadcast domain'in boyutunu düşürür?

   A. ISL

   B. 802.1Q

   C. VLAN'lar

   D. STP

8. Hangi VTP modu, switch'teki VLAN bilgisini değiştirmenize izin verir?

   A. Client

   B. STP

   C. Server

   D. Transparent

9. Hangi komut, VLAN üyelik bilgisini Ethernet frame'lerine yerleştirmede, IEEE standart yöntemini kullanmak için bir switch portunu yapılandıracaktır?

   A. `Switch(config)#switchport trunk encapsulation isl`

   B. `Switch(config)#switchport trunk encapsulation ietf`

   C. `Switch(config)#switchport trunk encapsulation dot1q`

   D. `Switch(config-if)#switchport trunk encapsulation isl`

   E. `Switch(config-if)#switchport trunk encapsulation ietf`

   F. `Switch(config-if)#switchport trunk encapsulation dot1q`

10. VTP hakkında aşağıdakilerden hangisi doğrudur?

    A. Tüm switch'ler varsayılan olarak VTP server'dır.

    B. Tüm switch'ler varsayılan olarak VTP transparent'dir.

    C. VTP, varsayılanda, tüm Cisco switch'lerde Cisco domain ismine sahiptir.

    D. Tüm switch'ler varsayılan olarak VTP client'tır.

11. Hangi protokol, yeni bir VLAN'ın bir domain'deki tüm switch'lere dağıtılması için yapılandırılmasına izin vererek, bir switch ağında yönetimsel yükü azaltır?

    A. STP

    B. VTP

    C. DHCP

    D. ISL

12. Aşağıdaki komutlardan hangisi, bir 2960 switch'deki trunk portu ayarlar?

    A. Trunk on

    B. Trunk all

    C. Switchport trunk on

    D. Switchport mode trunk

13. Aşağıdakilerden hangisi, frame etiketleme için bir IEEE standardıdır?

    A. ISL

    B. 802.1Z

    C. 802.1Q

    D. 802.3U

14. Bir hostu, switch portuna bağlıyorsunuz, fakat yeni host, aynı switch'e bağlı sunucuya bağlanamamaktadır. Problem ne olabilir? (En olası cevabı seçin.)

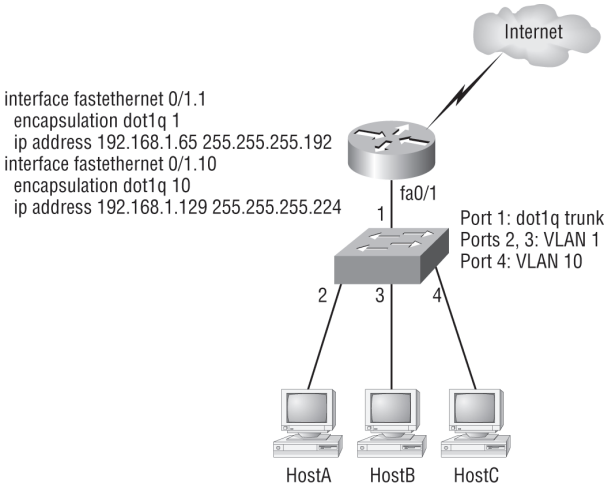
    A. Router yeni bir host için yapılandırılmamaktadır.

    B. Switch'teki VTP konfigürasyonu, yeni host için güncellenmemektedir.

    C. Host, geçersiz bir MAC adresine sahiptir.

    D. Host'un bağlı olduğu switch portu, doğru VLAN üyeliğiyle yapılandırılmamıştır.

15. Diyagrama göre, IEEE frame etiketleme versiyonu kullanarak router'ın Fast Ethernet interface'i ile bir link kurmak için hangi üç komut kullanılabilmektedir? (Üç şık seçin)




    A. `Switch(config)#interface fastethernet 0/1`

    B. `Switch(config-if)#switchport mode access`

    C. `Switch(config-if)#switchport mode trunk`

    D. `Switch(config-if)#switchport access vlan 1`

E. `Switch(config-if)#switchport trunk encapsulation isl`

F. `Switch(config-if)#switchport trunk encapsulation dot1q`

16. Şu iki switch, VLAN bilgisini paylaşmamaktadır. Aşağıdaki çıktıdan, bu switch'lerin VTP mesajlarını paylaşmamalarının sebebi nedir?

```
SwitchA#sh vtp status
VTP Version                     : 2
Configuration Revision          : 0
Maximum VLANs supported locally : 64
Number of existing VLANs        : 7
VTP Operating Mode              : Server
VTP Domain Name                 : RouterSim
VTP Pruning Mode                : Disabled

SwitchB#sh vtp status
VTP Version                     : 2
Configuration Revision          : 1
Maximum VLANs supported locally : 64
Number of existing VLANs        : 7
VTP Operating Mode              : Server
VTP Domain Name                 : GlobalNet
VTP Pruning Mode                : Disabled
```

A. Switch'lerden birinin, VTP version1'e ayarlanması gerekir.

B. Her iki switch, VTP server olarak ayarlanmıştır. Birinin client olarak ayarlanması gerekir.

C. VTP domain isimleri doğru bir şekilde ayarlanmamıştır.

D. VTP pruning etkin değildir.

17. Aşağıdakilerden hangisi, VLAN'lar arası iletişim sağlamaktadır? (İki şık seçin.)

A. ISL

B. VTP

C. 802.1Q

D. 802.3.Z

18. İki switch arasındaki VLAN bilgisini aktarmak için VLAN trunking protokolünü yapılandırmak için, olması gereken iki şey nedir? (İki şık seçin)

A. Trunk linklerin her iki ucu da, IEEE 802.1e encapsulation olarak ayarlanmalıdır.

B. Her iki switch'in VTP domain ismi aynı olmalıdır.

C. Her iki switch'teki tüm portlar, Access port olarak ayarlanmalıdır.

D. İki switch'ten birisi, VTP server olarak ayarlanmalıdır.

E. İki switch'i birbirine bağlamak için bir rollover kablo kullanılmalıdır.

F. VLAN'lar arasındaki VTP trafiğini iletmek için bir router kullanılmalıdır.

19. Aşağıdakilerden hangileri VLAN'ların faydalarıdır? (Üç şık seçin)
    A. Collision domain'lerin boyutunu arttırırlar.
    B. Kullanıcıların fonksiyonlarına göre mantıksal gruplanmasına izin verirler.
    C. Ağ güvenliğini arttırabilirler.
    D. Collision domain'lerin sayılarını azaltırken, broadcast domain'lerin boyutunu arttırırlar.
    E. Switch yönetimini kolaylaştırırlar.
    F. Broadcast domain'lerin boyutunu azaltırken, broadcast domain'lerin sayısını arttırırlar.
20. Bir switch portu, trunk VLAN olarak kullanıldığında, aşağıdaki modlardan hangisi geçerlidir? (Üç şık seçin)
    A. Blocking
    B. Dynamic auto
    C. Dynamic desirable
    D. Nonegotiate
    E. Access
    F. LearningG.

# Gözden Geçirme Sorularının Cevapları

1. Switch'ler VLAN bilgisini varsayılan olarak dağıtmazlar. VTP domain yapılandırmalısınız. VLAN Trunking Protocol (VTP), trunk link boyunca VLAN bilgisini dağıtmak için kullanılmaktadır.
2. B,E,F Bir switch'e bağlı, VLAN'lar arası iletişim sağlayan bir router, subinterface'ler kullanması için yapılandırılır. Router'a bağlı switch portu, ISL veya 802.1Q kullanmalıdır ve host'lar, Access port olarak bağlanmalıdır. Bu, switch portlarda varsayılan ayardır.
3. C 100Mbps veya 1Gbps Ethernet kullanabildikleri halde, en az 100Mbps gerekmektedir ve en iyi cevap budur. Switch'ten router'a olan linki, VLAN'lar arası iletişimde çalışır hale getirmek için trunk yapmanız gerekmektedir.
4. D Switch ağınızda VLAN'ları oluşturup, çalıştırarak, katman2'deki broadcast domain'leri ayırabilirsiniz. Farklı VLAN'lardaki host'ları haberleştirmek için bir router veya katman3 switch'e sahip olmalısınız.
5. C,D Cisco, ISL denilen tescilli bir protokole sahiptir. IEEE versiyonu, 802.1Q'dur.
6. A Varsayılan olarak tüm VLAN'ların trunk'ta olmasına izin verilir ve trunk linkte dolaşmasını istemediğiniz VLAN'ları elle kaldırmalısınız.
7. C Virtual LAN'lar, katman2 switch ağ topluluklarında, broadcast domain'leri ayırırlar.
8. C Bir switch üzerindeki VTP bilgisini, sadece server modda değiştirebilirsiniz.
9. F Şayet bir 2950 switch'teyseniz, 2950 sadece IEEE 802.1Q versiyonu kullanabildiğinden, komut sadece `switchport mode trunk`'tır. Bununla beraber, bir 3550, hem ISL hem de 802.1Q çalışabilir. Bu nedenle, `encapsulation` komutunu kullanmak zorundasınız. Trunking protokol olarak 802.1Q seçmek için değişken, `dot1q`'dur.
10. A Tüm Cisco switch'ler, varsayılan olarak VTP moddadır. Bir Cisco switch'te varsayılan olarak başka VTP bilgisi yapılandırılmaz. Aynı domain'de olmaları için tüm switch'lerde domain ismini aynı ayarlamalısınız. Yoksa VTP veritabanını paylaşmayacaklardır.
11. B Virtual Trunk Protocol (VTP), switch ağındaki tüm switch'lere VLAN veritabanını iletmek için kullanılır. Üç VTP modu, server, client ve transparent'dır.
12. D Bir switch portunu, tüm VLAN bilgisinin linke geçmesini sağlayan trunk moda ayarlamak için, `switchport mode trunk` komutunu kullanın.
13. C 802.1Q, farklı switch'ler arasında trunk linkler oluşturulmasına izin vermek için geliştirilmiştir.
14. D Bu soruda biraz belirsizlik vardır. Fakat en iyi cevap, port için VLAN üyeliğinin, yapılandırılmamasıdır.
15. A,C,F Bir switch portunda trunk linki oluşturmak için, ilk olarak interface'e gitmeniz gerekir (bu soruda Fast Ethernet 0/1'dir). Sonra, trunking komutu olarak, 2950 için `switchport mode trunk` (IEEE 802.1Q, 2960 switch'lerin tek çalıştığı versiyondur) veya bir 3560 switch için, `switchport trunk encapsulation dot1q` komutunu seçersiniz.
16. C Switch'lerden birini client olarak ayarlayabileceğiniz halde, bu onların, VTP ile VLAN bilgilerini paylaşmalarına engel olmayacaktır. Ancak, domain isimleri aynı ayarlanmazsa, VTP ile VLAN bilgilerini paylaşmayacaklardır.
17. A,C ISL, Cisco tescilli bir frame etiketleme yöntemidir. IEEE 802.1Q, frame etiketlemenin CISCO tescilli olmayan versiyonudur.
18. B,D Switch'ler arasında VLAN bilgisini paylaşmak için tüm switch'lerde aynı domain ismine sahip olmalısınız. En az switch'lerden birinin VTP server olması gerekir. Diğer switch'ler, VTP client olarak ayarlamalıdır.

19. B,C,F VLAN'lar broadcast domain'lerini ayırırlar. Yani daha küçük broadcast domain'leri oluş-tururlar. Fiziksel lokasyon yerine mantıksal fonksiyonla yapılandırmaya izin verirler ve doğru yapılandırılırlarsa, bazı güvenlik özellikleri sağlayabilirler.
20. B,C,D Bir switch'deki geçerli VLAN trunk modları, dynamic auto, dynamic desirable, trunk(on) ve nonegotiate'dir.21.

# Yazılı Lab 9'un Cevapları

1. Client
2. Inter-Switch Link (ISL)
3. Broadcast
4. Collision
5. Server
6. Trunking, aynı anda bir çok VLAN'ı tek bir portun parçası yapmanıza izin verir.
7. Frame belirleme (frame etiketleme), her frame'e benzersiz olarak kullanıcı tanımlı bir ID atar. Bu bazen, VLAN ID veya renk olarak belirtilir.
8. Doğru
9. Access link
10. Inter-Switch Link (ISL)

# 10 Güvenlik

# 10 Güvenlik

- Perimeter, Firewall ve Internal Router'lar
- Güvenlik Tehditlerinin Farkına Varmak
- Güvenlik Tehditlerinin Azaltılması
- Acces List'lere Giriş
- Standart Access List'ler
- Extended Access List'ler
- Gelişmiş Access List'ler
- Access List'lerin görüntülenmesi
- SDM kullanarak Access List'leri Yapılandırmak
- Özet
- Sınav Gereklilikleri
- Yazılı Lab 10.1
- Pratik Lab'lar
- Gözden Geçirme Soruları
- Gözden Geçirme Sorularının Cevapları
- Yazılı Lab 10.1'i Cevapları

# Güvenlik

Şayet bir sistem yöneticisiyseniz benim tahminim, kötü niyetli kullanımdan önemli veri ve ağ kaynaklarınızı koruma duyarlılığınız, sizin önceliğiniz olacaktır. Doğru sayfada olduğunuzu bilmek iyidir, Cisco, bunu yapmanız için gereken araçlarla donanımınızı sağlayacak gerçekten etkili güvenlik çözümlerine sahiptir.

Aslında, verilerinizi ve ağınızı korumak, bu bölümün odak noktası olacaktır. Saldırıların birçoğuna karşı, güçlü, entegre tespit yöntemleri öneren Cisco router ve IOS firewall'lar ile ağınızın güvenliği için en yaygın tehditleri bertaraf etmek ile ilgili çok şey öğreneceksiniz. Cisco IOS Firewall'un, hem iç hem de dış ağ kurulum ihtiyaçlarınız için gerçek güvenlik politikalarını nasıl sağladığını göstereceğim. Ayrıca, uzak lokasyonlarınıza güvenli bağlantıların nasıl kurulacağını göstereceğim.

Access control list'ler (ACL) vazgeçilmez ağ uygulamaları olduklarından, Cisco güvenlik çözümünün ayrılmaz bir parçasıdır. ACL'ler, verimliliğe ve ağınızın işleyişine güçlü bir şekilde yardımda bulunarak, ağ yöneticilerine şirket boyunca trafik akışı üzerinde güçlü bir kontrol olanağı sağlar. Access list'lerle, yöneticiler, akış ve güvenlik politikaları uygulanabilen çözümler üzerinde basit istatistikler toplayabilirler. Ayrıca hassas cihazlar, yetkisiz erişimlerden korunabilmektedir.

Bu bölümde, hem TCP/IP için access list'leri ve katman2 switch'teki MAC access list'leri tartışacağız hem de uygulanan access list'lerin işlevselliğini görüntülemek ve test etmek için uygun araçları göreceğiz.

Cisco IOS firewall ve CLI kullanarak access list'leri yapılandırmayı gördükten sonra, Cisco Security Device Manager (SDM) kullanarak onu yapmanın ne kadar kolay olacağını göstereceğim.

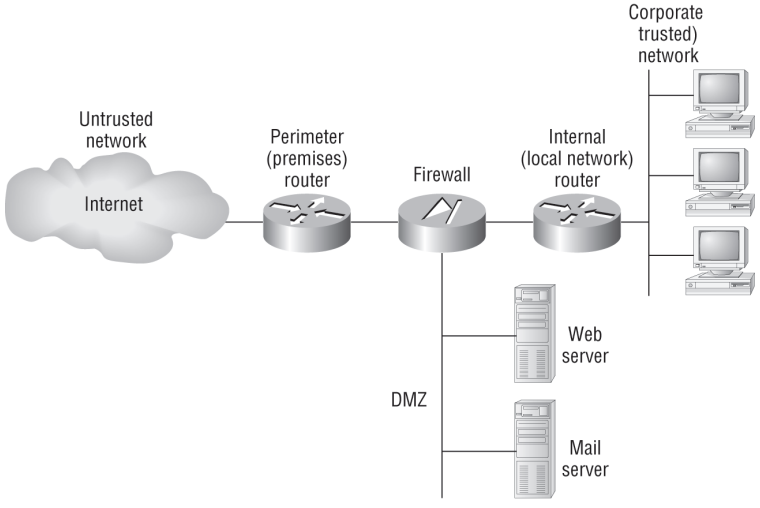
*Bu bölüm ile ilgili son güncellemeler için www.lammle.com ve/veya www.sybex.com adresine bakınız.*

**NOT**

Virtual Private Network'ler (VPN) firma güvenliğinizin en önemli parçalarından olduğu halde, VPN'leri"Wide Area Network" başlıklı bölüm 14'de işleyeceğim.

## Perimeter, Firewall ve Internal Router'lar

Güvenlik için değişik stratejilerin, internal ve Perimeter router'lar ve firewall cihazları çözümüne dayalı olduğunu, tipik olarak, orta ve geniş ölçekli şirket ağlarında çok sık görürsünüz. Internal router'lar, korumalı şirket ağının değişik bölümlerindeki trafiği göstererek, network için ilave güvenlik sağlarlar ve bunu access list'ler kullanarak yapar. Bu tip cihazların nerede bulunduğunu, Şekil 10.1'de görebilirsiniz.




***Şekil 10.1:*** *Tipik bir güvenli network.*

Trusted network ve untrusted network terimlerini, bu bölüm ve bölüm 11 ”Network Address Translation” boyunca kullanacağım. Bu nedenle, tipik güvenli bir ağda onların nerede bulunacağını anlamanız önemlidir. Demilitarized zone (DMZ), firewall'unuzu nasıl yapılandırdığınıza bağlı olarak, global (gerçek) internet veya özel adreslere sahip olabilir. Fakat tipik olarak, HTTP, DNS, e-mail ve diğer internet-tipi şirket sunucularının olduğu yerdir.

Router'a sahip olmak yerine, güvenilir ağların iç tarafındaki switch'lerle virtual local area network'leri (VLAN) kullanabilirsiniz. Kendi güvenlik özelliklerini içeren çok katmanlı switch'ler, VLAN mimarisinde daha yüksek performans sağlamak için, bazen dâhili (LAN) router'ların yerini almaktadır.

Gelin, tipik güvenli bir ağ topluluğu için güvenlik tehditlerini tartışalım, daha sonra,

Cisco IOS Firewall ayarları ve access list'lerini kullanarak ağ topluluğunu korumanın bazı yollarını göstereceğim.

# Güvenlik Tehditlerinin Farkına Varmak

Evet, doğru: Güvenlik atakları onların karmaşıklığı, tehdit seviyelerine ve hatta bazen WUI (witless user ignorance-bilgisiz kullanıcı hataları) olmasına bağlı olarak oldukça değişmektedir. (Bu terim bir sınav konusu değildir, fakat düşündüğünüzden çok olur!)

Görüldüğü gibi her şey planlamanın yapılması veya yapılmaması ile ilgili. Basit olarak, internetin bir gün ne kadar gerekli bir araç olacağını, onu ilk oluşturanlar bile kesinlikle öngörmemişlerdir.

Bu güvenliğin bir sorun olmasının ana sebebidir. Çoğu IP uygulaması, doğası itibarıyla güvensizdir. Yinede endişelenmeyin, çünkü Cisco, bununla mücadele için birçok özelliğe sahiptir. Fakat ilk olarak, bazı yaygın atak profillerini inceleyelim:

**Application-katman atakları:** Bu ataklar, tipik olarak dikkatlerini, sunucularda çalışan yazılımdaki, iyi-bilinen açıklara çevirirler. En gözde hedefler FTP, sendmail ve HTTP'dir. Bu servislere verilen izinlerin, çoğu zaman yetkili olmasından dolayı, kötü adamlar, kolayca yetkiyi ele geçirip erişirler ve belirttiğim uygulamaların çalıştığı makineleri kendi amaçları için kullanırlar.

**Autorooter'lar:** Bunları, hacker'ların bir tür robotları olarak düşünebilirsiniz. Kötü adamlar, tüm sistemin izlenmesini sağlayacak, stratejik konumdaki bilgisayardaki veriyi araştırıp, gözden geçirmek ve sonrada ele geçirip yönetmek için rootkit denilen araçlar kullanırlar.

**Backdoor'lar:** Bunlar basit olarak, bir bilgisayar veya ağa kılavuzluk yapan yollardır. Kötü adamlar, basit saldırılarla veya daha gelişmiş Truva atı kodları vasıtasıyla, siz onları algılayıp durdurana kadar, istedikleri zaman, belirli bir host veya ağa gizlice erişim sağlayabilirler.

**Denial of Service (DoS) ve Distributed Denial of Service (DDoS) atakları:** Bunlar oldukça kötüdür ve kurtulması da zordur. Fakat hacker'ların onları çalıştıran diğer hacker'lara bile saygısı yoktur. Çok kötü olmasına rağmen, gerçekleştirilmesi oldukça kolaydır. (Yani 10 yaşındaki çocuk sizi küçük düşürebilir) Basit olarak, sistemi boğarak, bir servisi kullanılamaz hale getirir. Farklı bazı türleri vardır:

**TCP SYN flood:** Bir istemci, görünüşte olağan görünen bir TCP bağlantısı başlattığında sunucuya bir SYN mesajı gönderir. Sunucu, istemci makineye bir SYN-ACK mesajı göndererek cevap verir. Sonra istemci, bir ACK mesajı dönerek, bağlantıyı kurar. Kulağa hoş geliyor, fakat bağlantı sadece tek yönlü açık olduğunda, bu proses esnasında kurban makine, yarı-açık bağlantıların saldırısına uğrar ve felç olur.

**“ping of death” atakları:** Muhtemelen, TCP/IP'nin maksimum paket boyutunun 65,535 oktet olduğunu biliyorsunuzdur. Bilmiyorsanız da önemli değil, sadece bunun, normalden büyük paketlerle ping atarak çalıştırılan atak olduğunu bilin yeter. Bir cihazın, sürekli reboot edilmesine, donmasına veya tamamıyla çökmesine neden olurlar.

**Tribe Flood Network (TFN) ve Tribe Flood Network 2000 (TFN2K):** Bu çirkin küçük saldırılar, birçok noktadan senkronize DoS atakları başlattıklarında çok karmaşıktırlar ve birçok cihazı hedef alabilirler. Bunu kısmen, en kısa sürede açıklayacağım "IP spoofing" olarak bilinen yöntemle yaparlar.

**Stacheldraht:** Bu atak aslında farklı yöntemlerin karışımıdır ve dikenli tel anlamında Almanca kökenli bir terimdir. Esasında TFN ile birleşirler ve çok küçük kodlar eklerler. En üst seviyeden bir saldırı başlatır ve sonunda, DoS atağıyla devam ederler.

**IP spoofing:** Ağınızın içinden veya dışından bir kötü adam, iki yöntmden birini kullanarak, güvenilen bir host makinesi olarak davranır: Ağınızın güvenilir adres aralığından bir IP adresi ile görünmek veya onaylı, güvenilir bir harici IP adresi kullanmak. Hacker'ın gerçek kimliği, aldatılan adresin arkasına gizlenmiştir. Bu problemlerinizin başlangıcıdır.

**Man-in-the-middle atakları:** Durdurmak! Fakat futbolda değil, ağınızın çok değerli veri paketlerinin bir kısmını. Genelde suçlu kişi, sniffer olarak bilinen bir aracı kullanır, (daha sonra ele alınacak) yönlendirme ve transport protokollleri ile artar.

**Network reconnaissance:** Bir ağa sızmadan önce hacker'lar çoğu kez ağa dair alabilecekleri tüm bilgiyi toplarlar. Ağ hakkında ne kadar çok bilgiye sahip olurlarsa, onu o kadar tehlikeye atabilirler. Port taraması, DNS sorguları ve ping taraması gibi yöntemler yardımıyla amaçlarına ulaşırlar.

**Packet sniffers:** Bu daha önce bahsettiğim bir araçtır. Fakat onun ne olduğunu söylemedim ve aslında bir yazılım olması sizi şaşırtabilir. Şöyle çalışır; bir network adapter kartı, rastgele bir moda ayarlanır. Böylece bir uygulama ile ağın fiziksel katmanından yakalanan tüm paketler istenen hedefe yönlendirilebilecektir. Bir paket sniffer, şifre, kullanıcı adlarını içeren çok önemli, hassas bilgileri çalabilir. Bu sebep ile kimlik hırsızları tarafından çok kullanılan bir araçtır.

**Password atakları:** Bu yöntem çok sayıda çeşitde sahiptir ve IP spoofing, paket sniffing ve Truva atları gibi daha karmaşık atak tipleri yardımıyla başarılabilmelerine rağmen onların tek amacı, kullanıcı şifrelerini bulmaktır, böylece hırsızlar geçerli bir kullanıcı gibi davranır ve sonra kullanıcının yetki ve kaynaklarına erişebilirler.

**Brute force atağı:** Hedef bir ağa yazılım-tabanlı ataklar yaparak, sunucu gibi paylaşılan bazı network kaynaklarına bağlanmayı dener. Hacker için, birçok hakka sahip erişim hesabına sahip olmak mükemmeldir. Çünkü artık kötü adamlar, daha sonra erişim sağlamada kullanmak için backdoor'ları şekillendirebilir ve şifre gereksinimini tamamıyla bypass edebilirler.

**Port redirection atakları:** Bu yaklaşım, hacker'ın zorla gireceği bir host makinesi gerektirir ve bir firewall boyunca (normalde geçişine izin verilmeyen) güvenilmez trafiği sağlamak için kullanılır.

**Truva atı (Trojan) atakları ve virüsler:** Bu ikisi oldukça benzerdir. Truva atı ve virüsler kötü niyetli kodlar olarak kullanıcı makinelerine bulaşırlar ve çeşitli derecelerde makinaya zarar verirler. Fakat bazı farklılıklar gösterirler. Virüsler, Windows sistemleri için ana yorumlayıcı olan, command.com gibi dosyalarına eklenebilen kötü niyetli programlardır. Virüsler daha sonra, makinedeki diğer dosyalara da bulaşır veya dosyaları silerler. Truva atı ve virüsler arasındaki fark, Trojan'lerin, kendilerini, gerçekte olduklarından farklı bir şey gibi görünmelerini sağlayan (örneğin basit, suçsuz bir oyun gibi), içlerine kod gizlenmiş komple uygulamalar olmasıdır.

**Güven sömürü atakları:** Birinin, ağınızdaki güven ilişkisini suiistimal ettiği zaman oluşan ataklardır. Örneğin, bir şirketin perimeter ağ bağlantısı genelde, SMTP, DNS ve HTTP gibi önemli servisleri tutar. Aynı segment'te olmaları, sunucuları, saldırıya açık hale getirir.

Dürüst olmak gerekirse, sadece bu kitabın kapsamı dışına çıkmasından değil, ayrıca size öğreteceğim yöntemlerin sizi genel olarak ataklardan tamamıyla koruyacak olmasından dolayı, şimdi bahsettiğim güvenlik tehditlerinin detaylarına girmeyeceğim. Tehditlerin azaltılması için yeterli ipuçlarını öğreneceksiniz. Hemen hemen, birçok kararlı kötü adam, sizi bırakacak ve daha kolay

av arayacaklardır. Basit olarak, bunu, güvenli ağ kurulumunun nasıl uygulanacağını gösteren bir bölüm olarak düşünün.

# Güvenlik Tehditlerinin Azaltılması

Güvenlik tehditlerini azaltmak için hangi çözümleri kullanmalıyız? Juniper'dan McAfee veya diğer firewall ürünlerini mi? Muhtemelen Cisco'dan bir şeyler kullanmalıyız. Cisco, Adaptive Security Appliance (ASA) denilen, oldukça güzel bir ürüne sahiptir. ASA (intrusion prevention gibi) seçtiğiniz modüllere bağlı olarak fiyatı değişen oldukça pahalı bir üründür. ASA bu kitabın hedefi dışındadır. Kişisel olarak ASA'nın piyasadaki en iyi ürün olduğunu düşünüyorum.

Cisco IOS yazılımları, internet backbone router'larının %80'ninden fazlasında çalışır. Muhtemelen, network altyapısının en kritik bölümleridirler. Gelin Cisco IOS Firewall feature set olarak bilinen, Cisco IOS'un yazılım-tabanlı güvenliğini, uçtan-uca İnternet, intranet ve uzak-erişim ağ güvenliği çözümlerimiz için kullanalım. Cisco ACL'ler en yaygın tehditlerin çoğunun azaltılması için oldukça etkili araçlar olduğundan onları kullanmak oldukça iyi bir seçimdir. Ve CCNA sınavınıza çalışıyorsanız, ACL'lerin nasıl çalıştığını bu bölümdeki her şeyden daha iyi anlamanız gerekir.

## Cisco IOS Firewall

Burası, bu Cisco IOS Firewall özelliklerini kullanarak, size daha önce verdiğim listedeki yaygın güvenlik tehditlerinin bazılarının nasıl engelleneceğinin anlaşılacağı yerdir:

**Stateful IOS Firewall kontrol motoru:** Dahili kullanıcılarınıza uygulama bazlı güvenli erişim kontrolü sağladığından, bu sizin perimeter koruma özelliğinizdir. İnsanlar genelde onu Context-Based Access Control (CBAC) olarak belirtirler.

**Intrusion detection:** En yaygın atak ve intrusion detection imzalarının 102 tanesini referans alarak gerçek zamanlı suiistimalleri görüntüleme, durdurma ve tepki göstermenize izin veren, detaylı bir paket inceleme aracıdır.

**Firewall voice traversal:** Hem arama akışı hem de uygun açık kanalları protokollerin anlaması temeline dayalı uygulama-seviyesi bir özelliktir. Hem H.323v2 hem de Session Initiation Protocol (SIP) voice protokollerini destekler.

**ICMP inspection:** Basit olarak diğer ICMP paketlerini reddederken, firewall'unuzdan gelen ping ve traceroute gibi ICMP paketlerine cevap verilmesine izin verir.

**Authentication Proxy:** HTTP, HTTPS, FTP ve Telnet yardımıyla, network kaynaklarına erişmek istedikleri zaman kullanıcılara kimlik doğrulaması yapan bir özelliktir. Kullanıcılar için kişisel network erişim profillerini tutar, bir RADIUS veya TACACS+ server'dan sizin için profilleri otomatik olarak alır ve uygular.

**Hedef URL policy yönetimi:** Genellikle URL Filtreleme olarak belirtilen özelliktir.

**Per-user firewall'lar:** Bunlar basit olarak, servis sağlayıcılar tarafından sağlanan, kişiselleştirilmiş, kullanıcıya özgü, indirilebilir firewall'lardır. Ayrıca, AAA sunucu profil deposu yardımıyla, kişiselleştirilmiş ACL'ler ve diğer ayarları da alabilirsiniz.

**Cisco IOS router ve firewall provisioning:** No-touch router tedariği, versiyon güncellemeleri ve güvenlik politikaları sağlar.

**Denial of service (DoS) tespiti ve önlemesi:** Paket başlıklarını kontrol eden ve şüpheli bulduğu paketleri engelleyen bir özelliktir.

**Dinamik port eşleştirmesi:** Standart olmayan portlarda, firewall'lar tarafından desteklenen uygulamalara izin veren bir adaptör türü.

**Java applet bloklama:** Sizi, tanınmayan Java applet'lerinden korur.

## Basit ve Gelişmiş Trafik Filtrelemesi

Standart, extended, hatta Cisco IOS Firewall ile Lock-and-Key trafik filtrelemesi gibi dinamik ACL'ler kullanabilirsiniz. ACL'leri istediğiniz network segmentine uygulamalısınız. Artı, herhangi bir segment boyunca geçmesine izin vermek istediğiniz trafik tipini belirtebilirsiniz.

**Politika tabanlı, multi-interface desteği:** Güvenlik politikanıza bağlı olarak, interface ve IP adresiyle kullanıcı girişlerini kontrol etmenizi sağlar.

**Network Address Translation (NAT):** Dahili ağı dışarıdan gizleyerek, güvenliği arttırır. (NAT' tan modül11'de bahsedeceğim)

**Time-based access list'ler:** Güvenlik politikalarını, günün belirli bir zamanı ve haftanın belirli bir gününe dayanarak tanımlar.

**Peer router authentication:** Router'ların, güvenilir routing bilgilerini, gerçek, güvenli kaynaklardan almasını garanti eder.(Bunun çalışır olması için, RIPv2, EIGRP veya OSPF gibi kimlik doğrulamayı destekleyen bir routing protokolüne ihtiyacınız vardır)

Şimdi güvenlik tehditleri, Cisco IOS Firewall'un belirli özellikleri ve bu yazılımı kullanmanın size nasıl avantaj sağladığı ile ilgili özet bilgi aldınız. Gelin access list'lerin dünyasına derinlemesine girelim ve ACL'lerin güvenlik tehditlerini nasıl azalttığını öğrenelim. Onlar gerçekten çok güçlü araçlardır, bu nedenle dikkatinizi verin!

# Acces List'lere Giriş

Bir access list, aslında paketleri sınıflandıran bir koşul listesidir. Network trafiği üzerinde kontrol sağlamaya ihtiyacınız olduğunda, gerçekten faydalı olabilirler. Bir access list, bu durumlarda seçebileceğiniz bir araç olabilir.

Access list'lerin en yaygın ve kolay kullanımlarından biri, güvenlik politikaları uyguladığınızda, istenmeyen paketlerin filtrelenmesidir. Örneğin, trafik şablonlarını düzenlemeyle ilgili çok özel kararlar vermek için onları uygulayabilirsiniz. Böylece diğerlerini kısıtlarken, İnternet'teki web kaynaklarına sadece belirli kullanıcılara izin verirler. Access list'lerin doğru kombinasyonları ile network yöneticileri, oluşturdukları herhangi bir güvenlik politikasını mecbur kılacak bir güce sahip olurlar.

Access list'ler, paketlerin bloklanmasını gerektirmeyen durumlarda bile kullanılabilirler. Örnek olarak, onları, ağların, dinamik routing protokolleri tarafından yayınlanıp yayınlanmamasını kontrol etmek için kullanabilirsiniz. Access list'lerin yapılandırılmaları aynıdır. Farklılık, onları nasıl uygulayacağınızdadır. (Bir interface yerine bir routing protokolüne uygulamak.) Bu yolla bir access list uyguladığınızda, o bir distribute list olarak belirtilir ve routing yayınlarını durdurmaz, sadece içeriklerini kontrol eder. Access list'leri ayrıca, kuyruklama veya QoS tipi servisler için paketleri sınıflandırmak ve belirli trafik tiplerinin, pahalı bir ISDN linkini etkinleştirebilmesini kontrol etmek için kullanabilirsiniz.

Access list'leri oluşturmak, gerçekten, `if-then` döngülerinin serisi olan bir programlama gibidir. Şayet belirlenen bir koşul oluşursa, o zaman belirlenen bir eylem gerçekleşir. Şayet belirli koşul oluşmazsa, hiçbir şey olmaz ve sonraki ifade değerlendirilir. Access-list ifadeleri basit olarak, paketlerin karşılaştırıldığı, onlarla sınıflandırılan ve etkilediği paketleri filtreler. Listeler oluşturulunca, herhangi bir interface'de geliş veya gidiş yönünde uygulanabilirler. Bir access list uygulamak, router'ın belirli yöndeki interface'i geçen paketleri analiz etmesine ve uygun bir eylem gerçekleştirmesine sebep olur.

Bir paket, access list ile karşılaştırıldığında, paketin izlemesi gereken önemli bazı kurallar vardır:

- Daima sıralı access list'in her satırıyla karşılaştırılır. Her zaman access list'in ilk satırıyla başlar, daha sonra 2.ci satır, 3.cü satır şeklinde devam eder.

- Bir eşleşme olana kadar, access list'in satırlarıyla karşılaştırılır. Paket, access list'in bir satırındaki koşul ile eşleşince, paket ona göre davranır ve başka karşılaştırma olmaz.
- Her access list'in sonunda gizli bir "deny" vardır. Yani, bir paket, satırlardan herhangi birindeki koşulla eşleşmezse, engellenecektir.

IP paketlerini access list'lerle filtrelediğinizde, bu kuralların her birinin güçlü sonuçları vardır. Bu nedenle, etkili access list'ler oluşturmak, tecrübe gerektirmektedir.

İki access list tipi vardır:

**Standart acess list'ler:** Bunlar, koşul için sadece IP paketindeki kaynak IP adresini kullanırlar. Tüm kararlar, kaynak IP adresi bazında yapılmaktadır. Yani, standart access list'ler basit olarak protokol ailesinin tamamını kabul eder veya reddeder. Web, Telnet, UDP vs. gibi birçok trafik tipinin hiçbirini ayırmaz.

**Extended access list'ler:** Extended access list'ler, bir IP paketinin katman3 ve katman4 başlığındaki protokol alanlarını değerlendirebilir. Onlar, kaynak ve hedef IP adreslerini, network katmanı başlığındaki protokol alanını ve transport katman başlığındaki port numarasını değerlendirmeye alabilirler. Bu extended access list'lere, trafiğin kontrolünde, çok daha detaylı kararlar verme kabiliyeti sağlar.

**Named access list'ler:** Hey, bir dakika. İki access list türü olduğunu söyledim, fakat üç tane listelendi. Named access list'ler hem standart hem de extended olduklarından, teknik olarak sadece iki tane vardır ve aslında o başka bir tür değildir. Standart ve extended access list'lerden farklı oluşturulup, belirtildiklerinden, onları ayırıyorum. Fakat işlevsel olarak aynıdırlar.

Bir access list oluşturulduğunda, biz onu uygulayana kadar hiçbir şey yapmayacaktır. Evet, onlar router'dadır fakat router'a onları ne yapacaklarını söyleyene kadar aktif değillerdir. Paket filtresi olarak bir access list kullanmak için trafiği filtrelemek istediğiniz router interface'ine uygulamanız gerekir. Ve access list'i uygulamak istediğiniz yönü belirtmek zorundasınız. Bunun için iyi bir sebebiniz var. Firmanızın internete çıkan trafiğini, internetten şirketinize gelen trafik için istediğinizden farklı şekilde kontrol etmek isteyebilirsiniz. Böylece trafiğin yönünü belirterek, tek bir interface için gidiş ve dönüş yönünde farklı access list'ler uygulayabilirsiniz (ve genelde bunu yapmanız gerekecektir):

NOT *Bu access list türlerine, bu modülde daha sonra detaylı olarak bakacağız.*

**Inbound access list'ler:** Bir access list, bir interface'teki gelen yönündeki paketlere uygulandığında, bu paketler, giden interface'e route edilmeden önce, access list ile işlemden geçirilir. Reddedilen bir paket, routing işlemine başvurulmadan önce atılacağından, route edilmeyecektir.

**Outband access list'ler:** Bir access list, bir interface'teki giden paketlere uygulandığında, bu paketler çıkış yönündeki interface'e route edilir ve sonra kuyruğa alınmadan önce, access list ile işlemden geçirilir.

Bir routerda access list'ler oluşturup uyguladığınızda, izlemeniz gereken genel bazı access-list yönergeleri vardır:

- Bir interface için bir yöne, sadece bir protokol için access list atayabilirsiniz. Yani, IP access list'ler oluşturulduğunda, interface başına sadece bir inbound access list ve bir outbound access list'e sahip olabilirsiniz.

NOT *Herhangi bir access list'in sonunda gizli deny içerdiğini düşündüğünüzde, aynı protokol için aynı yönde aynı interface'de çoklu access list'lere sahip olamamanız mantıklıdır. İlk access list'deki bazı koşullarla eşlemeyen bir paket, reddedileceğinden, ikinci bir access list ile karşılaştırmak için bir paket olmayacaktır.*

- Access list'lerinizde, daha özel koşulları, access list'in yukarısında olacak şekilde düzenleyin.

- Access list'e sonradan yeni bir kayıt eklendiğinde, o listenin altına yerleştirilecektir. Access list için bir text editor kullanılması önemle tavsiye edilir.
- Access list'ten bir satır silemezsiniz. Şayet bunu denerseniz, tüm listeyi silersiniz. List'i değiştirmeye çalışmadan önce, access list'i bir text editor'üne kopyalamak en iyisidir. Tek istisnai durum named access list'lerin kullanılmasıdır.

*Bir named access list'den bir satır silebilirsiniz. Bunu birazdan göstereceğim.*

NOT

- Access list'lerinizi, `permit any` komutu ile sonlandırmanız haricinde, listedeki herhangi bir koşula uymazlarsa, tüm paketler atılacaktır. Her list en az bir permit komutuna sahip olmalıdır, yoksa tüm trafiği reddedecektir.
- Access list'leri oluştur ve sonra onları bir interface'e uygula. Mevcut access list olmaksızın bir interface'e uygulanan bir access list, trafiği filtrelemeyecektir.
- Access list'ler, router'dan geçen trafiği filtrelemek için kullanılabilir. Onlar, router'dan gönderilen trafiği filtrelemeyeceklerdir.
- Standart IP access list'lerini mümkün olduğu kadar hedefe yakın yerleştirin. Aslında ağlarımızda standart access list'lerin çok kullanılmamasının sebebi budur. Bir standart access list'i kaynak hosta veya ağa yakın bir noktaya koyamazsınız, çünkü sadece kaynak adres bazında filtreleme yapabiliyorsunuz, yoksa hiçbir şey iletilmeyecektir.
- IP extended access list'lerini mümkün olduğu kadar kaynağa yakın bir yere yerleştirin. Extended access list'ler, adres ve protokoller bazında filtreleme yapabildiğinden, trafiğinizin tüm network boyunca dolaşmasını ve sonra reddedilmesini istemezsiniz bu list'i mümkün olduğu kadar kaynak adrese yakın bir noktaya yerleştirerek, trafiği, sizin değerli bant genişliğinizden harcamadan filtreleyebilirsiniz.

Basit ve gelişmiş access list'lerin nasıl yapılandırıldığına geçmeden, ACL'lerin bu bölümde daha önce anlattığım güvenlik tehditlerinin azaltılmasında nasıl kullanılabileceğini görelim.

## ACL'lerle Güvenlik Problemlerini Azaltmak

ACL'lerle riskini azaltabileceğiniz birçok güvenlik tehdidinin listesi aşağıdadır:

- IP adres spoofing, inbound
- IP adres spoofing, outbound
- Denial of Service (DoS) TCP SYN atakları, harici saat denetimi atakları
- TCP Intercept kullanan, DoS TCP SYN atakları
- DoS smurf atakları
- ICMP mesajlarını filtreleme, inbound
- ICMP mesajlarını filtreleme, outbound
- Traceroute filtreleme

Genelde herhangi bir dâhili host veya ağın kaynak adresini içeren IP paketlerini özel bir ağa kabul etmemek akıllıca olacaktır.

İnternetten, gerçek ağınıza yönelen güvenlik problemlerini hafifletmek için ACL'ler yapılandırdığınızda, uymanız gereken kuralların listesi şöyledir:

- İç ağınızdan herhangi bir adrese izin vermeyin.
- Herhangi bir host adresine (127.0.0.0/8) izin vermeyin.
- Herhangi bir rezerve özel adrese izin vermeyin.
- IP multicast adres aralığındaki adreslere izin vermeyin. (224.0.0.0/4)

Yukarıdaki adreslerin hiçbirinin, ağ topluluğunuza girmesine izin verilmemelidir. Son olarak, bazı basit ve gelişmiş access list'lerle çalışalım.

# Standart Access List'ler

Standart access list'ler, bir paketteki kaynak IP adresini inceleyerek, network trafiğini filtreler. 1-99 ve 1300-1999 (genişletilmiş aralık) arası access list numaralarını kullanarak, standart bir access list oluşturabilirsiniz. Access list tipleri, kullandıkları numaralarla farklılık gösterirler. Access list'in oluşturulduğunda kullanılan bu numaraya bağlı olarak router, oluşturulan access list'e hangi tür komutların girilmesi gerektiğini bilir. 1-99 veya 1300-1999 numaralarını kullanarak, router'a bir standart access list oluşturmak istediğinizi söylersiniz, bu nedenle router, sadece koşul satırında kaynak IP adresi belirten ifade olmasını bekleyecektir.

Aşağıda, ağınızdaki trafiği filtreleyebileceğiniz birçok access list numarasıyla ilgili örnek vardır:

```
Corp(config)#access-list ?
  <1-99>            IP standard access list
  <100-199>         IP extended access list
  <1100-1199>       Extended 48-bit MAC address access list
  <1300-1999>       IP standard access list (expanded range)
  <200-299>         Protocol type-code access list
  <2000-2699>       IP extended access list (expanded range)
  <700-799>         48-bit MAC address access list
  compiled          Enable IP access-list compilation
  dynamic-extended  Extend the dynamic ACL absolute timer
  rate-limit        Simple rate-limit specific access list
```

Gelin, standart bir access list oluşturulduğunda kullanılan söz dizimine bakalım:

```
Corp(config)#access-list 10 ?
  deny    Specify packets to reject
  permit  Specify packets to forward
  remark
```

Belirttiğim gibi 1-99 veya 1300-1999 numaralarını kullanarak, router'a bir standart access list oluşturmak istediğinizi söylersiniz.

Access list numarasını seçtikten sonra, `permit` veya `deny` komutu kullanmaya karar vermeniz gerekmektedir. Bu örnek için `deny` komutu kullanacaksınız:

```
Corp(config)#access-list 10 deny ?
  Hostname or A.B.C.D  Address to match
  any                  Any source host
  host                 A single host address
```

Sonraki adım, daha detaylı bir açıklama gerektirir. Üç seçenek vardır. Herhangi bir host veya ağı kabul etmek veya reddetmek için `any` parametresi kullanabilirsiniz, tek bir host veya onların belli bir aralığını belirtmek için bir IP adresi kullanabilirsiniz ya da sadece belirli bir host'u belirtmek için `host` komutunu kullanabilirsiniz. `any` komutu oldukça basittir. Herhangi bir kaynak adresi, komut ile eşleşir, böylece bu satır ile karşılaştırılan her paket eşleşecektir. `host` komutu, nispeten basittir. Onun kullanılmasıyla ilgili örnek şöyledir:

```
Corp(config)#access-list 10 deny host ?
  Hostname or A.B.C.D  Host address
Corp(config)#access-list 10 deny host 172.16.30.2
```

Bu komutlar, bu access list'in host 172.16.30.2'den gelen paketleri kabul etmeyeceğini belirtir. Varsayılan parametre, `host`'dur. Diğer bir deyişle, `access list 10 deny 172.16.30.2` yazarsanız, router host 172.16.30.2 demek istediğinizi düşünecektir.

Belirli bir host'u veya host aralığını belirtmenin başka bir yolu daha vardır. Wildcard mask işlemini kullanabilirsiniz. Zaten, host aralığı belirtmek için access list'te wildcard mask işlemi kullanmak zorundasınız.

Wildcard mask işlemi nedir? Şimdi, hem bir standart access list örneği kullanarak onun hakkında bilgi alacaksınız hem de bir sanal terminale erişimin nasıl kontrol edileceğini öğreneceksiniz.

## Wildcard Mask İşlemi

Wildcard'lar tek bir host'u, ağı veya bir ağın ya da ağların aralığını belirtmek için access list'lerle kullanılmaktadır. Bir wildcard'ı anlamak için, blok boyutunun ne olduğunu anlamanız gerekir. O, bir adres aralığını belirtmek için kullanılır. Farklı blok boyutlarının bazıları, 64, 32, 16, 8 ve 4'tür.

Belirli bir adres aralığı belirtmeniz gerektiğinde, ihtiyacınız için olandan sonraki en geniş blok boyutunu seçin. Örnek olarak, 34 network belirtmeniz gerekirse, blok boyutunuzun 64 olması gerekir. 18 host belirtmek isterseniz, 32 blok boyutuna ihtiyacınız vardır. Sadece 2 ağ belirteceksiniz, o zaman 4 blok boyutu çalışacaktır.

Wildcard'lar, router'a belirli bir adres aralığını filtrelemesini söylemek için, host veya network adresleriyle kullanılmaktadırlar. Bir host'u belirtmek için adres şöyle olacaktır:

```
172.16.30.5 0.0.0.0
```

Dört oktet, adresin her bir oktetini gösterir. Ne zaman bir sıfır görünürse, adresteki oktetin tamamen eşlemesi gerektiği anlamına gelir. Oktetin herhangi bir değer olabileceğini belirtmek için 255 değeri kullanılır. Örnek olarak, /24 subnet'inin, bir wildcard ile nasıl belirtildiği gösterilmektedir:

```
172.16.30.0 0.0.0.255
```

Bu router'a, ilk üç okteti tamamen eşleştirmesini, dördüncü oktetin herhangi bir değerde olabileceğini söyler.

Bu bölüm kolaydı. Ya sadece küçük bir adres aralığı belirtmek istersiniz? Burası blok boyutlarının önemli olacağı noktadır. Blok boyutlarında bir değer aralığı belirtmelisiniz. Örnek olarak aralık, 16 veya 32 olmalıdır, 20 olamaz.

172.16.8.0 ile 172.16.15.0 aralığındaki network bölümüne erişimi engellemek istediğinizi düşünelim. Network adresiniz 172.16.8.0 ve wildcard'ınız 0.0.7.255 olmalıdır. Bu nedir? 7.255, router'ın blok boyutunu belirlemek için kullanacağı kısımdır. Network ve wildcard, router'a 172.16.8.0 ile başlamasını ve 172.16.15.0 ağına sekiz blok boyutu gitmesini söyler. Gerçekten göründüğünden daha kolaydır. Sizin için binary hesaplaması yapabilirdim, fakat kimsenin buna ihtiyacı yok. Aslında tüm yapmanız gereken, wildcard'ın daima blok boyutundan bir küçük bir sayı olduğunu hatırlamanızdır. Bizim örneğimizde, blok boyutu 8 olduğundan, wildcard 7 olacaktır. Şayet 16 blok boyutu kullanıyorsanız, wildcard 15 olmalıdır. Kolay değil mi?

Onu iyice kavramanıza yardımcı olması için bazı örnekleri inceleyeceğim. Aşağıdaki örnek, router'a ilk üç okteti tamamen eşleştirmesini, dördüncü oktetin herhangi bir değer olabileceğini söyler:

```
Corp(config)#access-list 10 deny 172.16.10.0 0.0.0.255
```

Şimdiki örnek, router'a ilk iki okteti tamamen eşleştirmesini, son iki oktetin herhangi bir değer olabileceğini söyler:

```
Corp(config)#access-list 10 deny 172.16.0.0 0.0.255.255
```

Alttaki satırı anlamaya çalışın:

```
Corp(config)#access-list 10 deny 172.16.16.0 0.0.3.255
```

Bu konfigürasyon, router'a 172.16.16.0 ağıyla başlamasını ve 4 blok boyutunu kullanmasını söyler. Aralık 172.16.16.0'dan 172.16.19.0'a kadar olacaktır.

Aşağıdaki örnek, 172.16.16.0 ile başlayan ve 8 blok boyutuyla 172.16.23.0'a devam eden bir access list'i göstermektedir:

```
Corp(config)#access-list 10 deny 172.16.16.0 0.0.7.255
```

Şimdiki örnek, 172.16.32.0 ile başlar ve 16 blok boyutuyla 172.16.47.0'a kadar devam eder:

```
Corp(config)#access-list 10 deny 172.16.32.0 0.0.15.255
```

Sıradaki örnek, 172.16.64.0 ile başlar ve 64 blok boyutuyla 172.16.127.0'a kadar devam eder:

```
Corp(config)#access-list 10 deny 172.16.64.0 0.0.63.255
```

Son örnek, 172.16.160.0 ile başlar ve 32 blok boyutuyla 172.16.191.255'e kadar devam eder:

```
Corp(config)#access-list 10 deny 192.168.160.0 0.0.31.255
```

Blok boyutu ve wildcard ile çalıştığınızda aklınızda tutmanız gereken iki şey daha vardır:

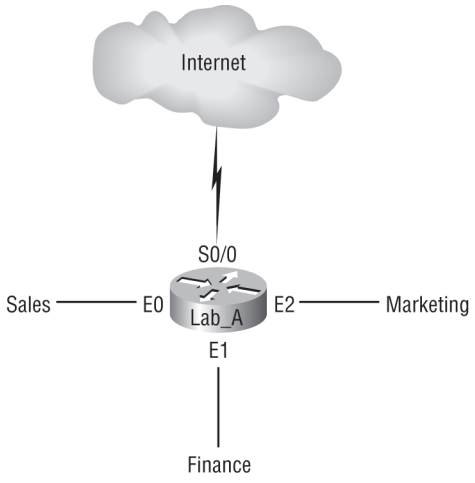
- Her blok boyutu 0 veya çoklu bir blok boyutuyla başlamak zorundadır. Örneğin, 8 blok boyutu istiyorum ve 12'den başlayacağım diyemezsiniz. 0-7, 8-15, 16-23 vs kullanmalısınız. 32 blok boyutu için, aralıklar, 0-31, 32-63, 64-95 vs olmalıdır.
- `any` komutu, wildcard 0.0.0.0 255.255.255.255 yazmakla aynı şeydir.

NOT *Wildcard mask işlemi, IP access list'ler kullanılacaksa, iyice öğrenilmesi gereken çok önemli bir konudur. Standart ve extended access list'lerde aynı şekilde kullanılmaktadır.*

## Standart Access List Örneği

Bu bölümde, belirli kullanıcıların Finans bölümü LAN'ına erişimini engellemek için bir standart access list'in nasıl kullanılacağını öğreteceğim.

Şekil 10.2'de bir router, üç LAN bağlantısı ve internet için bir WAN bağlantısına sahiptir. Satış LAN'ındaki kullanıcılar, Finans LAN'ına erişim hakkına sahip olmamalıdır. Pazarlama LAN'ının, uygulama servisleri için Finans LAN'ına erişimi gerekmektedir.




***Şekil 10.2:*** *Üç LAN ve bir WAN bağlantısı ile IP access list örneği.*

Şekildeki router'da, aşağıdaki IP access list'ler yapılandırılmıştır:

```
Lab_A#config t
Lab_A(config)#access-list 10 deny 172.16.40.0 0.0.0.255
Lab_A(config)#access-list 10 permit any
```

Aşağıda kullanılan wildcard mask işlemindeki ifade ile any ifadesinin aynı anlamda kullanıldığını bilmek önemlidir:

```
Lab_A(config)#access-list 10 permit 0.0.0.0 255.255.255.255
```

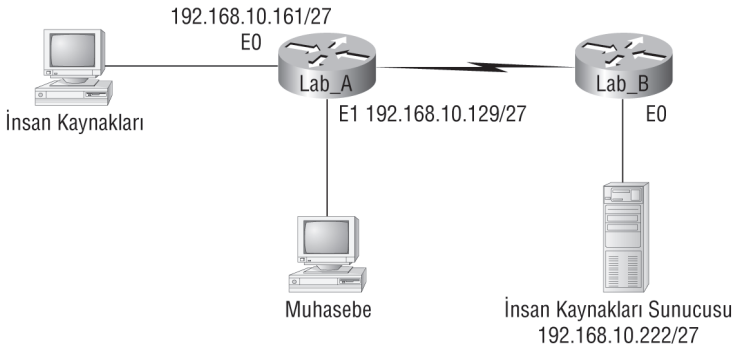
Wildcard mask'ın oktetlerden hiçbirinin değerlendirilmeyeceğini belirtmesinden dolayı, her adres test koşulunu eşleştirir. Bu işlevsel olarak any kullanmakla aynı şeydir.

Bu noktada access list, kaynak adreslerin Satış LAN'ından Finans LAN'ına erişimi reddetmek ve diğer her şeyi kabul etmesi için yapılandırılmıştır. Fakat hatırlayın, access list belirli bir yönde bir interface'e uygulanana kadar bir eylem olmayacaktır. Peki, bu access list nereye yerleştirilmelidir? Onu, E0'da geliş yönünde bir access list olarak yerleştirirseniz, tüm Satış LAN cihazlarının, router'a bağlı tüm ağlara erişimi engelleneceğinden, Ethernet interface'ini kapatmanız daha iyi olabilir. Bu access list'ini uygulayacağınız en iyi yer, bir outbound list olarak, E1 interface'idir:

```
Lab_A(config)#int e1
Lab_A(config-if)#ip access-group 10 out
```

Bu, 172.16.40.0'dan tüm trafiğin Ethernet 1'den gönderilmesini engeller. Bu hedeflere trafiğin, E1 interface'ine iletilmemesinden dolayı, Satış LAN'ındaki kullanıcıların Pazarlama LAN'ına ve İnternet erişimine bir etkisi yoktur. E1'den çıkmaya çalışan bir paket, ilk olarak access list'e uğramak zorundadır. Şayet E0'da yerleştirilmiş geliş yönünde bir access list varsa, interface E0'a girmeye çalışan bir paket, bir çıkış interface'ine route edilmeden önce access list'ten geçmek zorundadır.

Gelin standart access list'lerle ilgili başka bir örneğe bakalım. Şekil 10.3, üç LAN ve bir seri WAN bağlantısına sahip iki router'lı bir ağ topluluğunu göstermektedir.




***Şekil 10.3:*** *IP standart access list örneği 2.*

Muhasebe kullanıcılarının, Lab_B router'ına bağlı İnsan Kaynakları sunucusuna erişimini durdurmak, diğer tüm kullanıcıların bu LAN'a erişimine izin vermek istiyorsunuz. Hangi standart access list'i oluşturmalısınız ve onu nereye yerleştirmelisiniz?

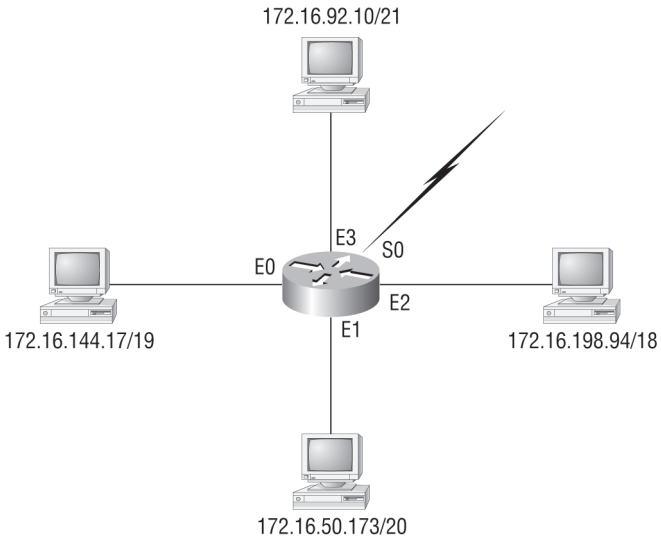
Asıl cevap, bir extended access list oluşturmanız ve onu kaynağa en yakın yere yerleştirmenizdir. Fakat soru, bir standart access list kullanılması gerektiğini belirtmektedir. Genel kural olarak, standart access list'ler hedefe en yakın yere yerleştirilir. Bu örnekte, Lab_B router'ının Ethernet0 gidiş yönüdür. Aşağıda, Lab_B router'ına yerleştirmeniz gereken access list görülmektedir.

```
Lab_B#config t
Lab_B(config)#access-list 10 deny 192.168.10.128 0.0.0.31
Lab_B(config)#access-list 10 permit any
```

```
Lab_B(config)#interface Ethernet 0
Lab_B(config-if)#ip access-group 10 out
```

Bir router'da Telnet erişimini kısıtlamaya geçmeden önce, bir standart access örneğine daha bakalım. Fakat bu acces list bazı şartlar gerektirecektir. Şekil 10.4'de, dört LAN bağlantısı ve İnternete bir WAN bağlantısı olan bir router vardır.

Diyagramda görülen dört LAN'ın her birinin İnternet'e erişimini durduracak bir access list yazmanız gerekmektedir. LAN'ların her biri, tek host'un IP adresini gösterir ve buradan access list'i yapılandırmak için subnet ve wildcard'ları belirlemelisiniz.




***Şekil 10.4:*** *IP standart access list örneği 3.*

Cevap şöyledir (E0'daki network ile başlar ve E3'e doğru çalışır):

```
Router(config)#access-list 1 deny 172.16.128.0 0.0.31.255
Router(config)#access-list 1 deny 172.16.48.0 0.0.15.255
Router(config)#access-list 1 deny 172.16.192.0 0.0.63.255
Router(config)#access-list 1 deny 172.16.88.0 0.0.7.255
Router(config)#access-list 1 permit any
Router(config)#interface serial 0
Router(config-if)#ip access-group 1 out
```

Peki, bu list'i hazırlamanın amacı ne olabilir? Bu access list'i gerçekten bir router'a uygularsanız, etkin olarak İnternet'e erişimi durdurursunuz. Öyleyse internet bağlantısına sahip olmamızın amacı nedir? Blok boyutlarının access list'lerle nasıl bloklandığını uygulayabilmeniz için bu örneği yazdım. CCNA hedeflerine çalıştığınızda başarınız için bu önemlidir.

## VTY (Telnet) Erişimlerinin Kontrol Edilmesi

Bir router'daki aktif interface'ler, VTY erişimi için kolay hedef olduklarından, bir router'ı kullanıcıların telnet yapmasından korumaya çalışmak için muhtemelen zorluk yaşarsınız. Router'daki tüm IP adreslerine Telnet erişimini sınırlandırmak için, bir extended access list oluşturmaya çalışabilirsiniz. Şayet bunu yaparsanız, onu tüm interface'lerde geliş yönüne uygulamak zorunda olursunuzve düzinelerce, hatta yüzlerce interface'e sahip büyük bir router'da kolayca yapamazsınız, değil mi? Daha iyi bir çözüm var: VTY hatlarına erişimi kontrol etmek için standart access list kullanın.

Bu neden çalışır? Çünkü VTY hatlarına bir access list uyguladığınızda, VTY'ye erişim, terminal erişimi anlamına geldiğinden, Telnet protokolünü belirtmek zorunda değilsiniz. Telnet oturumu

için kullanıcının kullandığı interface adresinin bir önemi olmadığından, bir hedef adres belirtmenize de gerek yoktur. Gerçekte sadece kullanıcının nereden geldiğini (onların kaynak adreslerini) kontrol etmeniz gerekir.

Bu fonksiyonu çalıştırmak için, aşağıdaki adımları izleyin:

1. Router'larınıza telnet yapabilmesini isteyeceğiniz host veya host'larınıza izin veren bir IP Access list oluşturun.
2. Access list'i, `access-class` komutuyla VTY line'larına uygulayın.

Bir router'a telnet yapmasına sadece 172.16.10.3 host'unun izin verildiği örnek şöyledir:

```
Lab_A(config)#access-list 50 permit 172.16.10.3
Lab_A(config)#line vty 0 4
Lab_A(config-line)#access-class 50 in
```

### Bir Router'daki Telnet Bağlantılarınızı Güvenli Yapmalı mısınız?

*Ağınızı görüntülüyorsunuz ve birilerinin* `show users` *komutunu kullanarak core router'ınıza telnet yaptığını fark ettiniz.* `Disconnect` *komutunu kullandınız ve onları router'dan bağlarını kopardınız. Fakat 5 dakika sonra tekrar router'a bağlandıklarını fark ettiniz. Router interface'lerine bir Access list koymayı düşünüyorsunuz, fakat router'ınız zaten birçok paketle uğraştığından, her interface'de gecikmeye sebep olmak istemiyorsunuz. VTY line'larının kendilerine bir access list koymayı düşünüyorsunuz, fakat bunu yapmadan önce, her interface'e bir access list koymanın güvenli bir alternatif olup olmadığından emin değilsiniz. Bu network için, VTY line'lara bir access list koymak iyi bir fikir midir?*

*Evet, kesinlikle ve bu bölümde gösterilen* `access-class` *komutu, bunu yapmak için en iyi yoldur. Niçin? Çünkü o, gelen ve giden her pakete bakan bir interface'de çalışan bir access list kullanmaz. Bu, route edilecek paketlerde ek yüke sebep olabilir.*

*VTY line'lara* `access-class` *komutunu koyduğunuzda, sadece router'a telnet yapmaya çalışan paketlere bakılacak ve eşleştirilecektir. Bu, ağınıza güzel, uygulaması kolay bir güvenlik sağlar.*

Router'daki IP adreslerinin hangisinin bir hedef olarak kullanıldığına bakmaksızın, access list, 172.16.10.3 host'u dışındaki host'ların router'a telnet yapmalarını durdurur.

NOT

*Cisco, bir router'ın VTY line'ında Telnet yerine Secure Shell (SSH) kullanmanızı önerir. SSH'ın nasıl yapılandırıldığı hakkında daha fazla bilgi almak için bölüm 4'e bakın.*

## Extended Access List'ler

Daha önceki standart access list örneklerinde, Satış LAN'ından, finans bölümüne tüm erişimin nasıl engellenmek zorunda kalındığına dikkat edin. Güvenlik ihtiyaçlarınız sebebiyle, Satışçıların, Finans LAN'ındaki tüm network servislerine değil de sadece belirli bir sunucuya erişimini sağlamaya gerek duyarsanız? Standart bir Access list ile kullanıcılar için belirli bir network servisine izin verilip de diğerlerine izin verilmemesini sağlayamazsınız. Başka bir deyişle, hem kaynak hem de hedef adresleri bazında kararlar vermeniz gerektiğinde, sadece kaynak adres bazında kararlar verdiğinden, bir standart access list ile bunu yapabilmeniz mümkün olmayacaktır.

Fakat bir extended access list, size bunu sağlayacaktır. Bundan dolayı, extended acces list'ler, hem kaynak ve hedef adresi belirlemenize hem de üst-katman protokol ve uygulamayı tanımlayan protokol ve port numarasını belirtmenize izin verecektir. Extended Access list'leri kullanarak, etkin şekilde kullanıcıların fiziksel bir LAN'a erişimlerine izin verebilir ve onların belirli host'lara, hatta bu host'lardaki belirli servislere erişimini engelleyebilirsiniz.

Bir extended Access list örneği aşağıdaki gibidir:

```
Corp(config)#access-list ?
 <1-99>            IP standard access list
 <100-199>         IP extended access list
 <1100-1199>       Extended 48-bit MAC address access list
 <1300-1999>       IP standard access list (expanded range)
 <200-299>         Protocol type-code access list
 <2000-2699>       IP extended access list (expanded range)
 <700-799>         48-bit MAC address access list
 compiled          Enable IP access-list compilation
 dynamic-extended  Extend the dynamic ACL absolute timer
 rate-limit        Simple rate-limit specific access list
```

İlk komut, uygun access-list numaralarını gösterir. 100 ile 199 arasında access list aralığını kullanacaksınız. Extended IP access list'ler için 2000-2699 aralığının da geçerli olduğuna dikkat ettiğinize emin olun.

Bu noktada, ne tür kayıt yaptığınıza karar vermeniz gerekir. Bu örnek için bir deny list kaydı seçeceksiniz:

```
Corp(config)#access-list 110 ?
  deny     Specify packets to reject
  dynamic  Specify a DYNAMIC list of PERMITs or DENYs
  permit   Specify packets to forward
  remark   Access list entry comment
```

Access list tipini seçince, bir protokol alan kaydı seçmeniz gerekir.

```
Corp(config)#access-list 110 deny ?
  <0-255>  An IP protocol number
  ahp      Authentication Header Protocol
  eigrp    Cisco's EIGRP routing protocol
  esp      Encapsulation Security Payload
  gre      Cisco's GRE tunneling
  icmp     Internet Control Message Protocol
  igmp     Internet Gateway Message Protocol
  ip       Any Internet Protocol
  ipinip   IP in IP tunneling
  nos      KA9Q NOS compatible IP over IP tunneling
  ospf     OSPF routing protocol
  pcp      Payload Compression Protocol
  pim      Protocol Independent Multicast
  tcp      Transmission Control Protocol
  udp      User Datagram Protocol
```

NOT

*Şayet Application katman protokolü ile filtreleme isterseniz, permit veya deny komutundan sonra, uygun 4.katman transport protokolünü seçmek zorundasınız. Örneğin, Telnet veya FTP'yi filtrelemek için Transport katmanında hem Telnet hem de FTP'nin, TCP kullanmasından dolayı TCP'yi seçin. Şayet IP seçseydiniz, sonradan belirli bir uygulama protokolü seçmenize izin verilmeyecekti.*

Burada TCP'yi protokol olarak seçerek, TCP'yi kullanan bir Application katman protokolü filtrelemeyi seçeceksiniz. Sonra, belirli bir TCP portu belirleyeceksiniz. Daha sonra, host veya ağın kaynak IP adresi belirtmeniz istenecek. (Herhangi bir kaynak adresini kabul etmek için any komutunu kullanabilirsiniz):

```
Corp(config)#access-list 110 deny tcp ?
  A.B.C.D  Source address
  any      Any source host
  host     A single source host
```

Kaynak adresi seçildikten sonra hedef adresi seçilir:

```
Corp(config)#access-list 110 deny tcp any ?
  A.B.C.D  Destination address
  any      Any destination host
  eq       Match only packets on a given port number
  gt       Match only packets with a greater port number
  host     A single destination host
  lt       Match only packets with a lower port number
  neq      Match only packets not on a given port number
  range    Match only packets in the range of port numbers
```

Aşağıdaki örnekte, 172.16.30.2 hedef IP adresine sahip bir kaynak adresi kabul edilmeyecektir.

```
Corp(config)#access-list 110 deny tcp any host 172.16.30.2 ?
  ack          Match on the ACK bit
  dscp         Match packets with given dscp value
  eq           Match only packets on a given port number
  established  Match established connections
  fin          Match on the FIN bit
  fragments    Check non-initial fragments
  gt           Match only packets with a greater port number
  log          Log matches against this entry
  log-input    Log matches against this entry, including input
interface
  lt           Match only packets with a lower port number
  neq          Match only packets not on a given port number
  precedence   Match packets with given precedence value
  psh          Match on the PSH bit
  range        Match only packets in the range of port numbers
  rst          Match on the RST bit
  syn          Match on the SYN bit
  time-range   Specify a time-range
  tos          Match packets with given TOS value
  urg          Match on the URG bit
  <cr>
```

Burada Enter tuşuna basabilir ve Access list olarak bırakabilirsiniz. Fakat bunu yaparsanız, 172.16.30.2 host'una tüm IP trafiği, hedef portuna bakılmaksızın reddedilecektir. Daha belirleyici

bile olabilirsiniz: Uygun yerleştirilmiş host adreslerine sahip olunca, reddedeceğiniz servis tipini belirtin. Aşağıdaki yardım ekranı, size uygun seçenekleri göstermektedir. Bir port numarası seçebilir veya uygulama ya da protokol adı kullanabilirsiniz

```
Corp(config)#access-list 110 deny tcp any host 172.16.30.2 eq ?
  <0-65535>    Port number
  bgp          Border Gateway Protocol (179)
  chargen      Character generator (19)
  cmd          Remote commands (rcmd, 514)
  daytime      Daytime (13)
  discard      Discard (9)
  domain       Domain Name Service (53)
  drip         Dynamic Routing Information Protocol (3949)
  echo         Echo (7)
  exec         Exec (rsh, 512)
  finger       Finger (79)
  ftp          File Transfer Protocol (21)
  ftp-data     FTP data connections (20)
  gopher       Gopher (70)
  hostname     NIC hostname server (101)
  ident        Ident Protocol (113)
  irc          Internet Relay Chat (194)
  klogin       Kerberos login (543)
  kshell       Kerberos shell (544)
  login        Login (rlogin, 513)
  lpd          Printer service (515)
  nntp         Network News Transport Protocol (119)
  pim-auto-rp  PIM Auto-RP (496)
  pop2         Post Office Protocol v2 (109)
  pop3         Post Office Protocol v3 (110)
  smtp         Simple Mail Transport Protocol (25)
  sunrpc       Sun Remote Procedure Call (111)
  syslog       Syslog (514)
  tacacs       TAC Access Control System (49)
  talk         Talk (517)
  telnet       Telnet (23)
  time         Time (37)
  uucp         Unix-to-Unix Copy Program (540)
  whois        Nicname (43)
  www          World Wide Web (HTTP, 80)
```

Bu noktada, sadece 172.16.30.2 host'una Telnet'i (port 23) engelleyelim. Şayet kullanıcılar FTP yapmak isterlerse, kabul edilecektir. Access list'in logunu tutmak için `log` komutu kullanılır. Bu, uygunsuz erişimleri görüntülemenin çok kullanışlı bir yöntemi olabilir. Bunu şöyle yaparsınız:

```
Corp(config)#access-list 110 deny tcp any host 172.16.30.2 eq 23
log
```

Son satırın varsayılan olarak gizli bir deny komutu olduğunu aklınızdan çıkarmayın. Şayet bu Access lis'ti bir interface'e uygularsanız, her access list'in sonunda gizli deny olduğundan, interface'i kapatırsanız iyi olur. Access list'e aşağıdaki komutu eklemeniz gerekmektedir:

```
Corp(config)#access-list 110 permit ip any any
```

0.0.0.0 255.255.255.255'in, any ile aynı komut olduğunu hatırlayın. Böylece komut şöyle görünecektir:

```
Corp(config)#access-list 110 permit ip 0.0.0.0 255.255.255.255
0.0.0.0 255.255.255.255
```

Access list oluşturulduğunda onu bir interface'e eklemeniz gerekir (IP sandart list ile aynı komut):

```
Corp(config-if)#ip access-group 110 in
```

Veya şunu:

```
Corp(config-if)#ip access-group 110 out
```

Şimdiki bölümde, extended Access list'in nasıl kullanılacağıyla ilgili bir örneği inceleyeceğiz.

## Extended Access List Örneği 1

Daha önce IP standart Access list örneğinde kullandığımız Şekil 10.2'yi kullanarak, aynı ağı seçelim ve Finans bölümü LAN'ındaki 172.16.30.5 host'unun, Telnet ve FTP servisleriyle erişimini engelleyelim. Bundaki diğer tüm servislerin ve diğer tüm host'ların, satış ve pazarlama bölümüne erişimi kabul edilmektedir.

Aşağıdaki Access list oluşturulmalıdır:

```
Lab_A#config t
Lab_A(config)#access-list 110 deny tcp any host
  172.16.30.5 eq 21
Lab_A(config)#access-list 110 deny tcp any host
  172.16.30.5 eq 23
Lab_A(config)#access-list 110 permit ip any any
```

Access-list 110, router'a bir extended IP access list oluşturduğunuzu söyler. TCP, Network katman başlığındaki protokol alanıdır. Şayet access list, burada tcp kullanmazsa, örnekte gösterildiği gibi, 21 ve 23 port numaralarıyla filtreleme yapamazsınız. (FTP ve Telnet vardır ve her ikisi de, connection-oriented servisler oldukları için TCP kullanırlar). Herhangi bir IP adresi anlamına gelen any komutu kaynaktır. Host ise hedef IP adresidir.

NOT

*Extended Access list oluşturduğumuzda, host 172.16.30.5 yerine 172.16.30.5 0.0.0.0 girebileceğimizi ve router'ın, running-config'inde komutu, host 172.16.30.5 olarak değiştirmesi dışında, sonucun değişmeyeceğini hatırlayın.*

List oluşturulduktan sonra, Ethernet 1 interface'ine gidiş yönünde uygulanması gerekir. Bu oluşturduğumuz politikayı tüm host'lara uygular ve yerel LAN'ın dışından gelen tüm FTP ve Telnet'in 172.16.30.5'e erişimini etkin şekilde engeller. Bu access list, sadece Satış LAN'ından erişimi engellemek için oluşturulsaydı, bu list'i kaynağa daha yakın bir yere veya Ethernet interface 0'a koyacaktık. Bu durumda, access list'i geliş trafiğine uygulayacaktık.

Devam edelim ve access list'i, E1 interface'ine uygulayalım ve host'a dışarıdan gelen tüm FTP ve Telnet erişimini engelleyelim:

```
Lab_A(config-if)#ip access-group 110 out
```

## Extended Access List Örneği 2

Bu örnekte tekrar dört LAN ve bir seri bağlantıya sahip Şekil 10.4'ü kullanacağız. Yapmamız gereken, Ethernet 1 ve Ethernet 2 interface'lerine bağlı ağlara Telnet erişimini durdurmaktır. Şayet sadece bir Access list kullansaydık, Ethernet 1 ve 2 interface'lerinde gecikmeye sebep olacağından (bu interface'den giden tüm paketlere bakılacağından) çok verimli olmazdı. Fakat iki access list oluştursaydık, doğru yapılandırdığımız her interface'de daha az gecikme olurdu. Bununla beraber CCNA konularını çalıştığımızdan, bunu sadece tek Access list ile yapacağız.

Cevap değişebileceği halde, router'daki yapılandırma, şöyle görünecektir:

```
Router(config)#access-list 110 deny tcp any 172.16.48.0 0.0.15.255
eq 23
Router(config)#access-list 110 deny tcp any 172.16.192.0
0.0.63.255
eq 23
Router(config)#access-list 110 permit ip any any
Router(config)#interface Ethernet 1
Router(config-if)#ip access-group 110 out
Router(config-if)#interface Ethernet 2
Router(config-if)#ip access-group 110 out
```

Bu access list'te anlamanız gereken önemli bilgiler şunlardır: İlk olarak, oluşturmak istediğiniz Access list türü için numara aralığınızın doğruluğunu kontrol etmeniz gerekmektedir. Bu örnekte, extended kullanılacaktır, bu nedenle aralık, 100-199 olmalıdır. İkincisi, protokol alanının, üst-katman proses veya uygulamayla eşleştiğinin doğruluğunu kontrol etmeniz gerekmektedir. Bu örnekte, port 23'dür (Telnet).

Telnet'in TCP kullanmasından dolayı, Protokol parametreleri TCP olmalıdır. Şayet soru TFTP kullanılmasını belirtseydi, o zaman protokol parametresi TFTP, UDP kullandığından dolayı UDP olacaktı. Üçüncüsü hedef port numarasının filtreleyeceğiniz uygulamayla eşleştiğini doğrulayın. Bu durumda, port 23, doğru şekilde, Telnet ile eşleşir. Son olarak, `permit ip any any` koşul komutunun Ethernet 1 ve Ethernet 2'ye bağlı LAN'lar için hedeflenen Telnet paketleri dışındaki diğer tüm paketlere izin verileceğinin belirtilmesi için list'in sonunda olması önemlidir.

# Gelişmiş Access List'ler

Bu bölümde, Access list'leri kullanmanın daha gelişmiş bazı yollarını göstereceğim. Gelişmiş Access-list başlıklarının çoğu bu kitabın hedefleri dışındadır. Bu nedenle, onları özet olarak işleyeceğim. Şayet ilgilenirseniz, Cisco web sitesinden daha fazla bilgi bulabilirisiniz.

Bununla beraber, bilmeniz gereken bazı önemli Access-list seçenekleri vardır. Bunların ilki Named Access list'lerdir.

## Named ACL'ler

Daha önce söylediğim gibi named access list'ler, standart ve extended access list'leri oluşturmanın farklı bir yoludur. Orta ile büyük ölçekli şirketlerde access list'lerin yönetimi gerçekten oldukça fazla zaman alabilir. Örnek olarak, bir access list'te değişiklik yapmanız gerektiğinde sıkça yapılan uygulama, access list'in bir text-editor'üne kopyalanması, numaranın değiştirilmesi, list'in düzenlenmesi ve sonra da yeni list'in router'a yapıştırılmasıdır.

Bunu yaparak sadece interface'deki access-list numarasını, eski olan ile yenisini değiştirebilirsiniz. Ağda, access list'in olmadığı bir durum olmayacaktır. Gereksiz şeyleri toplayıp hiçbir şeyi çöpe atmayan adam mantalitesi olmasaydı, bu oldukça iyi çalışacaktı. Eski access list'leri ne yapacağım? Onları silelim mi? sorusu akla gelir. Veya yeni access list ile problem yaşamam

durumunda, onu geri değiştirmek için kaydetmeli miyim? Bu ve sayısız senaryolarla ne olacak? Kendinizi, router'da çok sayıda uygulanmamış access list oluşturmuş olarak bulabilirsiniz. Onlar ne içindi? Önemli miydiler? Onlara ihtiyacım var mı? Hepsi güzel sorudur ve named access list'ler bunlara cevap verebilir.

Bunlar, çalışan access list'lere de uygulanabilir. Çalışan bir ağa geldiğinizi ve router'daki access list'lere baktığınızı düşünelim. 33 satır uzunluğundaki 177 nolu access list'i (extended Access list) bulduğunuzu farz edelim. Bu daha fazla gereksiz soru sormanıza neden olacaktır. Bu ne içindir? O niye buradadır? Bunun yerine, finans LAN isimli bir access list, 177'den daha belirleyici olmaz mıydı?

Named access list'ler, isimler kullanarak hem standart hem de extended Access list'ler uygulamanıza izin verir. İnsanlara anlaşılabilir bir yöntem olarak belirtilebilmeleri dışında bu access list'lerle ilgi yeni ve farklı bir şey yoktur. Fakat söz diziminde bazı ince değişiklikler vardır. Bu nedenle, named access list'ler kullanarak, Şekil 10.2'deki test ağımız için daha önce yazdığımız standart access list'i yeniden oluşturalım:

```
Lab_A#config t
Enter configuration commands, one per line.  End with CNTL/Z.
Lab_A(config)#ip access-list ?
  extended  Extended Acc
  logging   Control access list logging
  standard  Standard Access List
```

`ip access list` (access list değil) yazarak başladığıma dikkat edin. Bu benim named access list'e girmeme izin verir. Sonra onun bir standart access list olduğunu belirtmem gerekecektir:

```
Lab_A(config)#ip access-list standard ?
  <1-99>  Standard IP access-list number
  WORD    Access-list name

Lab_A(config)#ip access-list standard BlockSales
Lab_A(config-std-nacl)#
```

Bir standart access list belirttim, sonra bir isim ekledim: BlockSales. Bir standart access list için bir numara kullanabileceğime dikkat edin. Tanımlayıcı bir isim kullanmayı seçtim. İsim girdikten sonra, Enter tuşuna bastığımı ve komut istemcisinin değiştiğine de dikkat edin. Şimdi named access list konfigürasyon modundayım ve named access list giriyorum

```
Lab_A(config-std-nacl)#?
Standard Access List configuration commands:
  default  Set a command to its defaults
  deny     Specify packets to reject
  exit     Exit from access-list configuration mode
  no       Negate a command or set its defaults
  permit   Specify packets to forward

Lab_A(config-std-nacl)#deny 172.16.40.0 0.0.0.255
Lab_A(config-std-nacl)#permit any
Lab_A(config-std-nacl)#exit
Lab_A(config)#^Z
Lab_A#
```

Access list'i girdim ve sonra konfigürasyon modundan çıktım. Daha sonra Access list'in router'da olduğunu kontrol etmek için çalışan konfigürasyona göz atıyorum:

```
Lab_A#show running-config

!
ip access-list standard BlockSales
 deny   172.16.40.0 0.0.0.255
 permit any
!
```

BlockSales access list'i gerçekten oluşturulmuştur ve router'ın running-config'indedir. Sonra access list'i bir interface'e uygulamam gerekecektir:

```
Lab_A#config t
Enter configuration commands, one per line.  End with CNTL/Z.
Lab_A(config)#int e1
Lab_A(config-if)#ip access-group BlockSales out
Lab_A(config-if)#^Z
Lab_A#
```

Tamam, bu noktada named access list kullanarak, daha önce yapılan çalışmayı tekrar oluşturma işini tamamladık.

## Switch Port ACL'leri

Şimdi şunu hatırlayın: port ACL'leri sadece switch'inizdeki katman2 interface'lere uygulayabilirsiniz. Neden? Çünkü onlar sadece fiziksel katman2 interface'lerini desteklerler. Aklınızda tutmanız gereken diğer önemli şey, hem onları interface'inizin sadece geliş yönüne uygulayabileceğiniz hem de sadece named access list'leri kullanabileceğinizdir.

Desteklenen access list'lerin kısa bir listesi şöyledir:

Belirttiğim gibi standart access list'ler, trafiği filtrelemek için sadece kaynak adreslerini kullanır.

- Diğer taraftan extended access list'ler hem kaynak ve hedef adreslerini hem de seçmeli protokol bilgisi ve port numaralarını kullanır.
- Ayrıca, kaynak ve hedef MAC adresi ile seçmeli protokol tipini kullanan MAC extended access list'ler vardır.

Switch'ler, belirli bir interface'e uygulanan geliş yönündeki tüm ACL'leri dikkatle inceler ve trafiğin ACL ile iyi eşleşip, eşleşmediğine bağlı olarak trafiği kabul edip etmeyeceğine karar verir. Bu nedenle, ACL'lerin güvenlik için ne kadar önemli olduğunu anladığınıza eminim. Onlar, ağınızın belirli bir segmentine veya tamamına erişimi kabul edip etmeme gücüne sahip kapıcılardır.

ACL'ler, ayrıca LAN'lardaki trafiği kontrol etmek için de kullanılabilir. Bunu gerçekleştirmek için ACL'yi bir trunk porta uygulamanız gerekir. Eğer bunu, üzerinde voice VLAN olan bir portta yaptıysanız, bundan kaçının. Bu ACL'ler aslında veri VLAN'larınızı da filtreleyecektir. Bu sebeple burada dikkatli olun.

Port ACL'ler, IP Access list'ler üzerinden IP trafiğini kontrol eder. IP olmayan bir trafik, MAC adresi kullanılarak filtrelenir. Her iki filtre tipini tek bir interface'e uygulayabileceğiniz halde, onlardan sadece birini uygulamalısınız. Zaten ACL bulunan bir interface'e ilave ACL yerleştirmeye çalışırsanız, yeni LAN mevcut olanı geçersiz kılacaktır. Bu nedenle burada dikkatli olmak önemlidir. Başlamadan önce iyice düşünün.

Gelin access list'i kontrol edelim:

```
S1#config t
S1(config)#mac access-list ?
  extended  Extended Access List
S1(config)#mac access-list extended ?
  WORD  access-list name
S1(config)#mac access-list extended Todd_MAC_List
S1(config-ext-macl)#deny ?
  H.H.H  48-bit source MAC address
  any    any source MAC address
  host   A single source host
S1(config-ext-macl)#deny any ?
  H.H.H  48-bit destination MAC address
  any    any destination MAC address
  host   A single destination host
S1(config-ext-macl)#deny any host ?
  H.H.H  48-bit destination MAC address
S1(config-ext-macl)#deny any host 000d.29bd.4b85
S1(config-ext-macl)#permit ?
  H.H.H  48-bit source MAC address
  any    any source MAC address
  host   A single source host
S1(config-ext-macl)#permit any any
S1(config-ext-macl)#do show access-list
Extended MAC access list Todd_MAC_List
    deny   any host 000d.29bd.4b85
    permit any any
S1(config-ext-macl)#
```

Sadece extended Access list oluşturabileceğinizi görebilirsiniz. Başka seçeneğiniz yoktur. Ve sonunda `permit any any` eklemeyi unutmayın.

List'i bir switch port'una nasıl ekleyeceğinizi aşağıda görebilirsiniz:

```
S1(config-ext-macl)#int f0/6
S1(config-if)#mac access-group Todd_MAC_List in
```

Bu, `mac` komutuyla başlamanız dışında, bir IP list ile olanla neredeyse aynıdır.

Fakat gerçekten MAC adreslerini reddetmek ister misiniz? Başlamadan kulağa kötü geliyor. Başka seçeneğiniz olmadığı ve genellikle daha iyi olduğunu düşündüğünüz özel durumlarda doğru iken, bunun yerine, Ethernet frame başlığındaki ether-type alanı bazında erişimi reddedin. Şuna göz atın:

```
S1(config-ext-macl)#deny any any ?
  <0-65535>     An arbitrary EtherType in decimal, hex, or octal
  aarp          EtherType: AppleTalk ARP
  amber         EtherType: DEC-Amber
```

```
appletalk      EtherType: AppleTalk/EtherTalk
cos            CoS value
dec-spanning   EtherType: DEC-Spanning-Tree
decnet-iv      EtherType: DECnet Phase IV
diagnostic     EtherType: DEC-Diagnostic
dsm            EtherType: DEC-DSM
etype-6000     EtherType: 0x6000
etype-8042     EtherType: 0x8042
lat            EtherType: DEC-LAT
lavc-sca       EtherType: DEC-LAVC-SCA
lsap           LSAP value
mop-console    EtherType: DEC-MOP Remote Console
mop-dump       EtherType: DEC-MOP Dump
msdos          EtherType: DEC-MSDOS
mumps          EtherType: DEC-MUMPS
netbios        EtherType: DEC-NETBIOS
vines-echo     EtherType: VINES Echo
vines-ip       EtherType: VINES IP
xns-idp        EtherType: XNS IDP
<cr>
```

Çok hoş, biliyorum. Fakat burada aklınızı kaçırmayın ve tüm uygun ether-type numaralarını reddetmeye başlayın. Sonunda sizi üzecek bazı sorunlarla karşılaşacaksınız. Fakat gerçekte DecNet ve Apple Talk'u kim kullanır? Onlar güzel ağınıza erişimlerinin engellenmesini hak etmektedirler. Değil mi?

Şayet bölüm 1'i okuduysanız, 0x800'ü blokladığınızda, tüm IP'leri engellediğinizi bilmeniz gerekir, değil mi? Bu, gelecekte herkesi IPv6 kullanmayı zorlamak istediğinizde işe yarayabilir. Fakat şimdi değil.

## Lock and Key (Dinamik ACL'ler)

Bu ACL çeşitliliği, extended ACL'lerle kombinasyonunda, uzak veya lokal Telnet kimlik denetimine bağlıdır.

Bir dinamik ACL yapılandırabilmenizden önce, ona doğru trafiğin akışını durdurmak için router'ınıza bir extended ACL uygulamanız gerekir. Kuşatmadan kurtulabilmenin tek yolu, router'a telnet yapmak ve kimlik denetimine ulaşmaktır. Bu şöyle çalışır: Kullanıcının başlattığı telnet bağlantısı kesilir ve yerine, zaten var olan extended ACL'e eklenen, tek-kayıtlı dinamik bir ACL yerleştirilir. Bu trafiğin, belli bir süre kabul edilmesine sebep olur ve tahmin edebileceğiniz gibi, zaman-aşımları olabilecektir ve olur.

## Reflexive ACL'ler

Bu ACL tipi, üst katman oturum bilgilerine bağlı olarak IP paketlerini filtreler ve genelde dışarı giden trafiğin geçmesine izin verir, fakat gelen trafiği sınırlandırır. Reflexive ACL'leri numaralı veya standart IP ACL'leri ya da diğer protokol ACL'leri ile tanımlayamazsınız. Diğer standart veya statik extended named ACL'leri ile birlikte kullanılabilmektedir. Fakat onlar sadece extended named IP ACL'leri ile tanımlanmaktadır.

## Time-based ACL'ler

Zaman-bağımlı ACL'ler, daha çok extended ACL'lerin gibi çalışır. Fakat ACL türü, tamamıyla zaman yönelimlidir. Basit olarak, günün veya haftanın belirli bir zamanını belirtebilir ve sonra bir görev ile ilgili isim vererek belirli bir periyodu tanımlarsınız. Bu nedenle, gerek duyulduğunda, ilgili fonksiyon, belirlediğiniz zaman kısıtlaması hangisiyse ona dahil olur. Zaman dilimi, router'ın saatine bağlıdır. Network Time Protocol (NTP) senkronizasyon ile birlikte kullanımını tavsiye ederim.

İşte bir örnek:

```
Corp#config t
Corp(config)#time-range no-http
Corp(config-time-range)#periodic we?
Wednesday  weekdays  weekend
Corp(config-time-range)#periodic weekend ?
  hh:mm  Starting time
Corp(config-time-range)#periodic weekend 06:00 to 12:00
Corp(config-time-range)#exit
Corp(config)#time-range tcp-yes
Corp(config-time-range)#periodic weekend 06:00 to 12:00
Corp(config-time-range)#exit
Corp(config)#ip access-list extended Time
Corp(config-ext-nacl)#deny tcp any any eq www time-range no-
http
Corp(config-ext-nacl)#permit tcp any any time-range tcp-yes
Corp(config-ext-nacl)#interface f0/0
Corp(config-if)#ip access-group Time in
Corp(config-if)#do show time-range
time-range entry: no-http (inactive)
   periodic weekdays 8:00 to 15:00
   used in: IP ACL entry
time-range entry: tcp-yes (inactive)
   periodic weekend 8:00 to 13:00
   used in: IP ACL entry
Corp(config-if)#
```

`time-range` komutu oldukça esnektir ve temel network erişimlerini veya mesai saatleri dışında internete girmelerini engellerseniz kullanıcıları sinirlendirirsiniz. Yukarıdaki komutları, list'leri, gerçek ağınızda uygulamadan önce bir test ortamında kontrol ettiğinizden emin olun.

## Remark'lar

Bu, `remark` anahtar sözcüğünün kullanımını kavramanız için gerçekten önemlidir. Çünkü size IP standart ve extended ACL'lerinizde yaptığınız kayıtlarla ilgili yorum veya çok sayıda açıklama kabiliyeti sağlayacaktır. Remark'lar, çok mükemmeldir. Çünkü ACL'lerinizi büyük oranda inceleme ve anlama kabiliyetinizi etkin bir şekilde artırır. Onlarsız, bu numaraların manasını hatırlamanızda size yardımcı olacak bir şey olmadan, anlamsız numaraların bataklığında tuzağa düşerdiniz.

Açıklamalarınızı bir `deny` veya `permit` ibaresinden önce ya da sonra yerleştirme seçeneğiniz olmasına rağmen, ben tamamıyla onları düzenli olarak hangi açıklamanın `permit` hangisinin `deny` ibaresi ile ilgili olduğu konusunda kafanızı karıştırmayacak şekilde yerleştirmenizi öneririm.

Bunun hem standart hem de extended ACL'ler için olması için `access-list access list number remark` ***remark*** global configuration komutunu kullanın. Şayet bir remark'tan kurtulmak isterseniz, sadece komutun `no` formunu kullanın.

`remark` komutunun nasıl kullanılacağı ile ilgili örnek şöyledir:

```
R2#config t
    R2(config)#access-list 110 remark Permit Bob from Sales Only
    To Finance
    R2(config)#access-list 110 permit ip host 172.16.10.1
    172.16.20.0 0.0.0.255
    R2(config)#access-list 110 deny ip 172.16.10.0 0.0.0.255
    172.16.20.0 0.0.0.255
    R2(config)#ip access-list extended No_Telnet
    R2(config-ext-nacl)#remark Deny all of Sales from Telnetting
    to Marketing
    R2(config-ext-nacl)#deny tcp 172.16.30.0 0.0.0.255
    172.16.40.0 0.0.0.255 eq 23
    R2(config-ext-nacl)#permit ip any any
    R2(config-ext-nacl)#do show run
    [output cut]
    !
    ip access-list extended No_Telnet
     remark Stop all of Sales from Telnetting to Marketing
     deny   tcp 172.16.30.0 0.0.0.255 172.16.40.0 0.0.0.255 eq
    telnet
     permit ip any any
    !
    access-list 110 remark Permit Bob from Sales Only To Finance
    access-list 110 permit ip host 172.16.10.1 172.16.20.0
    0.0.0.255
    access-list 110 deny   ip 172.16.10.0 0.0.0.255 172.16.20.0
    0.0.0.255
    !
```

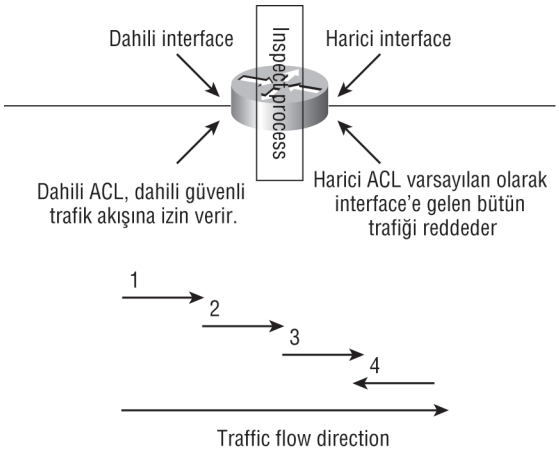
Hem extended hem de standart named access list'e bir remark ekleyebildim. Bununla beraber, bu açıklamaları, `show access-list` komutu çıktısında göremezsiniz, sadece `running-config`'te görebilirsiniz. Bu komutu, SDM kullanarak yapılandırdığımızda tekrar göstereceğim.

## Context-based Access Control (Cisco IOS Firewall)

Context-Based Access Control (CBAC) kullanımını sağlaması için IOS'ta ayarlı Cisco IOS Firewall'a sahip olmalısınız ve ikisini ayırt etmeyi bir yerlerden, hatta Cisco'dan bile çok nadir duyarsınız. İnsanlar sadece Cisco IOS Firewall olarak belirtirler. Peki, o nedir? CBAC'nin görevi, firewall'dan geçen trafiği dikkatle incelemektir. Böylece TCP ve UDP için oturum bilgilerini alabilir ve kontrol edebilir. Ve o, firewall'un access list'lerine geçici bir route oluşturup oluşturmamayı belirlemek için bir araya toplanan birçok bilgiyi kullanır.

Bunu yapmak için trafiğin aktığı aynı yöndeki `ip inspect` list'leri yapılandırmalısınız. Şayet bunu yapmazsanız, herhangi bir dönüş trafiği geri gelemeyecektir. Dahili ağın içinden üretilen oturum bağlantılarını olumsuz etkileyecektir.

Cisco IOS Firewall'un (CBAC) nasıl çalıştığını çok basit olarak gösteren Şekil 10.5'e bakın.




***Şekil 10.5:*** *Cisco IOS Firewall (CBAC) örneği.*

Cisco IOS firewall ile yapılandırılmış bir router, trafiği aşağıdaki şekilde işlemden geçirir:

1. İlk olarak, dahili ACL onaylarsa, router tüm dahili paketleri ona gönderecektir.
2. Sonra onaylanmış trafik, firewall'un `ip inspect` işleminden geçecektir. O, onaylı bağlantının durum bilgisini durum tablosuna ekler.
3. Son olarak trafik, IP inspect prosesine geçer. IP inspect prosesi dinamik bir ACL kaydı oluşturur ve dönüş trafiğinin tekrar router'dan geçişinin kabul edilmesi için onu harici ACL'e koyar.
4. SDM kullanarak bir firewall oluşturduğumda, bunu birazdan göstereceğim.

### Authentication Proxy

Bunu tüm router'larıma kurdum, fakat bunu yapabilmeniz için ayrıca Cisco IOS firewall özelliği kurulumuna sahip olmalısınız. Authentication Proxy, faydalı olduğundan, yapılandırma kurulumuna sahibim. Bu doğrudur, çünkü gelen kullanıcılar, giden kullanıcıları veya her ikisine de kimlik doğrulaması yapar. Normalde bir ACL tarafından bloklanacak olan kullanıcılar, firewall'a ulaşmak için bir browser kullanabilir ve sonra bir TACACS+ veya RADIUS sunucusunda kimlik doğrulaması yapabilirler.

## Access List'lerin Görüntülenmesi

Router'ın yapılandırmasının doğruluğunun kontrol edilebilmesi daima iyidir. Tablo 10.1, konfigürasyonu doğrulamak için kullanılabilen komutları listelemektedir.

**Tablo 10.2:** Access List Yapılandırmalarını Doğrulamak İçin Kullanılan Komutlar

| Komut | Etkisi |
|---|---|
| `show access-list` | Tüm access list'leri ve router'da ayarlanmış parametrelerini görüntüler. Bu komut list'in ayarlandığı interface'i göstermez. |
| `show access-list 110` | Sadece access list 110 için parametreleri gösterir. Bu komut list'in ayarlandığı interface'i göstermez. |
| `show ip access-list` | Sadece router'da ayarlanmış IP access list'leri gösterir. |
| `show ip interface` | Hangi interface'lerin ayarlı access list'lere sahip olduğunu gösterir. |
| `show running-config` | Access list'leri ve hangi interface'lerin ayarlı access list'lere sahip olduğunu gösterir. |
| `show mac access-group` | Tüm (sadece katman2 switch'lerde kullanılan) katman2 interface'lerine veya belirlenmiş katman2 interface'ine uygulanan MAC access list'lerini görüntüler. |

Show running-config komutunu ayrıca bir named access list'in, hem router'da hem de bir MAC access list'i olarak katman2 switch'te olduğunu doğrulamak için kullanmaktayız.

Show access-list komutu, hem router'daki tüm access list'leri hem de onların bir interface'e uygulanıp uygulanmadığını gösterir:

```
Lab_A#show access-list
Standard IP access list 10
    deny   172.16.40.0, wildcard bits 0.0.0.255
    permit any
Standard IP access list BlockSales
    deny   172.16.40.0, wildcard bits 0.0.0.255
    permit any
Extended IP access list 110
    deny tcp any host 172.16.30.5 eq ftp
    deny tcp any host 172.16.30.5 eq telnet
    permit ip any any
Lab_A#
```

İlk olarak, bu list'te, access list 10 ve named access list'imiz göründüğüne dikkat edin. İkinci olarak, access list 110'daki TCP portları için gerçek numaralar girdiğim halde show komutunun, okunabilirlik için TCP portları yerine protokol isimlerini verdiğine dikkat edin. (Herkes onları ezberleyemez ki!)

Show ip interface komutunun çıktısı aşağıdaki gibidir:

```
Lab_A#show ip interface e1
Ethernet1 is up, line protocol is up
  Internet address is 172.16.30.1/24
  Broadcast address is 255.255.255.255
  Address determined by non-volatile memory
  MTU is 1500 bytes
  Helper address is not set
  Directed broadcast forwarding is disabled
  Outgoing access list is BlockSales
  Inbound  access list is not set
  Proxy ARP is enabled
  Security level is default
  Split horizon is enabled
  ICMP redirects are always sent
  ICMP unreachables are always sent
  ICMP mask replies are never sent
  IP fast switching is disabled
  IP fast switching on the same interface is disabled
  IP Null turbo vector
  IP multicast fast switching is disabled
  IP multicast distributed fast switching is disabled
  Router Discovery is disabled
```

```
  IP output packet accounting is disabled
  IP access violation accounting is disabled
  TCP/IP header compression is disabled
  RTP/IP header compression is disabled
  Probe proxy name replies are disabled
  Policy routing is disabled
  Network address translation is disabled
  Web Cache Redirect is disabled
  BGP Policy Mapping is disabled
Lab_A#
```

Bu interface'deki dışarı giden access list'in, BlockSales olduğunu işaret eden koyu renkli satıra dikkat ettiğinize emin olun. Fakat inbound access list yoktur. Bir tane daha doğrulama komutu var, sonra SDM kullanarak firewall güvenliğini ayarlamaya geçeceğiz.

Daha önce belirttiğim gibi, tüm access list'leri görmek için `show running-config` komutunu kullanabilirsiniz. Bununla beraber, bir katman2 switch'te, interface ayarlarınızı `show mac access-group` komutu ile doğrulayabilirsiniz:

```
S1#sh mac access-group
Interface FastEthernet0/1:
   Inbound access-list is not set
   Outbound access-list is not set
Interface FastEthernet0/2:
   Inbound access-list is not set
   Outbound access-list is not set
S1#
```

Kaç tane interface'de MAC adres list'leri oluşturduğunuza bağlı olarak, belirli interface'lere göz atmak için `interface` komutunu kullanabilirsiniz:

```
S1#sh mac access-group interface f0/6
Interface FastEthernet0/6:
   Inbound access-list is Todd_MAC_List
   Outbound access-list is not set
```

Gelin, ağlarımızda güvenliği, SDM'in nasıl sağlayabileceğini tartışalım.

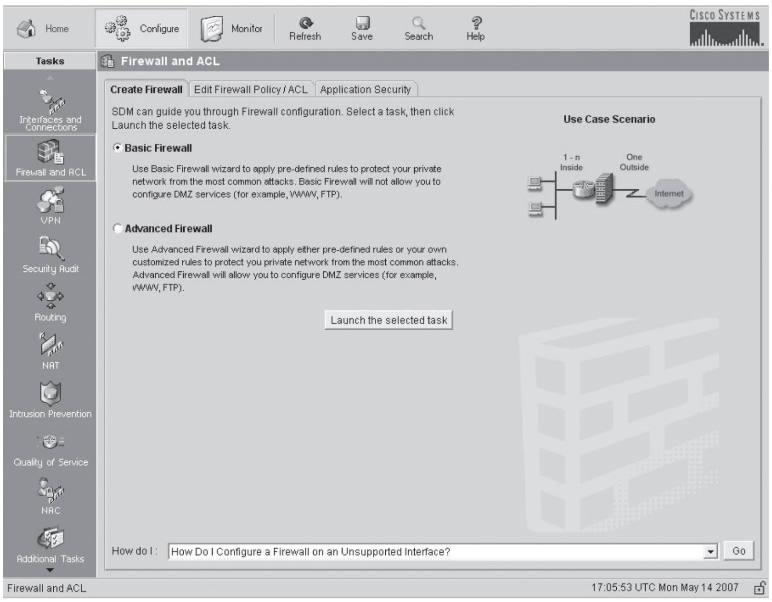
# SDM Kullanarak Access List'leri Yapılandırmak

Bu bölüme size SDM kullanarak bir access list'in nasıl oluşturulacağını göstererek başlayacağım ve sonra ACL'ler oluşturan Cisco IOS Firewall eklemek için firewall wizard'ını kullanacağım. Sadece wizard kullanmak onun en kolay yoludur. Size her iki yolu da göstereceğim. Fakat wizard kullanarak, çok fazla şey yapmanıza gerek yoktur, birkaç Next butonuna basmak, güvenli bir router oluşturacaktır.

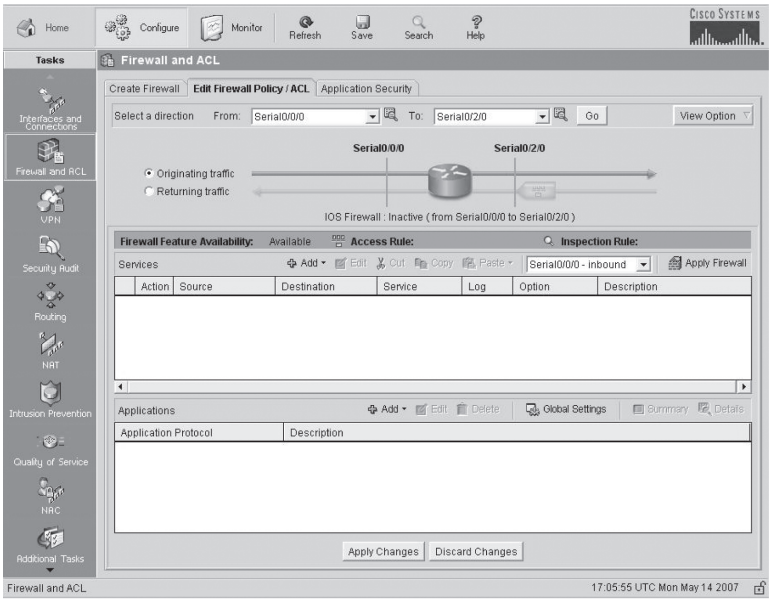
## ACL'leri SDM ile Oluşturmak

SDM kullanarak basitçe bir ACL oluşturmaya başlayalım. Sadece named access list'leri oluşturabilirsiniz. Gelin bir göz atalım.

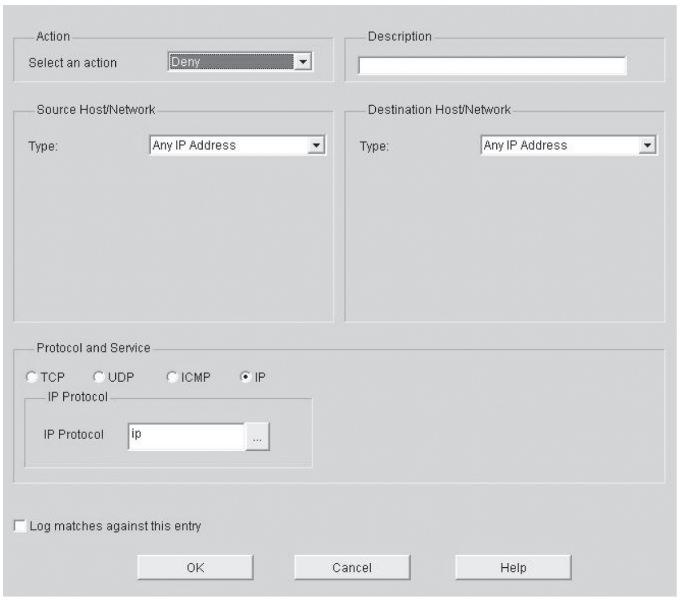
Açıkça, SDM kullanımındaki ilk adımınız, onu açmaktır. Sonra Configure ➢ Firewall and ACL'ye tıklayın. Create Firewall ekranına düşeceksiniz.



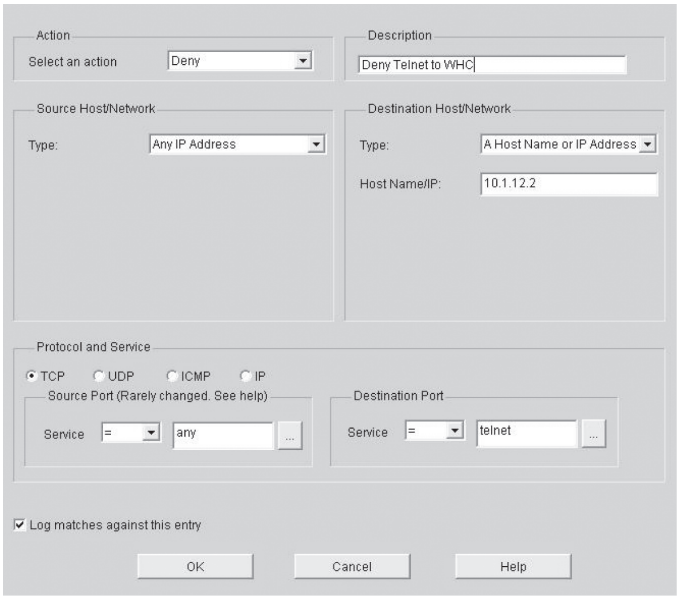

Sonra, Edit Firewall policy/ACL sekmesine tıklayın



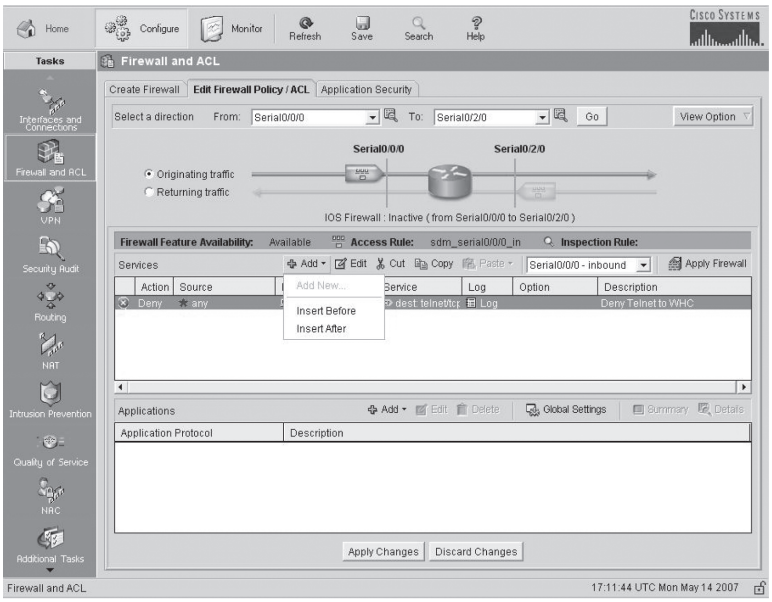

Yukarda, açılır menülerden From ve To'dan interface'leri seçin. Ben zaten From'dan s0/0/0 interface'ini, To interface'imden s0/2/0'ı seçtim. Sonra sayfanın ortasındaki, aşağı çekmeli menüyü sağlayan +Add butonuna tıkladım. Add New seçin, aşağıdaki ekran görünecektir.



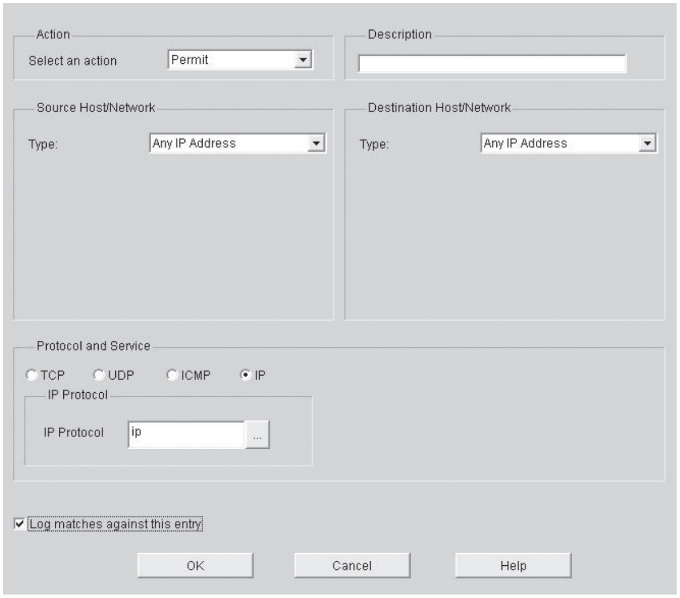

List'i Wireless HostC (WHC)'yi s0/0/0 interface'inden gelen herhangi bir host'tan telnet'i (23) reddetmesi için yapılandırdım ve ayrıca eşleşmeleri log'lamasını seçtim. Ok'i tıklayacağım ve sonra router interface'imi kapatmayacak bir `permit` komutu oluşturacağım.



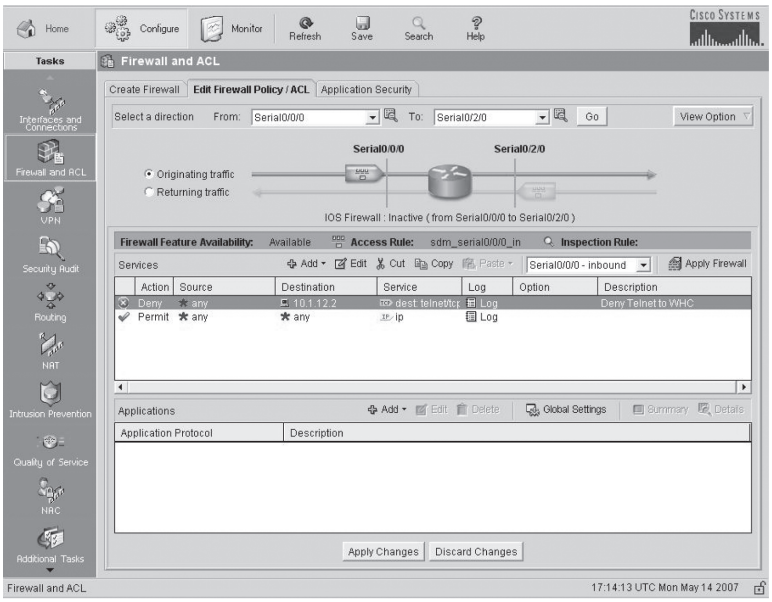

List'leri SDM üzerinden oluşturmakla ilgili güzel bir şey, +Add menüsünün yeni bir koşul komutu oluşturmak mı istediğinizi yoksa onu access list'te zaten sahip olduğunuz satırın önüne veya sonuna yerleştirmek mi istediğinizi sormasıdır. Bu çok güzeldir, çünkü SDM kullanarak, ACL'lerinizi çabuk ve etkili şekilde düzenleyebilirsiniz.




Sonra, basit bir `permit ip any` komutu kullanacağım.




OK butınuna tıkladıktan sonra, bana list'lerimi gösteren ana ekran geldi. Buradan kolayca ACL'lerimi ekleyip, silip yönetebiliyorum. Gerçekten çok hoş!




Şimdi router'ın sahip olduğu running-config'e bir bakalım:

```
!
ip access-list extended sdm_serial0/0/0_in
 remark SDM_ACL Category=1
 remark Deny Telnet to WHC
 deny   tcp any host 10.1.12.2 eq telnet log
 permit ip any any log
!
!
interface Serial0/0/0
 description 1st Connection to R1$FW_INSIDE$
 ip address 10.1.2.1 255.255.255.0
 ip access-group sdm_serial0/0/0_in in
```

**Looks good—let's try to telnet to host 10.1.12.2 and see what shows up on the Corp console:**

```
Corp#
*May 14 17:34:36.503: %SEC-6-IPACCESSLOGP: list sdm_
serial0/0/0_in denied tcp 10.1.2.2(30491) -> 10.1.12.2(23), 1
packet
```

**Okay—now I'm going to telnet to host 10.1.12.1:**

```
*May 14 17:34:53.023: %SEC-6-IPACCESSLOGP: list sdm_
serial0/0/0_in permitted  tcp 10.1.2.2(16774) ->
10.1.12.1(23), 1 packet
Corp#
```

Host 10.1.12.2 hedef IP adresi ve 23 hedef portu ile s0/0/0'a giren paketlere izin verilmemektedir. Fakat ben 10.1.12.1 host'una telnet yaptığımda kabul edildi.

Tüm bunların SDM kullanarak ne kadar kolay olduğunu ve router'ı işleyişte daha güvenli yaptığını göstereceğim.

## SDM ile Firewall'ların Oluşturulması

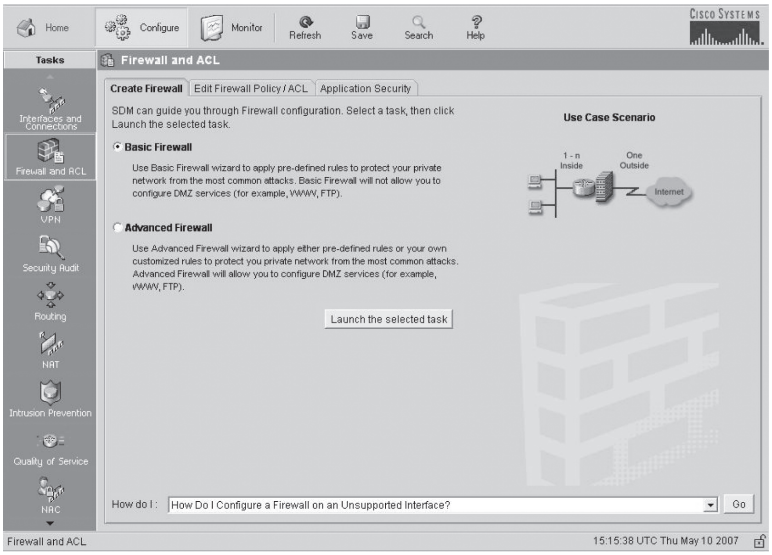
Bu bölüm, Cisco IOS Firewall yazılımı kurmak için, Basit ve Gelişmiş Firewall sihirbazlarının nasıl kullanılacağını gösterecek. O, router'larınızda güvenliği yapılandırdığınızda sizin için gerçekten en iyi seçenektir.

Configure ➢ Firewall and ACL'ye tıklayın. Bir Firewall kuralı oluşturma yöntemiyle sizi yönlendirecek bir wizard'a ulaşacaksınız. Fakat belirttiğim gibi, seçmek için iki wizard vardır:

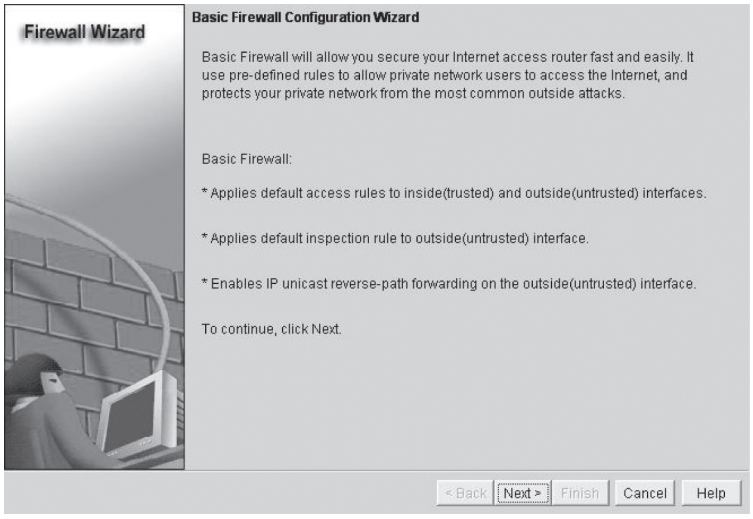
**Basic Firewall:** Şayet amacınız, sadece host'lardan oluşan ağa bağlanmak, sizi dış dünya (internet vs) ile bağlayan sunuculara bağlanmak değilse, kullanacağınız wizard budur. Basic Firewall'u seçtikten sonra sağda küçük bir diyagram görünecektir. Sadece Launch butonuna tıklayın.

**Advanced Firewall:** Bu sizin, hem host makineleri hem de İnternette bulunan dışarıdaki host'lara erişmek için ihtiyacınız olan sunucularınızın bulunduğu bir ağa bağlanmanız için kullanacağınız wizard'dır. Bu wizard'ı seçtikten sonra, sağda, ağınızı resmeden küçük bir diyagram göreceksiniz. Web, email veya İnternet üzerinden haberleşmeniz gereken diğer sunucularla iletişim için Advanced Firewall'u seçin ve sonra Launch butonuna tıklayın.

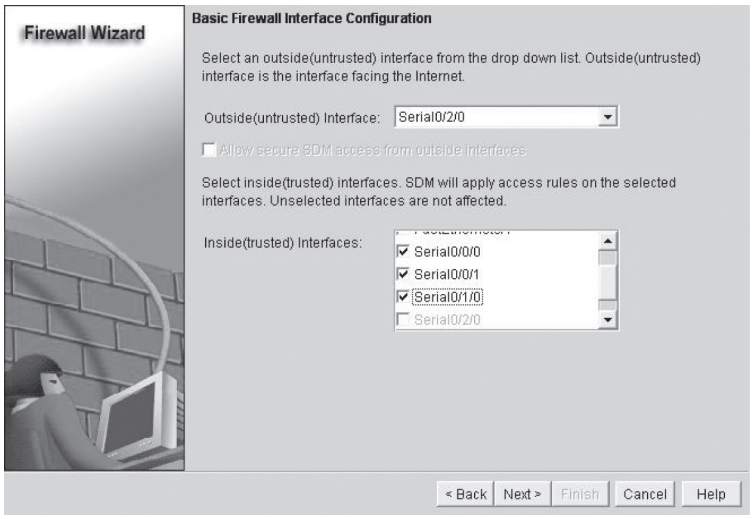
İlk ekran Create Firewall ekranıdır.



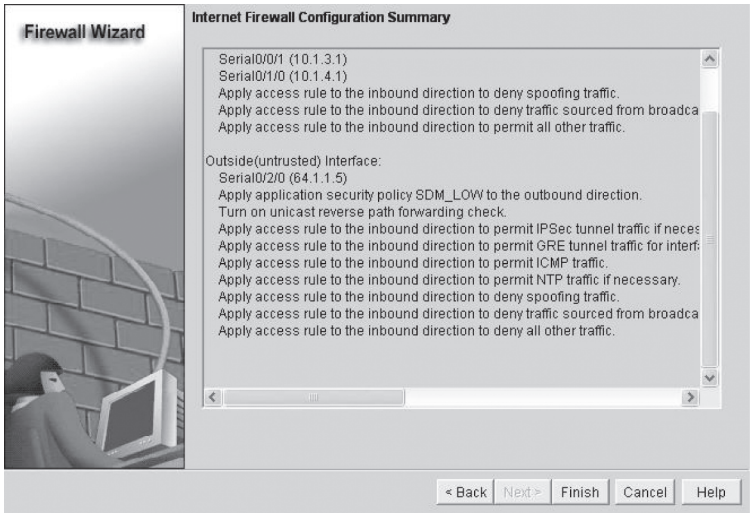

Buradan, connect and create a basic firewall seçiyorum. Sonraki ekran bana Basic Firewall Configuration Wizard'ın ne yapacağını söyler.



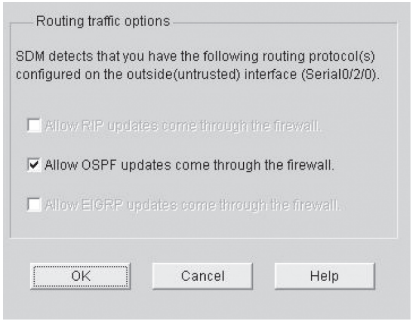
Her şeyi yapacak gibi görünüyor ve biliyor musunuz, o yapacak! Next butonuna tıkladım ve bu ekrandan iç ve dış (güvenilen ve güvenilmeyen) adreslerimi seçebiliyorum.




İç ve dış interface'lerimi seçmeyi tamamlayınca, Next'i tıkladım ve router'ıma bir ton access-list koşul komutu uygulandı.

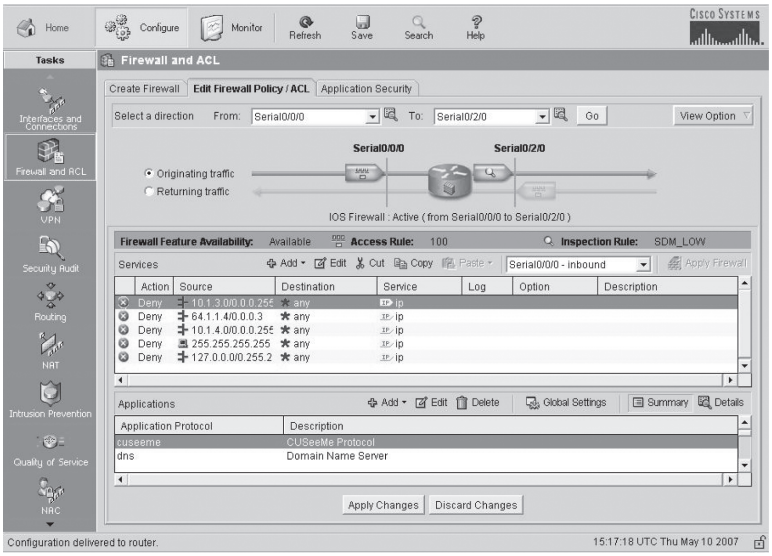



Sonra Finish butonuna tıkladım ve dış interface boyunca routing protokolü yapılandırmak isteyip istemediğimi sordu.




OK butonuna tıkladım. (Bana güvenin, bunu gerçekten bir ağda yapmak istemezsiniz.) Daha sonra router'ınızda firewall'u başarıyla kurdunuz ekranı geldi. Hayal mi kuruyorum yoksa bu gerçekten çalışacak mı?

Gelin görelim. OK butonuna tıkladıktan sonra, şu ekran geldi.




Router'ımda firewall kurmak için birkaç tıklama yetti. Fakat running-config'e ne koyduğunu gerçekten görmek istiyorum. Bunu kontrol edelim:

```
Corp#sh run
Building configuration...
[output cut]
!
ip inspect name SDM_LOW cuseeme
ip inspect name SDM_LOW dns
ip inspect name SDM_LOW ftp
ip inspect name SDM_LOW h323
ip inspect name SDM_LOW https
ip inspect name SDM_LOW icmp
ip inspect name SDM_LOW imap
ip inspect name SDM_LOW pop3
ip inspect name SDM_LOW netshow
ip inspect name SDM_LOW rcmd
ip inspect name SDM_LOW realaudio
ip inspect name SDM_LOW rtsp
ip inspect name SDM_LOW esmtp
ip inspect name SDM_LOW sqlnet
ip inspect name SDM_LOW streamworks
ip inspect name SDM_LOW tftp
ip inspect name SDM_LOW tcp
ip inspect name SDM_LOW udp
ip inspect name SDM_LOW vdolive
!
[output cut]
```

Basic Firewall Configuration Wizard, CBAC (Contex-Based Access Control) olarak da bilinen IOS firewall ekledi. Cisco onu sadece IOS Firewall olarak belirtir.

İp `inspect` komutu, her bireysel uygulama için atakları azaltmak için basit uygulama denetimini açmıştır.

Bu `inspect` komutunun her birinin, access list'e ayrı protokoller eklediğine dikkat edin. Wizard olmaksızın, incelenmesini istediğiniz protokolleri manuel girmeniz gerekir. SDM kullanarak onu yapmak çok daha kolay ve hala en iyi yoldur. Basic Firewall Confgiuration Wizard sizi iki interface ile sınırlandırır. Onun uygulandığı interface'leri kontrol edin:

```
!
interface Serial0/2/0
 description Connection to R3$FW_OUTSIDE$
 ip address 64.1.1.5 255.255.255.252
 ip access-group 103 in
 ip verify unicast reverse-path
 ip nat outside
 ip inspect SDM_LOW out
 ip virtual-reassembly
 clock rate 2000000
!
```

`ip inspect SDM_LOW out` kontrol prosesinin uygulandığı yerdir. Gelin SDM'in yaptığı, yapılandırmanın geri kalanına bakalım:

```
!
access-list 100 remark auto generated by SDM firewall
configuration
access-list 100 remark SDM_ACL Category=1
access-list 100 deny   ip 10.1.3.0 0.0.0.255 any
access-list 100 deny   ip 64.1.1.4 0.0.0.3 any
access-list 100 deny   ip 10.1.4.0 0.0.0.255 any
access-list 100 deny   ip host 255.255.255.255 any
access-list 100 deny   ip 127.0.0.0 0.255.255.255 any
access-list 100 permit ip any any
access-list 101 remark auto generated by SDM firewall
configuration
access-list 101 remark SDM_ACL Category=1
access-list 101 deny   ip 10.1.2.0 0.0.0.255 any
access-list 101 deny   ip 64.1.1.4 0.0.0.3 any
access-list 101 deny   ip 10.1.4.0 0.0.0.255 any
access-list 101 deny   ip host 255.255.255.255 any
access-list 101 deny   ip 127.0.0.0 0.255.255.255 any
access-list 101 permit ip any any
access-list 102 remark auto generated by SDM firewall
configuration
access-list 102 remark SDM_ACL Category=1
access-list 102 deny   ip 10.1.3.0 0.0.0.255 any
access-list 102 deny   ip 10.1.2.0 0.0.0.255 any
access-list 102 deny   ip 64.1.1.4 0.0.0.3 any
```

```
access-list 102 deny   ip host 255.255.255.255 any
access-list 102 deny   ip 127.0.0.0 0.255.255.255 any
access-list 102 permit ip any any
```

100-102 ACL'lerinin, bir iç interface'de içeriye doğru olduğuna dikkat edin. Bu ACL'ler, dışarıya gitmesine izin verilen trafiği ve inspect firewall'dan geçmesine izin verilen trafiği tanımlar. Her ACL, hem router'ın hakkında bilgisi olan tüm ağları hem de loopback adreslerini engeller.

Gelin SDM'in oluşturduğu son list'e bakalım:

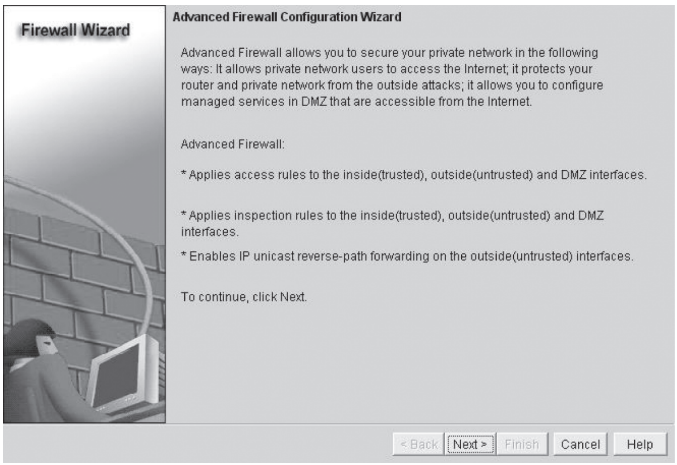
```
access-list 103 remark auto generated by SDM firewall
configuration
access-list 103 remark SDM_ACL Category=1
access-list 103 deny   ip 10.1.3.0 0.0.0.255 any
access-list 103 deny   ip 10.1.2.0 0.0.0.255 any
access-list 103 deny   ip 10.1.4.0 0.0.0.255 any
access-list 103 permit icmp any host 64.1.1.5 echo-reply
access-list 103 permit icmp any host 64.1.1.5 time-exceeded
access-list 103 permit icmp any host 64.1.1.5 unreachable
access-list 103 permit ospf any any
access-list 103 deny   ip 10.0.0.0 0.255.255.255 any
access-list 103 deny   ip 172.16.0.0 0.15.255.255 any
access-list 103 deny   ip 192.168.0.0 0.0.255.255 any
access-list 103 deny   ip 127.0.0.0 0.255.255.255 any
access-list 103 deny   ip host 255.255.255.255 any
access-list 103 deny   ip host 0.0.0.0 any
access-list 103 deny   ip any any log
!
```

Access list 103'ün, biraz fazla sayıda maddeyle paket reddettiğini görebilirsiniz. İlk olarak, bu dışınız içindir ve içerde bulunan tüm ağlar ve tüm özel adres aralıklarının girişi engellenmektedir.

Access list 103 bazı ICMP ve OSPF trafiğini kabul etmektedir, o kadar. Şimdi, güvenilen interface'den gelen trafik gönderilse dahi, dönüş trafiği engellenecektir. Fakat bu firewall prosesi içindir. Güvenli trafik gittiğinde, durum bilgisi, tabloya konacaktır ve dönüş trafiğinin engellenmesi yerine kabul edilmesi için bu ACL için dinamik bir ACL kaydı yazılacaktır.

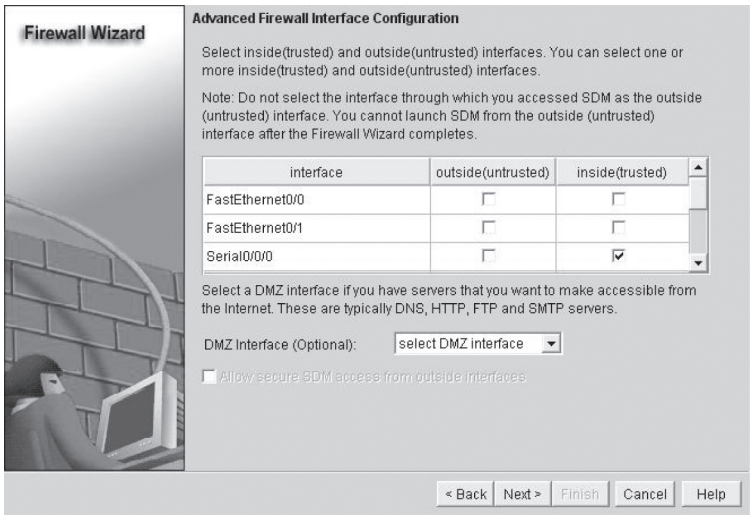
Güzel, bu bazen hayallerin gerçekleştiğini gösterir. Tüm yaptığım birkaç küçük butona basmak oldu. Bu modülü atlayabilecekmişim gibi görünmüş olabilir. Fakat access list kavramlarını gerçekten iyi anlamanız gerekmektedir ve bu bölüm size temel bazı bilgileri verdi.

Fakat hala Advanced Firewall Configuration Wizard'a hızlıca bakmamız gerekmektedir. Firewall and Configuration butonuna geri dönüp Advanced Firewall'u tıkladıktan sonra şu ekran geldi.




Advance ve Basic Firewall Wizard arasında birkaç değişiklik vardır. Gelişmiş sihirbaz, erişim kurallarını iç, dış ve DMZ interface'lerine uygular. Basic Firewall Wizard, sadece iç ve dış interface'lere uygular, DMZ interface'leri, Wizard kullanılarak yapılandırılmaz. Diğer bir farklılık, basit olan, inceleme kurallarını sadece dış interface'lere uygularken, gelişmiş olanın tüm iç, dış ve DMZ interface'lerine uygulamasıdır.

Şimdiki ekran bana iç, dış ve DMZ interface'lerimi seçmemi söyler.




Bunu yapınca, herhangi bir router dış interface'ini kullanarak SDM'e erişemezsiniz. Sihirbazın geri kalan kısmı, basit olanla nerdeyse aynı şekilde çalışır. Sadece router'ınızda, bu kitabın kapsamı dışında olan daha gelişmiş ayarlara ulaşırsınız. Bir SDM kopyasını indirmenizi ısrarla öneririm. Bu GUI ile mümkün olduğunca çok pratik yapın ve ona aşina olun!

Belirttiğim gibi, SDM, ACL, NAT VPN gibi gelişmiş yapılandırmalar için kullanışlıdır. Önceki çıktı, hangi interface'inizin iç, hangisinin dış olduğunu anlayarak, deneyimli bir uzman gibi ACL ve firewall güvenliğinizi yapılandırabileceğinizi göstermektedir.o

# Özet

Bu bölümde IP trafiğini düzgün bir şekilde filtrelemesi için standart access list'lerin nasıl yapılandırıldığından bahsettim. Bir standart access list'in ne olduğunu ve ağınızın güvenliğini artırmak için onun bir Cisco router'a nasıl uygulandığını öğrendiniz. IP trafiğini daha detaylı filtrelemesi için extended access list'lerin nasıl yapılandırıldığını da öğrendiniz. Ayrıca hem standart ve extended access list'ler arasındaki farklardan hem de onların Cisco router'lara nasıl uygulandığından bahsettim.

Daha sonra named access list'lerin nasıl yapılandırıldığını ve router'daki interface'lere nasıl uygulandıklarını göstermeye geçtim. Named access list'ler, kolayca fark edilebilme avantajı sağlarlar. Bundan dolayı, belirsiz numaralarla tanımlanan access list'lerden daha kolay yönetim olanağı verirler.

Hem router hem de bir switch'te seçili access list işlemlerinin nasıl görüntüleneceğini ve doğrulanacağını gördük. IP ve MAC access list'leri doğrulamak için kullanılan bazı görüntüleme komutlarını inceledik.

Son olarak, SDM kullanarak bir router için ACL'leri ve firewall politikaları oluşturmanın ne kadar kolay olduğunu gösterdim.

# Sınav Gereklilikleri

**Standart ve extended access-list numara aralıklarını hatırlamak:** Standart IP access list yapılandırmak için kullanabileceğiniz numara aralığı, 1-99 ve 1300-1999'dur. Bir extended IP access list için numara aralığı, 100-199 ve 2000-2699'dur.

**İmplicit deny terimini anlamak:** Her access list'in sonundaki, gizli bir deny'dır. Yani, bir paket, access list'teki satırların hiçbiri ile eşleşmezse, engellenecektir. Ayrıca hiçbir şey olmasa da, access list'inizdeki `deny` ibaresi sebebiyle access list, herhangi bir paketi kabul etmeyecektir.

**Standart IP access-list yapılandırma komutunu anlamak:** Bir standart IP access list yapılandırmak için, global configuration modda 1-99 veya 1300-1999 list numaralarını kullanın. `Permit` veya `deny` seçin, sonra bu modülde işlenen üç teknikten birini kullanarak filtrelemek istediğiniz kaynak IP adresini seçin.

**Extended IP access-list yapılandırma komutunu anlamak:** Bir extended IP access list yapılandırmak için, global configuration modda 100-199 veya 2000-2699 list numaralarını kullanın. Permit veya deny'ı, Network katmanı protokol alanını, filtrelemek istediğiniz kaynak IP adresini, filtrelemek istediğiniz hedef IP adresini ve son olarak (şayet seçiliyse) Transport katmanı port numarasını seçin.

**Bir router interface'indeki access list'i doğrulamak için kullanılan komutu hatırlamak:** Access list'in bir interface'de ayarlanıp ayarlanmadığını ve hangi yönü filtreleyeceğini görmek için, `show ip interface` komutunu kullanın. Bu komut size, sadece interface'e uygulanan access list'in içeriğini göstermeyecektir.

**Access-list yapılandırmasını doğrulamak için kullanılan komutu hatırlamak:** Router'ınızdaki yapılandırılmış access list'leri görmek için show `access-list` komutunu kullanın. Bu komut size hangi interface'lerin bir access list'e sahip olduğunu göstermeyecektir.

# Yazılı Lab 10.1

Bu bölümde, aşağıdaki soruların cevaplarını yazın:

1. 172.16.0.0 ağındaki tüm makinelerin, Ethernet ağınıza erişimini engellemek amacıyla bir standart access list yapılandırmak için hangi komutu kullanırsınız?
2. 1.ci sorudaki access list'i bir Ethernet interface'ine uygulamak için hangi komutu kullanırsınız?
3. 192.168.15.5 host'unun bir Ethernet ağına erişimine izin vermemek amacıyla bir access list oluşturmak için hangi komut kullanılır?
4. Access list'in doğru girildiğini doğrulamak için hangi komut kullanılır?
5. Hangi iki komut, access list'lerin Ethernet interface'lerine düzgün şekilde uygulandığını doğrular?

6. 172.16.10.1 host'unun, 172.16.30.5 host'una telnet yapmasını durduran bir extended access list oluşturmak için hangi komutu kullanırsınız?
7. Bir VTY line'da access list ayarlamak için hangi komutu kullanırsınız?
8. Soru 1'deki standart access list'i, bir named access list olarak yazın.
9. Soru 2'den, bir interface'de oluşturduğunuz named access list'i uygulayacağınız komutu yazın.
10. Hangi komut bir access list'in yerleşimini ve yönünü doğrular?

(Yazılı lab 10'un cevapları, bu bölüm için gözden geçirme sorularına cevaptan sonra bulunabilir.)

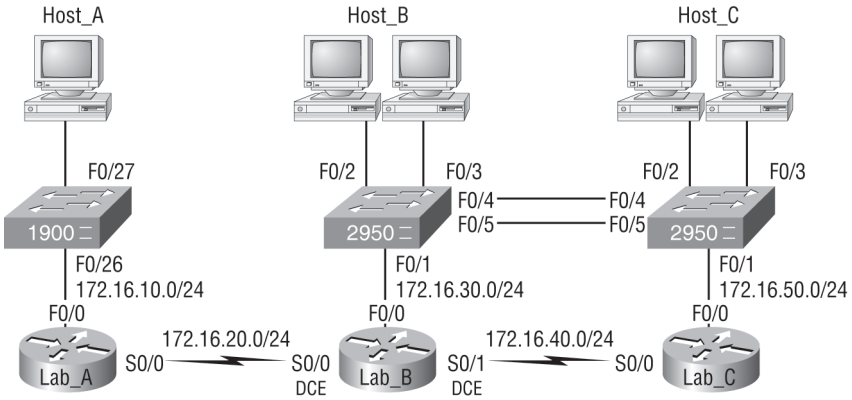
# Pratik Lab'lar

Bu bölümde, iki lab'ı tamamlayacaksınız. Bu lab'ları tamamlamak için en az üç router'a ihtiyacınız var. Şayet RouterSim veya Sybex yazılım programlarını kullanıyorsanız, lütfen bu programlarda bulunan lab'ları kullanın.

Lab 10.1: Standart IP Access List'ler

Lab 10.2: Extended IP Access List'ler

Bu lab'ların hepsi, router'ları yapılandırmak için aşağıdaki diyagramı kullanacaktır.




## Pratik Lab 10.1: Standart Access List'ler

Bu lab'da, sadece 172.16.30.0 ağından HostB'den gelen paketlerin 172.16.10.0 ağına girmesine izin vereceksiniz.

1. LabA'ya gidin ve `config t` yazarak global configuration moda girin.
2. Global configuration moddan, farklı access list'lerin tamamının bir listesini almak için `access-list` yazın.
3. Standart IP access list oluşturmanıza izin verecek bir access-list numarası seçin.
4. HostB'nin adresi olan 172.16.30.2'ye izin vermeyi seçin:

```
Lab_A(config)#access-list 10 permit 172.16.30.2 ?
  A.B.C.D  Wildcard bits
  <cr>
```

Sadece 172.16.30.2'yi belirtmek için, 0.0.0.0 wildcard'ı kullanın:

```
Lab_A(config)#access-list 10 permit 172.16.30.2
  0.0.0.0
```

5. Şimdi access list oluşturuldu, onu çalışır hale getirmek için bir interface'e uygulamalısınız:

```
Lab_A(config)#access-list 10 permit 172.16.30.2 ?
 A.B.C.D  Wildcard bits
 <cr>
```

Sadece 172.16.30.2'yi belirtmek için 0.0.0.0 widcard'ı kullanın:

```
Lab_A(config)#access-list 10 permit 172.16.30.2
  0.0.0.0
```

6. Aşağıdaki komutlarla access list'inizi doğrulayın:

```
Lab_A#sh access-list
Standard IP access list 10
    permit 172.16.30.2
Lab_A#sh run
[output cut]
interface FastEthernet0/0
ip address 172.16.10.1 255.255.255.0
ip access-group 10 out
```

7. Access list'inizi, HostB (172.16.30.2)'den HostB (172.16.10.2)'yi pingleyerek test edin.
8. LabB ve LabC'den HostA (172.16.10.2)'yı pingleyin. Access list'iniz doğruysa pinglemenin başarısız olması gerekir.

## Pratik Lab 10.2: Extended IP Access List'ler

Bu lab'da 172.16.10.2 host'unun, router LabB (172.16.20.2)'ye bir Telnet oturumu oluşturmasını engellemek için bir extended IP access list kullanacaksınız. Bununla beraber, host hala Lab_B router'ını ping'leyebilmelidir. IP extended list'ler, kaynağa yakın yerleştirilmelidir. Bu nedenle extended list'i Lab_A router'a ekleyin.

1. Lab_A'daki tüm bir access list'leri silin ve Lab_B'ye bir extended access list ekleyin.
2. Bir extended access list oluşturmak için numara seçin. IP extended list'ler, 100-199 veya 2000-2699 kullanırlar.
3. Bir deny komutu kullanın (diğer trafiklerin hala çalışması için 7. Adımda bir permit komutu ekleyin)

```
Lab_A(config)#access-list 110 deny ?
 <0-255>  An IP protocol number
 ahp      Authentication Header Protocol
 eigrp    Cisco's EIGRP routing protocol
 esp      Encapsulation Security Payload
 gre      Cisco's GRE tunneling
 icmp     Internet Control Message Protocol
 igmp     Internet Gateway Message Protocol
 igrp     Cisco's IGRP routing protocol
 ip       Any Internet Protocol
 ipinip   IP in IP tunneling
 nos      KA9Q NOS compatible IP over IP tunneling
 ospf     OSPF routing protocol
```

```
pcp      Payload Compression Protocol
tcp      Transmission Control Protocol
udp      User Datagram Protocol
```

4. Telnet'i iptal edeceğinizden, Transport katman protokolü olarak TCP'yi seçin:

```
Lab_A(config)#access-list 110 deny tcp ?
 A.B.C.D  Source address
 any      Any source host
 host     A single source host
```

5. Filtrelemek istediğiniz kaynak IP adresini ekleyin ve sonra hedef host IP adresini ekleyin. Wildcard bit'leri yerine `host` komutunu kullanın.

```
Lab_A(config)#access-list 110 deny tcp host
 172.16.10.2 host 172.16.20.2 ?
 ack          Match on the ACK bit
 eq           Match only packets on a given port
              number
 established  Match established connections
 fin          Match on the FIN bit
 fragments    Check fragments
 gt           Match only packets with a greater
              port number
 log          Log matches against this entry
 log-input    Log matches against this entry,
              including input interface
 lt           Match only packets with a lower port
              number
 neq          Match only packets not on a given
              port number
 precedence   Match packets with given precedence
              value
 psh          Match on the PSH bit
 range        Match only packets in the range of
              port numbers
 rst          Match on the RST bit
 syn          Match on the SYN bit
 tos          Match packets with given TOS value
 urg          Match on the URG bit
 <cr>
```

6. Bu noktada, 172.16.10.2 host'unun 172.16.20.2'ye telnet yapmasını filtrelemek için `eq telnet` komutunu kullanın. Komutun sonunda log komutu kullanılabilir. Böylece access-list satırı kullanıldığında, konsolda bir log üretilecektir.

```
Lab_A(config)#access-list 110 deny tcp host
 172.16.10.2 host 172.16.20.2 eq telnet log
```

7. Bir permit ifadesi oluşturmak için sona bu satırı eklemek önemlidir (0.0.0.0 255.255.255.255'in `any` komutuyla aynı olduğunu hatırlayın).

```
Lab_A(config)#access-list 110 permit ip any 0.0.0.0
 255.255.255.255
```

   Bir `permit` deyimi oluşturmalısınız. Sadece bir `deny` kullanırsanız, hiçbir şeye izin verilmeyecektir. Permit komutuyla ilgili daha detaylı bilgi için bu modülde önceki bölümlere bakın lütfen.

8. İlk router interface'ine gelir gelmez, Telnet trafiğini durdurmak için, Lab_A'daki FastEthernet0/0'a access-list'i uygulayın.

```
Lab_A(config)#int f0/0
Lab_A(config-if)#ip access-group 110 in
Lab_A(config-if)#^Z
```

9. 172.16.20.2 hedef IP adresini kullanarak 172.16.10.2 host'undan, Lab_A'ya telnet yapmaya çalışın. Lab_A'nın konsolunda, aşağıdaki mesajın üretilmesi gerekir. Bununla beraber ping komutu çalışmalıdır:

```
From host 172.16.10.2: C:\>telnet 172.16.20.2
```

   Lab_A'nın konsolunda, bu aşağıdaki gibi görünür:

```
01:11:48: %SEC-6-IPACCESSLOGP: list 110 denied tcp
  172.16.10.2(1030) -> 172.16.20.2(23), 1 packet
01:13:04: %SEC-6-IPACCESSLOGP: list 110 denied tcp
  172.16.10.2(1030) -> 172.16.20.2(23), 3 packets
```

# Gözden Geçirme Soruları

*Aşağıdaki sorular, bu modülün materyallerini anladığınızı test etmek için tasarlanmıştır. İlave sorulara nasıl sahip olacağınızla ilgili daha fazla bilgi için lütfen bu kitabın Giriş bölümüne bakın.*

NOT

1. Aşağıdakilerden hangisi bir standart access list örneğidir?

   A. `access-list 110 permit host 1.1.1.1`

   B. `access-list 1 deny 172.16.10.1 0.0.0.0`

   C. `access-list 1 permit 172.16.10.1 255.255.0.0`

   D. `access-list standard 1.1.1.1`

2. 192.168.160.0 ile 192.168.191.0 network aralığındaki host'ları engelleyecek bir access list oluşturmanız gerekmektedir. Aşağıdaki access list'lerden hangisini kullanırsınız?

   A. `access-list 10 deny 192.168.160.0 255.255.224.0`

   B. `access-list 10 deny 192.168.160.0 0.0.191.255`

   C. `access-list 10 deny 192.168.160.0 0.0.31.255`

   D. `access-list 10 deny 192.168.0.0 0.0.31.255`

3. Blocksales adında bir named access list oluşturdunuz. Router'ınızın s0 interface'ine girmeye çalışan paketlere bunu uygulamak için geçerli komut aşağıdakilerden hangisidir?

   A. `(config)#ip access-group 110 in`

   B. `(config-if)#ip access-group 110 in`

   C. `(config-if)#ip access-group Blocksales in`

   D. `(config-if)#blocksales ip access-list in`

4. Bir IP access list'inde sadece 172.16.30.55 host'unu belirtmenin geçerli yolu aşağıdakilerden hangisidir? (İki şık seçin)

   A. `172.16.30.55 0.0.0.255`

   B. `172.16.30.55 0.0.0.0`

   C. `any 172.16.30.55`

   D. `host 172.16.30.55`

   E. `0.0.0.0 172.16.30.55`

   F. `ip any 172.16.30.55`

5. Aşağıdaki access list'lerden hangisi, sadece HTTP trafiğine 196.15.7.0 ağı için izin verecektir?

   A. `access-list 100 permit tcp any 196.15.7.0 0.0.0.255 eq www`

   B. `access-list 10 deny tcp any 196.15.7.0 eq www`

   C. `access-list 100 permit 196.15.7.0 0.0.0.255 eq www`

   D. `access-list 110 permit ip any 196.15.7.0 0.0.0.255`

   E. `access-list 110 permit www 196.15.7.0 0.0.0.255`

6. Hangi router komutu, belirli bir interface'de bir access list'in etkin olup olmadığını belirlemenize izin verir?

   A. `show ip port`

   B. `show access-lists`

   C. `show ip interface`

   D. `show access-lists interface`

7. Hangi router komutu tüm access list'lerin içeriğini görmenize izin verir?

   A. `Router#show interface`

   B. `Router>show ip interface`

   C. `Router#show access-lists`

   D. `Router>show all access-lists`

8. Sadece 192.168.10.0 ağı için tüm Telnet bağlantılarını engellemek istiyorsunuz. Hangi komutu kullanabilirsiniz?

   A. `access-list 100 deny tcp 192.168.10.0 255.255.255.0 eq telnet`

   B. `access-list 100 deny tcp 192.168.10.0 0.255.255.255 eq telnet`

   C. `access-list 100 deny tcp any 192.168.10.0 0.0.0.255 eq 23`

   D. `access-list 100 deny 192.168.10.0 0.0.0.255 any eq 23`

9. 200.200.10.0 ağından, 200.199.11.0 ağına FTP erişimini engellemek, diğer bütün trafiğe izin vermek istiyorsunuz. Aşağıdaki komut topluluklarından hangisi geçerlidir?

   A. `access-list 110 deny 200.200.10.0 to network 200.199.11.0 eq ftpaccess-list 111 permit ip any 0.0.0.0 255.255.255.255`

   B. `access-list 1 deny ftp 200.200.10.0 200.199.11.0 any any`

   C. `access-list 100 deny tcp 200.200.10.0 0.0.0.255 200.199.11.0 0.0.0.255 eq ftp`

   D. `access-list 198 deny tcp 200.200.10.0 0.0.0.255 200.199.11.0 0.0.0.255 eq ftpaccess-list 198 permit ip any 0.0.0.0 255.255.255.255`

10. 172.16.50.1/20 subnet host'larını engelleyecek bir standart access list oluşturmak istiyorsunuz. Access list'inize aşağıdakilerden hangisi ile başlamalısınız?

    A. `access-list 10 deny 172.16.48.0 255.255.240.0`

    B. `access-list 10 deny 172.16.0.0 0.0.255.255`

    C. `access-list 10 deny 172.16.64.0 0.0.31.255`

    D. `access-list 10 deny 172.16.48.0 0.0.15.255`

11. Bir router interface'ine bir access list uygulamak için hangi komutu kullanırsınız?

    A. `ip access-list 101 out`

    B. `access-list ip 101 in`

    C. `ip access-group 101 in`

    D. `access-group ip 101 in`

12. 172.16.198.94/19 subnet host'larını engelleyecek bir standart access list oluşturmak istiyorsunuz. Access list'inize aşağıdakilerden hangisi ile başlamalısınız?

    A. `access-list 10 deny 172.16.192.0 0.0.31.255`

    B. `access-list 10 deny 172.16.0.0 0.0.255.255`

    C. `access-list 10 deny 172.16.172.0 0.0.31.255`

    D. `access-list 10 deny 172.16.188.0 0.0.15.255`

13. 172.16.144.17/21 subnet host'larını engelleyecek bir standart access list oluşturmak istiyorsunuz. Access list'inize aşağıdakilerden hangisi ile başlamalısınız?

    A. `access-list 10 deny 172.16.48.0 255.255.240.0`

    B. `access-list 10 deny 172.16.144.0 0.0.7.255`

    C. `access-list 10 deny 172.16.64.0 0.0.31.255`

    D. `access-list 10 deny 172.16.136.0 0.0.15.255`

14. Aşağıdakilerden hangi komut, access list 110'u inbound olarak ethernet0 interface'ine bağlar?

    A. `Router(config)#ip access-group 110 in`

    B. `Router(config)#ip access-list 110 in`

    C. `Router(config-if)#ip access-group 110 in`

    D. `Router(config-if)#ip access-list 110 in`

15. Hangi komut SMTP trafiğini sadece 1.1.1.1 host'u için mümkün kılar?

    A. `access-list 10 permit smtp host 1.1.1.1`

    B. `access-list 110 permit ip smtp host 1.1.1.1`

    C. `access-list 10 permit tcp any host 1.1.1.1 eq smtp`

    D. `access-list 110 permit tcp any host 1.1.1.1 eq smtp`

16. Aşağıdaki access list'i oluşturdunuz:

```
access-list 110 deny tcp 10.1.1.128 0.0.0.63 any eq smtp
access-list 110 deny tcp any eq 23
int ethernet 0
ip access-group 110 out
```

    Bu access list neye sebep olacaktır?

    A. Email ve Telnet, E0'dan çıkışı kabul edilecektir.

    B. Email and Telnet'in E0'dan gelmesine izin verilecektir.

    C. Email ve Telnet dışında, E0'dan her şeyin gitmesine izin verilecektir.

    D. Hiçbir IP trafiğin E0'dan gitmesine izin verilmeyecektir.

17. Aşağıdaki komut serisinden hangisi router'a Telnet erişimini kısıtlayacaktır?

    A.
```
Lab_A(config)#access-list 10 permit 172.16.1.1 Lab_A(config)#line
con 0
Lab_A(config-line)#ip access-group 10 in
```

    B.
```
Lab_A(config)#access-list 10 permit 172.16.1.1 Lab_A(config)#line
vty 0 4
Lab_A(config-line)#access-class 10 out
```

    C.
```
Lab_A(config)#access-list 10 permit 172.16.1.1 Lab_A(config)#line
vty 0 4
Lab_A(config-line)#access-class 10 in
```

    D.
```
Lab_A(config)#access-list 10 permit 172.16.1.1 Lab_A(config)#line
vty 0 4
Lab_A(config-line)#ip access-group 10 in
```

18. Aşağıdakilerden hangisi, access list'lerin bir interface'e uygulanması hakkında doğrudur?

    A. Bellek dolana kadar, bir interface istediğiniz kadar access list yerleştirebilirsiniz.

    B. Bir interface'e sadece bir tane access list uygulayabilirsiniz.

    C. Bir interface'de yapılandırılmış her katman3 protokolü için bir yöne, bir access list ayarlayabilirsiniz.

    D. Bir interface'e iki access list uygulayabilirsiniz.

19. Belirli fonksiyonlara erişimi kısıtlayan, yetkili seviyeler kuran bir router'da çalışıyorsunuz. `Show running-configuration` komutunu çalıştıramadığınızı fark ettiniz. Router'ınızdaki Ethernet 0 interface'ine uygulanan access list'leri nasıl görüp, doğrulayabilirsiniz?

    A. `show access-lists`

    B. `show interface Ethernet 0`

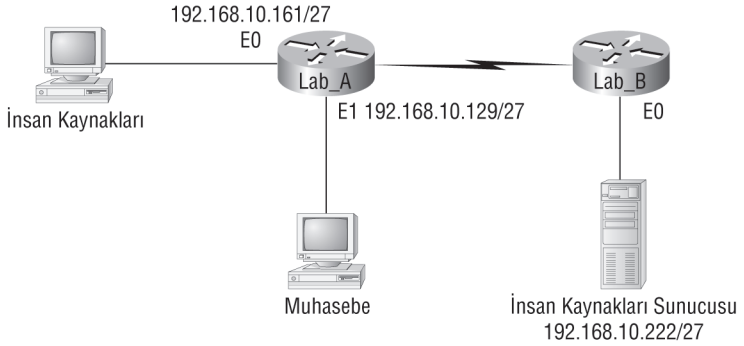
    C. `show ip access-lists`

    D. `show ip interface Ethernet 0`

20. Muhasebe LAN'ı kullanıcılarının, İnsan Kaynakları sunucusuna erişim hakları olmasını istemiyorsunuz. Aşağıdaki access list oluşturuldu:

    ```
    Access-list 10 deny 192.168.10.128 0.0.0.31
    Access-list 10 permit any
    ```

    Aşağıdaki diyagrama göre, muhasebe kullanıcılarının, Lab_B'nin E0 interface'ine bağlı ağa erişimini engellemek için access list hangi router'ın hangi interface'ine ve hangi yönde yerleştirilmelidir?




    A. Lab_A, S0 out

    B. Lab_A, E1 in

    C. Lab_A, E1 out

    D. Lab_B, S1 in

    E. Lab_B, E0 out

    F. Lab_B, E0 in

# Gözden Geçirme Sorularının Cevapları

1. B Standart IP access list'leri 1-99 ve 1300-1999 numaralarını kullanır ve sadece kaynak IP adres bazında filtreleme yapar. C şıkkı yanlıştır, çünkü mask, wildcard formatında olmalıdır.
2. C 192.168.160.0'dan 192.168.191.0'a kadar olan aralık, 32 blok boyutundadır. Network adresi, 192.168.160.0'dır ve mask 255.255.224.0 olmalıdır. Bir access list için 0.0.31.255 wildcard format'ında olmalıdır. 31, 32 blok boyutu için kullanılmaktadır. Wildcard, daima blok boyutundan bir düşüktür.
3. C Bir named access list kullanarak, router interface'ine list'i uyguladığınızda kullanılan numarayı kullanın. `ip access-group Blocksales in` doğrudur.
4. B,D wildcard 0.0.0.0 router'a dört oktetin tamamının eşleşmesini söyler. Bu wildcard versiyonu, `host` komutu yerine kullanılabilir.
5. A Bu tür sorularda kontrol edilecek ilk şey, access-list numarasıdır. Bir standart access list kullandığından, ikinci şık yanlıştır. Kontrol edilecek ikinci şey, protokoldür. Şayet üst-katman protokolü ile filtreleme yapıyorsanız, ya UDP ya da TCP kullanmalısınız. Bu dördüncü şıkkı eler. Üçüncü ve son şıklar doğru söz dizimine sahiptir.
6. C Sadece `show ip interface` komutu size, hangi interface'lerin access list'e sahip olacağını söyleyecektir. `Show access-lists`, hangi interface'lerin access list'e sahip olduğunu göstermez.
7. C `show access-list` komutu, tüm access list'lerin içeriğini görmenize izin verecektir. Fakat access list'lerin uygulandığı interface'leri göstermeyecektir.
8. C Extended access list aralığı 100-199 ve 2000-2699'dur, bu nedenle 100 access-list numarası geçersizdir. Telnet TCP kullanır, öyleyse TCP protokolü geçerlidir. Şimdi sadece kaynak ve hedef adreslerine bakmanız gerekir. Sadece üçüncü şık, parametrelerin doğru sıralamasına sahiptir. Cevap B çalışabilir, fakat soru sadece 192.168.10.0 ağını belirtmektedir ve cevap B'deki wildcard çok geniştir.
9. D Extended access list aralığı 100-199 ve 2000-2699'dur ve kaynak ve hedef IP adresleri, protokol numarası ve port numarası temelinde filtreler. Son şık, `permit ip any any` 'i belirten ikinci satırdan dolayı doğrudur. (`any` seçeneği ile aynı olan `0.0.0.0 255.255.255.255`'i kullandım). Üçüncü şık buna sahip değildir, bu nedenle erişime izin vermeyecektir, fakat kalanlara da izin vermeyecektir.
10. D ilk olarak, /20'nin, üçüncü oktette 16 blok boyuta sahip 255.255.240.0 olduğunu bilmelisiniz. 16'ları sayarak, subnetimizi üçüncü oktette 48 yapar ve üçüncü oktet için wildcard, daima blok boyutundan bir düşük olmasından dolayı 15 olacaktır.
11. C Bir access list uygulamak için, uygun komut, `ip access-group 101 in`'dir.
12. A İlk olarak, üçüncü oktette 32 blok boyuta sahip 255.255.224.0 olduğunu bilmelisiniz. 32'leri sayarak, subnetimizi üçüncü oktette 192 yapar ve üçüncü oktet için wildcard, daima blok boyutundan bir düşük olmasından dolayı 31 olacaktır.
13. İlk olarak, üçüncü oktette 8 blok boyuta sahip 255.255.248.0 olduğunu bilmelisiniz. 8'leri sayarak, subnetimizi üçüncü oktette 144 yapar ve üçüncü oktet için wildcard, daima blok boyutundan bir düşük olmasından dolayı 7 olacaktır.
14. Bir interface'e access-list yerleştirmek için, interface configuration modda `ip access-group` kullanın.
15. D Bir access-list sorusuna en iyi cevabı bulmaya çalıştığınızda, daima access-list numarasını ve sonra port numarasını kontrol edin. Bir üst-katman protokolüne filtreleme yapıldığında, bir extended list (numaraları 100-199 ve 2000-2699) kullanmalısınız. Ayrıca, bir üst-katman protokolüne filtreleme yapıldığında, protokol alanında tcp veya udp kullanmalısınız. Şayet protokol alanında ip olursa, üst-katman protokollerini filtreleyemezsiniz. SMTP, TCP kullanır.

16. D Bir interface'e bir access list eklerseniz ve en az bir permit ifadeniz olmazsa, her list'in sonundaki gizli `deny any`'den dolayı, etkin olarak interface'inizi kapatırsınız.
17. C Router'a Telnet erişimi, router'ın VTY line'ı giriş yönüne bir standart veya extended IP access list kullanarak kısıtlanmaktadır. `Access-class` komutu, VTY line'lara access list'leri uygulamak için kullanılır.
18. C bir Cisco router, bir router interface'inizdeki access list'lerin yerleştirmesi hakkında kurallara sahiptir. Bir interface'de yapılandırılmış her katman3 protokol için her yöne bir access list koyabilirsiniz.
19. D Bir interface'e hangi access list'ler uygulandığını gösteren tek komut, `show ip interface Ethernet 0`'dır. `Show access-lists`, yapılandırılmış tüm access list'leri görüntüler ve `show ip access-lists`, tüm IP access list'leri görüntüler, fakat iki komutta, görüntülenen access list'lerin bir interface'e uygulanıp uygulanmadığını belirtmez.
20. E Bir standart access list'inde, access list, mümkün olduğu kadar hedefe yakın yerleştirilir. Bu örnekte, bu Lab_B router'ının Ethernet 0'ın çıkışıdır.

# Yazılı Lab 10.1'i Cevapları

1. access-list 10 deny 172.16.0.0 0.0.255.255

   ```
   access-list 10 permit any
   ```

2. ip access-group 10 out
3. access-list 10 deny host 192.168.15.5

   ```
   access-list 10 permit any
   ```

4. show access-lists
5. show running-config

   ```
   sh ip interface
   ```

6. access-list 110 deny tcp host

   ```
   172.16.10.1 host 172.16.30.5 eq 23
   access-list 110 permit ip any any
   ```

7. line vty 0 4

   ```
   access-class 110 in
   ```

8. ip access-list standard No172Net

   ```
   deny 172.16.0.0 0.0.255.255
   permit any
   ```

9. ip access-group No172Net out
10. show ip interfaces

# 11 Network Address Translation (NAT)

# 11 Network Address Translation (NAT)

- NAT'ı Ne Zaman Kullanırız?
- Network Address Translation Tipleri
- NAT İsimleri
- NAT Nasıl Çalışır?
- NAT'ı Test Etmek ve Hata Tespiti Yapmak
- Ağ Topluluğumuzda NAT Yapılandırmak
- SDM Kullanarak NAT Yapılandırmak
- Özet
- Sınav Gereklilikleri
- Yazılı Lab 11
- Pratik Lab'lar
- Lab 11.1: NAT İçin Hazırlanmak
- Lab 11.2: Dinamik NAT'ı Yapılandırmak
- Lab 11.3: PAT'ı Yapılandırmak
- Gözden Geçirme Soruları
- Gözden Geçirme Sorularının Cevapları
- Yazılı Lab 11'in Cevapları

# Network Address Translation (NAT)

Bu bölümde, Network Address Translation (NAT), Dynamic NAT ve NAT Overload olarak da bilinen Port Address Translation (PAT) konusundan bahsedeceğim. Tabii ki kitap boyunca kullandığım ağ topluluğunda NAT'ı göstereceğim ve sonra NAT'ı kolay yolla nasıl yapılandırılabileceğinizi görebilmeniz için SDM kullanarak bu bölümü tamamlayacağım.

NAT yapılandırmalarımızda access list'ler kullanmamız gerektiğinden, bu bölümü okumadan önce, bölüm 10'a göz atmanız faydalı olacaktır.

*Bu bölüm ile ilgili son güncellemeler için www.lammle.com ve/veya www.sybex.com adreslerine bakınız.*

NOT

## NAT'ı Ne Zaman Kullanırız?

Classless Inter-Domain Routing'e (CIDR) benzer şekilde NAT'ın orijinal amacı; genel IP adreslerinden bazı küçük numaralarla belirtilen birçok özel IP adresini kabul ederek, geçerli IP adres uzayının tükenmesini azaltmaktır.

Daha sonra NAT'ın, ağın taşınması ve birleştirilmesi, server yük paylaşımı ve sanal sunucular oluşturulmasında da kullanışlı bir araç olduğu keşfedildi. Bu nedenle, bu bölümde NAT işlevselliğinin temellerini ve NAT için yaygın terminolojileri açıklayacağım.

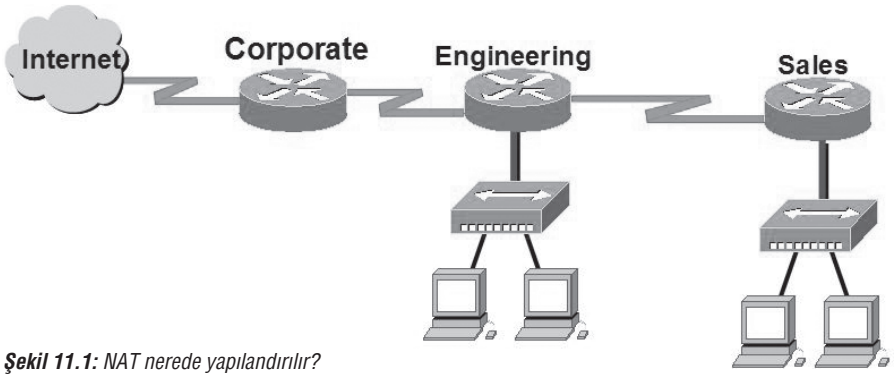
Zaman zaman NAT, network ortamınızda gerekli genel (public) IP adres sayısını gerçekten düşürmektedir. Dahili adresleme planını çoğaltan iki firma birleştiğinde, NAT gerçekten çok maharetlidir. Bir şirket Internet service provider'ını (ISP) değiştirdiğinde ve ağ müdürü, dahili adres planının değişmesiyle ilgili bir sıkıntı yaşamak istemediğinde de çok güzel bir araçtır.

NAT'a sahip olmanızın çok iyi olduğu durumların listesi şöyledir:

- İnternete bağlanmanız gerekmektedir ve host'larınız, global olarak benzersiz IP adreslerine sahip değildir.
- Yeni bir ISP'ye geçtiniz ve ağınızı yeniden adreslendirmeniz gerekmektedir.
- Adreslerin çoğaldığı iki intranet'inizi birleştirmeniz gerekmektedir.

NAT'ı tipik olarak, bir sınır router'da (border router) kullanırsınız. Bunun gösterildiği Şekil 11.1'e bakın.

"NAT tamamıyla çok güzel. Çok mükemmel bir network aracı ve ona sahip olmalıyım" diye düşünebilirsiniz. Bir dakika bekleyin. Aslında NAT kullanımıyla ilgili bazı ciddi problemler vardır. Beni yanlış anlamayın: NAT gerçekten bazen size kazandırabilir, fakat bilmeniz gereken bazı karanlık noktalar da vardır. NAT kullanımıyla ilgili lehte ve aleyhteki görüşlerin görselliği için Tablo 11.1'i kontrol edin.




***Şekil 11.1:*** *NAT nerede yapılandırılır?*

*NAT'la ilgili en belirgin avantaj, sizin resmi kayıtlı adres planlamanızı korumasıdır. Bundan dolayı IPv4 adreslerimiz tükenmemektedir.*

NOT

**Tablo 11.1:** NAT Uygulamanın Avantaj ve Dezavantajları

| Avantajlar | Dezavantajlar |
|---|---|
| Yasal olarak register edilen adresleri korur. | Dönüştürme, anahtarlama yol gecikmelerini ortaya çıkartır. |
| Adres çakışmalarını azaltır. | Uçtan-uca IP izlenebilirliğin kaybı. |
| İnternete bağlanıldığında, esnekliği artırır. | Belirli uygulamalar, NAT'ın etkin olmasıyla çalışamayacaktır. |
| Network değişikliğinde adreslerin yeniden numaralanmasını ortadan kaldırır. | |

# Network Address Translation Tipleri

Bu bölümde, sizinle üç NAT tipini inceleyeceğim:

**Statik NAT:** Bu NAT tipi, yerel ve global adresler arasında bire bir eşleştirme sağlamak için tasarlanmıştır. Statik versiyonun, ağınızdaki her host için bir gerçek IP adresi gerektirdiğini unutmayın.

**Dinamik NAT:** Bu versiyon size kayıtlı olmayan bir IP adresini, kayıtlı IP adreslerin havuzu dışında kayıtlı bir IP adresi ile eşleştirme kabiliyeti verir. Statik NAT'taki gibi, bir dahili adresi, bir dış adresle eşleştirmek için router'ınızı statik olarak yapılandırmanıza gerek yoktur. Fakat herkesin internete paketlerini gönderip alabilmesi için yeterli gerçek IP adreslerine sahip olmanız gerekmektedir.

**Overloading:** Bu, NAT yapılandırmasının en tutulan türüdür. Overloading, farklı portlar kullanarak, çok sayıda kayıtlı olmayan IP adresini, tek bir kayıtlı IP adresi ile eşleştiren Dinamik NAT'ın bir türüdür. Bu neden bu kadar özeldir? Çünkü Port Address Translation (PAT) olarak da bilinmektedir. Ve PAT (NAT Overload) kullanarak, sadece bir gerçek IP adresi ile binlerce kullanıcıyı internete bağlayabilirsiniz. Gerçekten, NAT Overload, internetteki geçerli IP adreslerin tükenmemesinin gerçek sebebidir.

NOT *Üzülmeyin, bu üç NAT tipinin nasıl yapılandırıldığını bölümün sonuna doğru göstereceğim.*

# NAT İsimleri

NAT ile kullandığımız adresleri açıklamak için kullandığımız isimler çok basittir. NAT çevirilerinden sonra kullanılan adresler, global adresler olarak belirtilir. Bunlar genellikle, internette kullanılan genel adreslerdir. Fakat internete çıkmayacaksanız, genel adreslere ihtiyacınız olmadığını hatırlayın.

NAT çevirisinden önce kullandığımız adresler lokal adreslerdir. Outside lokal adres, hedef host'un adresi olurken, inside lokal adres aslında, internete çıkmaya çalışan host'un özel adresidir. İkincisi çoğunlukla genel (web adresi, mail sunucusu vs.) bir adrestir ve paketin yolculuğuna başladığı adrestir.

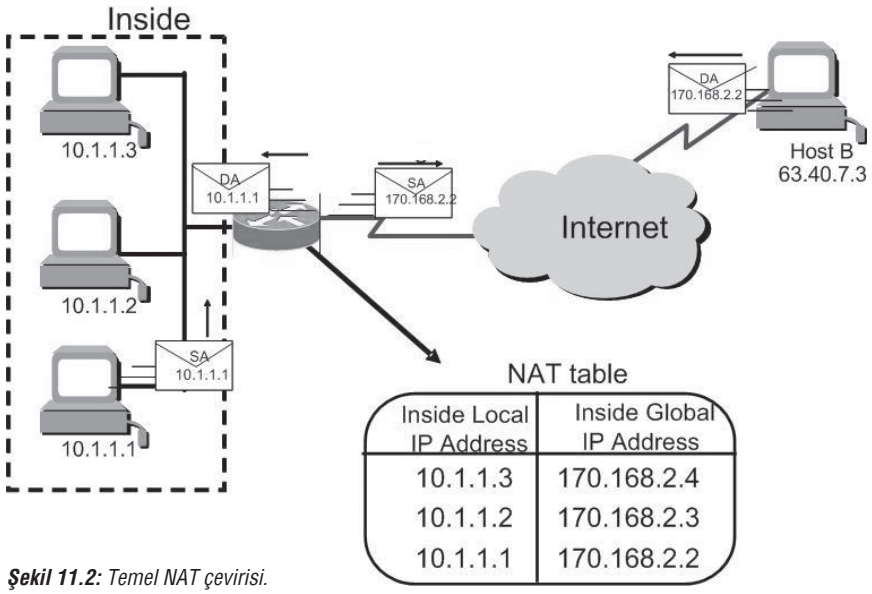
Çeviriden sonra inside lokal adres, inside global adres olarak tanımlanır ve outside global adres, hedef host'un ismi olur. NAT'la kullanılan çeşitli isimleri netleştirmek için tüm bu terminolojiyi listeleyen Tablo 11.2'yi inceleyin.

**Tablo 11.2:** NAT Terimleri

| İsimler | Anlamları |
|---|---|
| Inside local | Çeviriden önceki dahili kaynak adresinin adı. |
| Outside local | Çeviriden önceki hedef host'un adı |
| Inside global | Çeviriden sonraki dahili host'un adı |
| Outside global | Çeviriden sonraki dış hedef host'un adı |

# NAT Nasıl Çalışır?

NAT'ın nasıl çalıştığına bakma zamanı geldi. NAT'ın temel dönüştürmesini açıklamak için Şekil 11.2'i kullanarak başlayacağım.



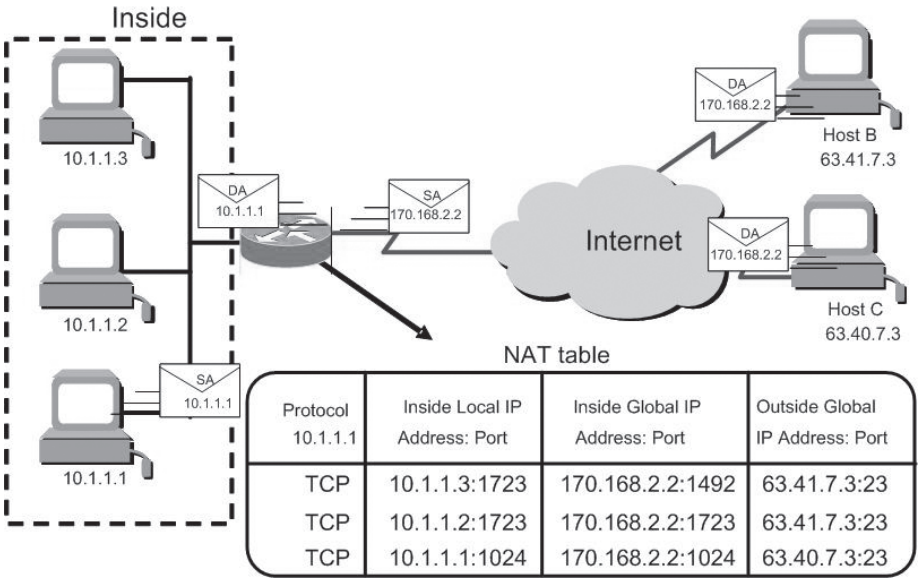

***Şekil 11.2:*** *Temel NAT çevirisi.*

Şekil 11.2'de gösterilen örnekte, host 10.1.1.1, NAT ile yapılandırılmış sınır router'a outbound paket gönderir. Router, dış bir network için hedeflenen adresi, bir inside lokal adres olarak tespit eder, çevirir ve çevriyi NAT tablosuna kaydeder.

Paket, çevrilmiş yeni kaynak adresiyle dış interface'e gönderilir. Harici host, paketi hedef host'a geri gönderir ve NAT router, inside global IP adresini, NAT tablosunu kullanarak tekrar inside lokal IP adresine dönüştürür. Bu kadar basittir.

Overloading (PAT-Port Address Translation) kullanarak daha karmaşık konfigürasyonlara bakalım. İnternetteki geçerli IP adreslerin tükenmemesinin sebebi, PAT kullanımıdır.

Tekrar Şekil 11.3'deki NAT tablosuna bakalım. İnside lokal IP adres ve outside global IP adreslerine ilave olarak şimdi port numaralarına sahibiz. Bu port numaraları router'ın, host'un dönüş trafiğini almasını belirlemesine yardımcı olur.




***Şekil 11.3:*** *NAT overloading (PAT) örneği.*

Port numaraları, bu örnekte lokal host'ları belirlemek için Transport katmanında kullanılmaktadır. Kaynak host'ları belirlemek için statik NAT denilen IP adreslerini kullanmak zorunda olsaydık, adreslerin tamamını tüketirdik. PAT, host'ları tanımlamak için Transport katmanını kullanmamıza izin verir. Bu, gerçek bir adresle, teorik olarak 65000 host'u mümkün kılar.

## Statik NAT Yapılandırması

Basit bir temel statik NAT yapılandırmasına bir göz atalım:

```
ip nat inside source static 10.1.1.1 170.46.2.2
!
interface Ethernet0
 ip address 10.1.1.10 255.255.255.0
 ip nat inside
!
interface Serial0
 ip address 170.46.2.1 255.255.255.0
 ip nat outside
!
```

Yukarıdaki router çıktısında `ip nat inside source` komutu, hangi IP adreslerin çevrileceğini belirler. Bu yapılandırma örneğinde `ip nat inside source` komutu, 10.1.1.1 inside lokal IP adresi ile 170.46.2.2 outside global IP adresi arasında statik bir çeviri yapılandırır.

Konfigürasyonun daha aşağılarına bakarsak, her interface'in altında bir `ip nat` komutu olduğunu görürüz. `ip nat inside` komutu, interface'i inside interface olarak tanımlar. `ip nat outside` komutu interface'i outside interface olarak tanımlar. Tekrar `ip nat inside source` komutuna baktığınızda komutun inside interface'i kaynak veya çevirinin başlangıç noktası olarak işaret ettiğini görürsünüz. Komut şöyle de kullanılabilirdi: `ip nat outside source`. Bu outside interface olarak atadığınız interface'i, kaynak veya çeviri için başlangıç noktası olarak belirtir.

## Dinamik NAT Yapılandırması

Dinamik NAT, içerdeki bir grup kullanıcıya, gerçek IP adresleri sağlamak için kullanacağımız bir adres havuzuna sahibiz anlamına gelir. Port numaraları kullanmayız. Bu nedenle yerel ağın dışına çıkmaya çalışan her kullanıcı için gerçek IP adresine sahip olmak zorundayız.

Aşağıda bir dinamik NAT yapılandırması örneği vardır:

```
ip nat pool todd 170.168.2.2 170.168.2.254
    netmask 255.255.255.0
ip nat inside source list 1 pool todd
!
interface Ethernet0
 ip address 10.1.1.10 255.255.255.0
 ip nat inside
!
interface Serial0
 ip address 170.168.2.1 255.255.255.0
 ip nat outside
!
access-list 1 permit 10.1.1.0 0.0.0.255
!
```

`ip nat inside source list 1 pool todd` komutu router'a, `todd` isimli IP NAT havuzunda bulunan adresleri `access-list 1` ile eşleşen IP adreslerine çevirmesini söyler. Bu örnekteki access list, bizim güvenlik sebepleriyle trafiği filtrelemek amacıyla kullandığımız şekilde izin vermek veya vermemek için kullanılır. İlgili trafik, access list ile eşleştiğinde, ACL çeviri yapması için NAT prosesini başlatır. Bu access list'lerin yaygın kullanışıdır. ACL'ler daima, sadece bir interface'deki trafiği bloklama işi yapmaz.

`ip nat pool todd 170.168.2.2 170.168.2.2` komutu, NAT gerektiren bu kullanıcıların dağıtılacağı adres havuzunu oluşturur.

## PAT (Overloading) Yapılandırması

Bu son örnek, inside global adres overloading'in nasıl yapılandırıldığını gösterir. Bu günümüzde kullandığımız tipik NAT'tır. Örneğin bir sunucuyu statik olarak eşleştirmediğimiz müddetçe, statik veya dinamik NAT'ı çok nadir kullanırız.

Aşağıda bir PAT yapılandırması ile ilgili örnek vardır:

```
ip nat pool globalnet 170.168.2.1 170.168.2.1
   netmask 255.255.255.0
ip nat inside source list 1 pool globalnet overload
!
interface Ethernet0/0
 ip address 10.1.1.10 255.255.255.0
 ip nat inside
!
interface Serial0/0
 ip address 170.168.2.1 255.255.255.0
```

```
 ip nat outside
!
access-list 1 permit 10.1.1.0 0.0.0.255
```

PAT'la ilgili güzel şey, bu yapılandırma ile önceki dinamik NAT yapılandırması arasındaki tek farkın, havuzumuzun tek IP adresine sahip olması ve `ip nat inside source` komutunun sonuna `overload` gelmesidir.

Kullanmamız için havuzdaki tek IP adresin, outside interface'inin IP adresi olduğuna dikkat edin. NAT Overload'u evde kendiniz için veya ISP'den sadece bir IP adresi almış küçük bir ofis için yapılandırıyorsanız bu mükemmeldir. Bununla beraber, şayet adres sizin için uygunsa, 172.168.2.2 gibi ilave bir adresi de kullanabilirdiniz. Bu, dışarıda overload edilen birden fazla IP adresine sahip olmanız gereken, çok sayıda dahili kullanıcınızın olabileceği çok geniş uygulamalarda faydalı olabilir.

## NAT'ın Basit Doğrulanması

Kullanacağınız (tipik olarak PAT) NAT tipini yapılandırınca, konfigürasyonunuzu doğrulamanız gerekmektedir.

Basit IP adres çeviri bilgisine bakmak için aşağıdaki komutu kullanın:

```
Router#show ip nat translation
```

IP NAT çevirilerine bakıldığında, aynı host için hedefteki aynı host'a çok sayıda çeviri görebilirsiniz. Bu tipik olarak Web için birçok bağlantının olmasıdır.

İlave olarak, `debug ip nat` komutu ile NAT yapılandırmanızı doğrulayabilirsiniz. Bu çıktı her debug satırında gönderilen adresi, çeviriyi ve hedef adresini gösterecektir:

```
Router#debug ip NAT
```

NAT kayıtlarınızı, çeviri tablosundan nasıl silersiniz? `clear ip nat translation` komutunu kullanın. NAT tablosundaki tüm kayıtları silmek için komutun sonunda yıldız işaretini (*) kullanın.

# NAT'ı Test Etmek ve Hata Tespiti Yapmak

Cisco'nun NAT'ı konfigürasyonların oldukça basit olmasından dolayı, fazla efor gerektirmeden size ciddi kuvvet sağlar. Fakat hepimiz hiçbir şeyin mükemmel olmadığını biliyoruz. Bu nedenle, bu olayda bazı yanlışlar vardır. Olası problemlerin listesini inceleyerek, yaygın bazılarının sebeplerini anlayabilirsiniz:

- Dinamik havuzların, doğru adres aralığıyla oluşturulduğunu kontrol edin.
- Dinamik havuzların çakışıp çakışmadığını kontrol edin.
- Adreslerin statik eşleşmede kullanılıp kullanılmadığını ve bunun dinamik havuz çakışmasına neden olup olmadığını kontrol edin.
- Access list'inizin çeviri için doğru adresleri belirttiğinden emin olun.
- Orada olması gereken bütün adreslerin dahil edildiğinden ve orda olmaması gereken hiçbir adresin dahil edilmediğinden emin olun.
- Düzgün şekilde sınırlanmış iç ve dış interface'leriniz olduğuna emin olmak için kontrol edin.

Yeni bir NAT yapılandırmasıyla ilgili en yaygın problemlerden biri, tamamen NAT'a özgü değildir. O genellikle routing aksamasıdır. Bu nedenle, bir paketteki kaynak veya hedef adresini değiştirdiğiniz için router'ın çeviriden sonra yeni adresi ne yapacağını bildiğinden emin olun.

NAT tablosunun tutabileceği eşleşme sayısının sonsuz olduğu varsayılır. Fakat gerçekte hafıza, CPU veya geçerli adres tipleri ya da portlar tarafından oluşturulan sınırlar düşünüldüğünde mümkün olan kayıt sayısında limit olmak zorundadır. Her NAT eşleşmesi, hafızadan 160 byte harcar. Ve bazen (çok sık değil) kayıt sayısı, performans uğruna veya politika kısıtlamalarından dolayı sınırlanmalıdır. Bu gibi durumlarda yardım için `ip nat translation max-entries` komutunu kullanın.

Hata tespiti için diğer bir kullanışlı komut, `ip nat statistics`'dir. Bunu çalıştırmak size NAT yapılandırmasının bir özetini verecektir ve aktif çeviri tipi sayısını verecektir. Hesaplananlar hem mevcut bir eşleşmeye işaret eder hem de kayba. İkincisi bir eşleşme oluşturma girişimine sebep olacaktır. Dinamik havuzları, tiplerini, toplam geçerli adresleri, kaç adresin tahsis edildiğini, kaçının başarısız olduğunu ve meydana gelen çevirilerin sayısını kontrol etmek isterseniz, `pool (refcount)` komutunu kullanın.

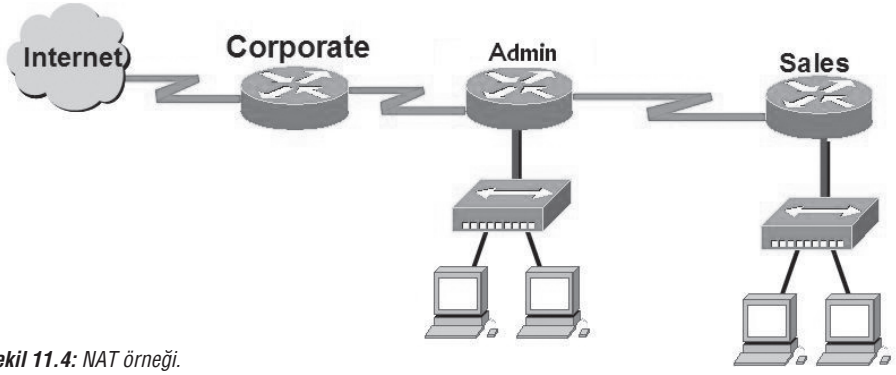
Dinamik NAT kayıtlarını, NAT tablosundan manuel olarak silebileceğinizi biliyor musunuz? Süresi dolana kadar beklemeden belirli bir bozuk kayıttan kurtulmanız gerektiğinde oldukça kullanışlı gelebilir. Bir adres havuzunu yeniden yapılandırmak için tüm NAT tablosunu temizlemek istediğinizde, manuel temizleme oldukça kullanışlıdır.

Ayrıca, Cisco IOS yazılımının, havuzdaki adreslerden biri NAT tablosuyla eşleşirse adres havuzlarını değiştirmeniz ve silmenize izin vermeyeceğini bilmeniz gerekir. `clear ip nat translations` komutu kayıtları siler. Global ve lokal adres yardımıyla ve TCP ile UDP çevirileri (portlar) boyunca tek bir kayıt belirtebilir veya tüm tabloyu yok etmek için sadece yıldız işareti (*) kullanabilirsiniz. Bu komut statik kayıtları silmediğinden, bunu yaparsanız sadece dinamik kayıtlar silinecektir.

Yani, başlangıç eşleşmeleri NAT tablosunda tutulmalı ki, belirli bir bağlantıdan ulaşan tüm paketler, sürekli olarak çevrilsinler. Kayıtların NAT tablosunda tutulması, düzenli olarak aynı paketlerin, aynı dış hedeflere paketleri gönderdiği her sefer oluşan arama tekrarlarını azaltacaktır.

Demek istediğim şudur: Kayıt, ilk olarak NAT tablosuna yerleştirildiğinde bir timer çalışmaya başlar. Bu timer'ın süresi `translation timeout` olarak bilinir. Verilen bir kayıt için bir paketin router tarafından çevrildiği her sefer, timer sıfırlanır. Şayet timer'ın süresi dolarsa kayıt, NAT tablosundan silinecektir ve dinamik olarak atanan adres havuza geri dönecektir. Cisco'nun varsayılan çeviri zaman aşımı 86,400 saniyedir (24 saat), ancak siz bunu `ip nat translation timeout` komutu ile değiştirebilirsiniz.

Yapılandırmayla ilgili bölüme ve hakkında bahsettiğim komutları kullanmaya geçmeden önce, birkaç NAT örneğini inceleyelim ve kullanmanız gereken yapılandırmaları yapıp yapamayacağınızı görelim. Başlamak için Şekil 11.4'e bakın ve kendinize iki şeyi sorun: bu tasarımda NAT'ı nereye uygularsınız ve hangi çeşit NAT yapılandırırsınız?



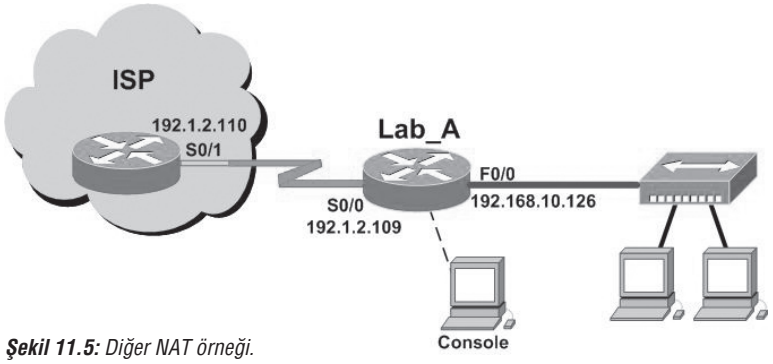

***Şekil 11.4:*** *NAT örneği.*

Şekil 11.4'de, NAT yapılandırması corporate router'a yerleştirilmelidir ve konfigürasyon, overload ile dinamik NAT olmalıdır (PAT). Bu örnekte, hangi tür NAT kullanılmaktadır?

```
ip nat pool todd-nat 170.168.10.10 170.168.10.20 netmask
255.255.255.0
```

Yukarıdaki komut dinamik NAT kullanır. Komuttaki pool cevabı ortaya çıkarır, artı havuzda birden fazla adres vardır. Bu muhtemelen PAT kullanmadığımız anlamına gelir. Şimdiki NAT örneğinde, gerekli olan konfigürasyonu anlayıp anlamadığımızı görmek için Şekil 11.5'i kullanacağız.

Şekil 11.5'teki örnek, NAT yapılandırılması gereken ve 192.1.2.109'dan 114'e kadar 6 genel IP adresi kullanılmasına izin verecek bir sınır router'ı göstermektedir. Bununla beraber iç ağda, 192.168.10.65'den 126'ya kadar özel adres kullanan 63 host'unuz vardır. Sınır router'ındaki NAT yapılandırmanız ne olacaktır?




***Şekil 11.5:*** *Diğer NAT örneği.*

Burada iki farklı cevap çalışacaktır, fakat benim tercihim aşağıdakidir:

```
ip nat pool Todd 192.1.2.109 192.1.2.109 netmask 255.255.255.248
access-list 1 permit 192.168.10.64 0.0.0.63
ip nat inside source list 1 pool Todd overload
```

`ip nat pool Todd 192.1.2.109 192.1.2.109 netmask 255.255.255.248` komutu, havuzun adını `Todd` olarak ayarlar ve NAT'ın 192.1.2.109 adresini kullanması için dinamik bir adres havuzu oluşturur. `Netmask` komutu yerine prefix-length 29 ibaresini kullanabilirsiniz (düşündüğünüzü biliyorum, ama bunu router interface'inde de kullanamazsınız). İkinci cevap, inside global olarak sadece 192.1.2.109'a sahip olmakla aynı neticeyle sonlanacaktır, fakat şunu yazabilirsiniz ve yine çalışacaktır: `ip nat pool Todd 102.1.2.109 192.1.2.114 netmask 255.255.255.248`. İkinciden altıya kadar olan adreslerin, sadece bir TCP port numarası çakışması olması halinde kullanılacağından bu işe yaramaz.

Şayet access-list'in ayarlandığı ikinci satırı anlamadıysanız, Modül 10 "Güvenlik" e bakın lütfen.

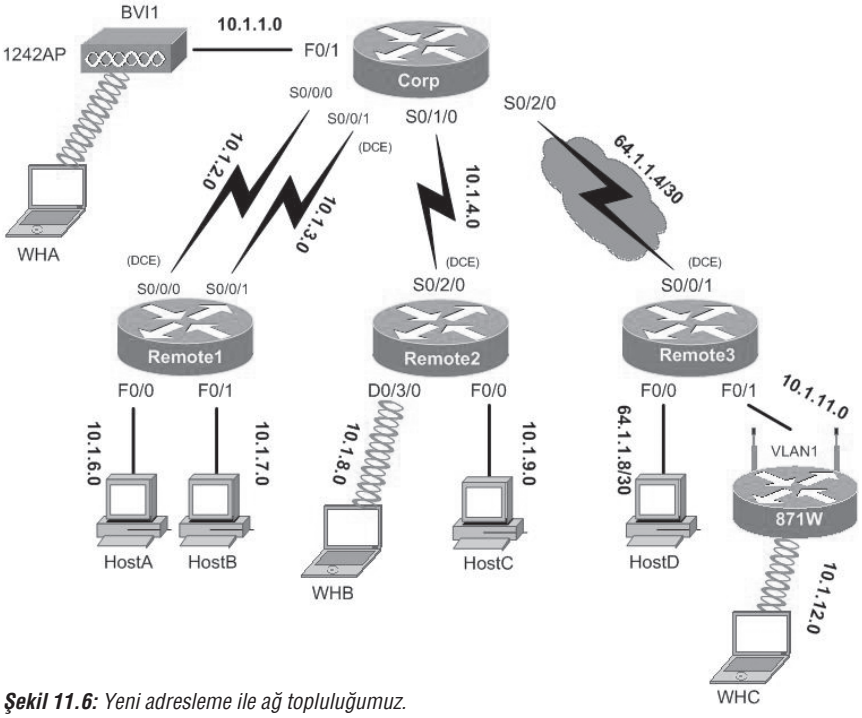
`ip nat inside source list 1 pool Todd overload` komutu, overload kullanarak Port Address Translation (PAT) kullanmak için dinamik bir havuz kullanır.

`ip nat inside` ve `ip nat outside` ifadelerini uygun interface'lere eklediğinizden emin olun.

# Ağ Topluluğumuzda NAT Yapılandırmak

Şimdi 64.1.1.4/30 ağını kullanarak R3 ile Corp router'ımızı ve 64.1.1.8/30 ağını kullanarak R3'deki LAN F0/0 linkini bağlayacağım. NAT çalıştıktan sonra, bu bölüm boyunca bahsettiğim doğrulama komutlarını kullanacağım.

Ağ topluluğumuz, Şekil 11.6'da görülmektedir ve bu kitap boyunca kullandığım, inside lokal adresleri, Tablo 11.3'de gösterilmektedir.




***Şekil 11.6:*** *Yeni adresleme ile ağ topluluğumuz.*

Şekil 11.6'nın, kullanmakta olduğum ağın aynısı olduğunu biliyorum, fakat bir değişiklik vardır. Corp router ile R3 router arasındaki bağlantı şimdi, global PAT adresleri kullanmaktadır. Diğer Corp bağlantıları özel adresleri kullandığında, onlar görüşemezler. (gerçek dünyada, ISP bunu bloklardı, değil mi? Öyleyse gelin bunu çalışır hale getirelim). NAT kullandığımızda (yani çeviriden önce), onları "inside locals" olarak çağırdığımızı ve ISP'mizin özel IP adres aralığını blokladığını hatırlayın. Ne yaparız? İlk olarak Corp router'ında NAT yapılandırmamız gerekir, öyleyse hadi çalışmaya başlayalım!

**Tablo 11.3:** IP Ağı İçin Network Adreslemesi

| Router | Network Adresi | Interface | Adres |
|---|---|---|---|
| Corp | | | |
| Corp | 10.1.1.0 | F0/1 | 10.1.1.1 |
| Corp | 10.1.2.0 | S0/0/0 | 10.1.2.1 |
| Corp | 10.1.3.0 | S0/0/1(DCE) | 10.1.3.1 |
| Corp | 10.1.4.0 | S0/1/0 | 10.1.4.1 |
| Corp | 64.1.1.4/30 | S0/2/0 | 64.1.1.5/30 |
| R1 | | | |
| R1 | 10.1.2.0 | S0/0/0 (DCE) | 10.1.2.2 |
| R1 | 10.1.3.0 | S0/0/1 | 10.1.3.2 |
| R1 | 10.1.6.0 | F0/0 | 10.1.6.1 |
| R1 | 10.1.7.0 | F0/1 | 10.1.7.1 |
| R2 | | | |
| R2 | 10.1.4.0 | S0/2/0 (DCE) | 10.1.4.2 |
| R2 | 10.1.8.0 | D0/3/0 | 10.1.8.1 |
| R2 | 10.1.9.0 | F0/0 | 10.1.9.1 |
| R3 | | | |

**Tablo 11.3:** IP Ağı İçin Network Adreslemesi *(devamı)*

| Router | Network Adresi | Interface | Adres |
|---|---|---|---|
| R3 | 64.1.1.4/30 | S0/0/1 (DCE) | 64.1.1.6/30 |
| R3 | 64.1.1.8/30 | F0/0 | 64.1.1.9/30 |
| R3 | 10.1.11.0 | F0/1 | 10.1.11.1 |
| 871W | | | |
| 871W | 10.1.11.0 | Vlan 1 | 10.1.11.2 |
| 871W | 10.1.12.0 | Dot11radio0 | 10.1.12.1 |
| 1242 AP | | | |
| 1242 AP | 10.1.1.0 | BVI 1 | 10.1.1.2 |

Şimdi Corp router'ına bağlı tüm ağdan, 64.1.1.5/30 yeni global adresini kullanan R3 router'a bağlı tüm ağlarla haberleşebilmesi gerektiğini hepimiz biliyoruz. Doğru mu? Kafanızı evet şeklinde sallıyorsunuz, evet doğru.

```
Corp#config t
Corp(config)#ip nat pool Todd 64.1.1.5 64.1.1.5 net
255.255.255.252
Corp(config)#access-list 1 permit 10.1.0.0 0.0.255.255
Corp(config)#ip nat inside source list 1 pool Todd overload
```

Interface yapılandırmasını eklemeden önce havuzun başlangıç ve bitiş adresi olarak Corp'un 64.1.1.5 dış interface adresini kullandığıma dikkat edin. PAT kullandığımda, onun oldukça iyi çalışmasından dolayı böyle yaptım.

Her neyse, tüm interface'lerde NAT'ı yapılandırmayı unutmamak çok önemlidir:

```
Corp(config)#int s0/2/0
Corp(config-if)#ip nat outside
Corp(config-if)#int f0/1
Corp(config-if)#ip nat inside
Corp(config-if)#int s0/0/0
Corp(config-if)#ip nat inside
Corp(config-if)#int s0/0/1
Corp(config-if)#ip nat inside
Corp(config-if)#int s0/1/0
Corp(config-if)#ip nat inside
Corp(config-if)#
```

Şimdi PAT yapılandırıldı ve tüm interface'lerimiz ayarlandı. Gelin HostC'den, HostD'yi ping'leyelim. İlk olarak host'tan, host'u ping'leyeceğim sonra telnet yapacağım:

```
Corp#sh ip nat trans
Pro Inside global      Inside local       Outside local
Outside global
icmp 64.1.1.5:271      10.1.9.2:271       64.1.1.10:271
64.1.1.10:271
```

```
tcp 64.1.1.5:11000     10.1.9.2:11000     64.1.1.10:23
64.1.1.10:23
Corp#
```

Şimdi Corp router'da, `debug ip nat`'ı açacağım ve sonra HostB'den HostD'ye telnet yapacağım. Gelin Corp router'ın çıktısına bir göz atalım:

```
Corp#debug ip nat
*May  9 22:57:47.679: NAT*: TCP s=11000->1024, d=23
*May  9 22:57:47.679: NAT*: s=10.1.6.2->64.1.1.5, d=64.1.1.10 [0]
*May  9 22:57:47.683: NAT*: TCP s=23, d=1024->11000
*May  9 22:57:47.683: NAT*: s=64.1.1.10, d=64.1.1.5->10.1.6.2 [0]
*May  9 22:57:47.699: NAT*: TCP s=11000->1024, d=23
*May  9 22:57:47.699: NAT*: s=10.1.6.2->64.1.1.5, d=64.1.1.10 [1]
*May  9 22:57:47.703: NAT*: TCP s=23, d=1024->11000
*May  9 22:57:47.703: NAT*: s=64.1.1.10, d=64.1.1.5->10.1.6.2 [1]
*May  9 22:57:47.707: NAT*: TCP s=11000->1024, d=23
*May  9 22:57:47.707: NAT*: s=10.1.6.2->64.1.1.5, d=64.1.1.10 [2]
*May  9 22:57:47.711: NAT*: TCP s=11000->1024, d=23
*May  9 22:57:47.711: NAT*: s=10.1.6.2->64.1.1.5, d=64.1.1.10 [3]
*May  9 22:57:47.719: NAT*: TCP s=23, d=1024->11000
*May  9 22:57:47.719: NAT*: s=64.1.1.10, d=64.1.1.5->10.1.6.2 [2]
*May  9 22:57:47.723: NAT*: TCP s=23, d=1024->11000
*May  9 22:57:47.723: NAT*: s=64.1.1.10, d=64.1.1.5->10.1.6.2 [3]
*May  9 22:57:47.723: NAT*: TCP s=11000->1024, d=23
*May  9 22:57:47.723: NAT*: s=10.1.6.2->64.1.1.5, d=64.1.1.10 [4]
*May  9 22:57:47.731: NAT*: TCP s=11000->1024, d=23
*May  9 22:57:47.731: NAT*: s=10.1.6.2->64.1.1.5, d=64.1.1.10 [5]
*May  9 22:57:47.735: NAT*: TCP s=23, d=1024->11000
*May  9 22:57:47.735: NAT*: s=64.1.1.10, d=64.1.1.5->10.1.6.2 [4]
*May  9 22:57:47.735: NAT*: TCP s=11000->1024, d=23
*May  9 22:57:47.735: NAT*: s=10.1.6.2->64.1.1.5, d=64.1.1.10 [6]
*May  9 22:57:47.747: NAT*: TCP s=11000->1024, d=23
*May  9 22:57:47.747: NAT*: s=10.1.6.2->64.1.1.5, d=64.1.1.10 [7]
*May  9 22:57:47.951: NAT*: TCP s=11000->1024, d=23
*May  9 22:57:47.951: NAT*: s=10.1.6.2->64.1.1.5, d=64.1.1.10 [8]
*May  9 22:57:48.103: NAT*: TCP s=23, d=1024->11000
*May  9 22:57:48.103: NAT*: s=64.1.1.10, d=64.1.1.5->10.1.6.2 [5]
Corp#
```

Bu biraz ilginç bir çıktıdır. İlk satırın, HostB'de kullanılan kaynak ve hedef port numaralarımızı gösterdiğini görebilirsiniz. İkinci satır, sonda listelenen outside local/global adres ile inside global adresimize ve sonra outside host'tan tekrar HostB'ye çevrilen inside kaynak adresimizi gösterir. Gelin bütün bunları `show ip nat translation` komutu ile doğrulayalım:

```
Corp#sh ip nat trans
Pro Inside global      Inside local       Outside local
Outside global
tcp 64.1.1.5:11000     10.1.9.2:11000     64.1.1.10:23
64.1.1.10:23
Corp#
```

Şimdi Corp router'ında `show ip nat statistics` komutunu kullanalım:

```
Corp#sh ip nat stat
Total active translations: 2 (0 static, 2 dynamic; 2 extended)
Outside interfaces:
  Serial0/2/0
Inside interfaces:
  FastEthernet0/1, Serial0/0/0, Serial0/0/1, Serial0/1/0
Hits: 269  Misses: 13
CEF Translated packets: 227, CEF Punted packets: 0
Expired translations: 27
Dynamic mappings:
— Inside Source
[Id: 1] access-list 1 pool Todd refcount 2
 pool Todd: netmask 255.255.255.252
        start 64.1.1.5 end 64.1.1.5
        type generic, total addresses 1, allocated 1 (100%),
misses 0
Queued Packets: 0
Corp#
```

Burada hem yapılandırmamızın özetini, iki aktif çeviriyi hem de kullanılmakta olan iç ve dış interface'leri görebiliriz. Havuz, çıktının aşağısına doğru listelenmiştir. Ve her şey güzel görünmektedir. Öyleyse, SDM kullanarak NAT yapılandırmasının zamanıdır.

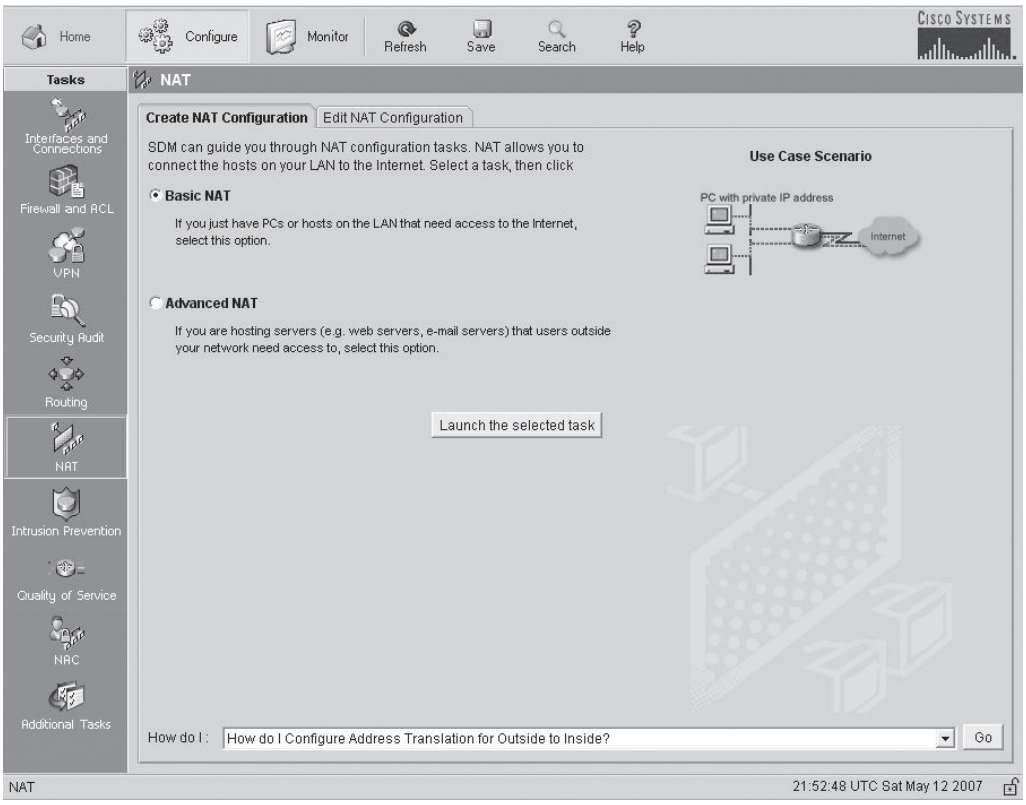
# SDM Kullanarak NAT Yapılandırmak

SDM kullanarak NAT'ı yapılandırmak, sizin dışınızdaki herkesin düşündüğünden çok daha kolaydır. Çünkü siz bölüm 10 boyunca zaten görmüştünüz. Tüm yapmanız gereken, Configure  NAT'a tıklamaktır. Ve bir NAT kuralı oluşturmak için size yardımcı olacak kullanışlı bir wizard gelir karşınıza. Bölüm 10'da firewall'umuzu oluştururken kullandığımıza çok benzer ve bölüm 10'daki gibi birden fazla wizard vardır. Siz yine basit ve gelişmiş arasında seçim yapacaksınız:

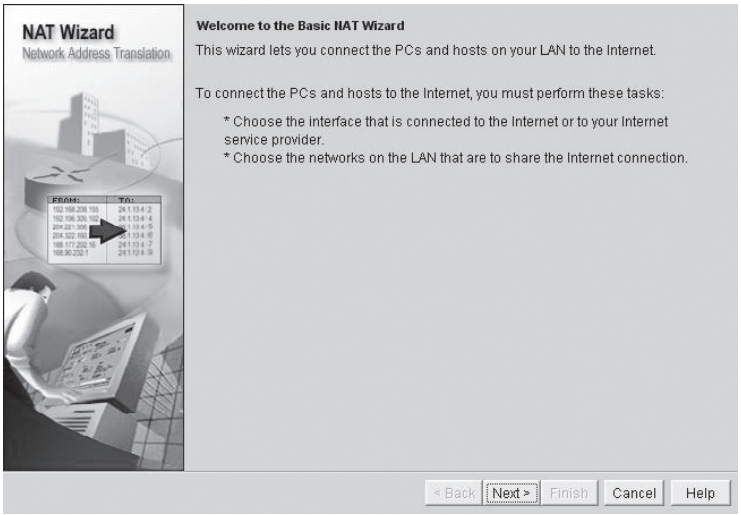
**Basit NAT:** Şayet internete erişmesi gereken güvenli ağınızda bazı basit PC'ler/Host'lar'a sahipseniz, bu wizard'ı kullanın. Bu wizard, basit bir NAT yapılandırma işleminde size kılavuzluk yapacaktır.

**Gelişmiş NAT:** Şayet bir DMZ'iniz veya dışarıdan kullanıcıların erişmesi gereken iç ağınızdaki sunucularınız varsa, Advanced NAT yapılandırmasını seçmek zorundasınız.

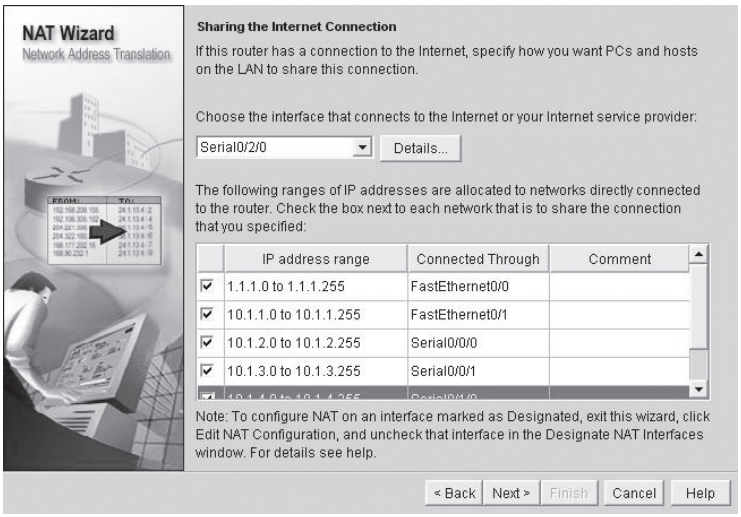
İlk ekran, Create NAT Configuration ekranıdır.



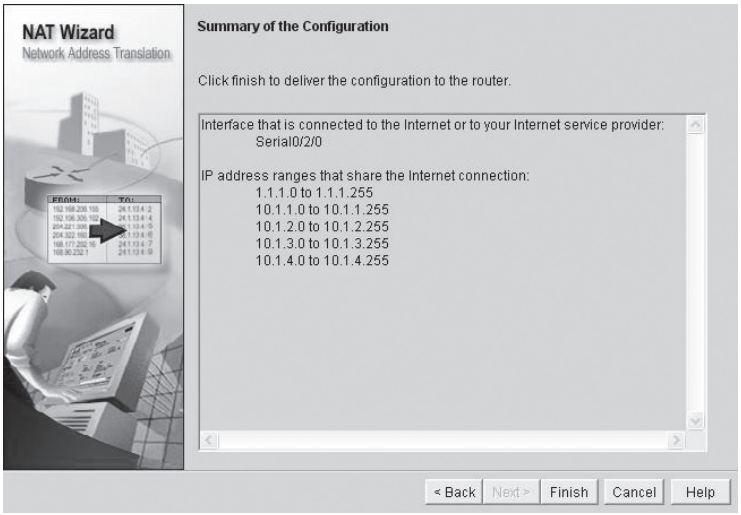

Buradan, basit olarak bağlanacağım ve bir basic NAT oluşturacağım. Bundan sonra Launch the Selected Task'ı seçtim ve bana Basic NAT Wizard'ın gittiğimi söyleyen sonraki ekrana geldim.




Tüm yapmam gereken, bütün iç ve dış adreslerimi seçebileceğim ekrana geçmek için Next butonuna tıklamaktır. Aşina geliyor mu? Güzel, bu dikkatinizi verdiğiniz anlamına geliyor.




İç ve dış interface'lerini seçtikten sonra Next butonuna tıklıyorum. NAT havuzu oluşturuldu ve bütün iç ve dış konfigürasyonlara interface'ler atandı.




Son olarak Finish butonuna tıkladım. Gelin router'ımda neler olduğuna bakalım. Yapılandırdığımız interface'lerimiz şöyledir:

```
!
interface FastEthernet0/0
 ip address 1.1.1.1 255.255.255.0
 ip nat inside
 ip virtual-reassembly
 duplex auto
 speed auto
!
interface FastEthernet0/1
 description Connection to 1242 AP
 ip address 10.1.1.1 255.255.255.0
 ip nat inside
 ip virtual-reassembly
 duplex auto
```

```
 speed auto
!
[output cut]
!
interface Serial0/2/0
 description Connection to R3$FW_OUTSIDE$
 ip address 64.1.1.5 255.255.255.252
 ip access-group 103 in
 ip verify unicast reverse-path
 ip nat outside
 ip inspect SDM_LOW out
 ip virtual-reassembly
 clock rate 2000000
!
[output cut]
Here is the ip nat inside source list it created:
ip nat inside source list 2 interface Serial0/2/0 overload
!
[output cut]
```

Ve son olarak, bir iç ağda gibi seçtiğim tüm interface'ler için oluşturulan access list buradadır:

```
access-list 2 remark SDM_ACL Category=2
access-list 2 permit 1.1.1.0 0.0.0.255
access-list 2 permit 10.1.4.0 0.0.0.255
access-list 2 permit 10.1.1.0 0.0.0.255
access-list 2 permit 10.1.2.0 0.0.0.255
access-list 2 permit 10.1.3.0 0.0.0.255
```

Bunu kitapta sık sık söylediğimi biliyorum fakat SDM'in, ACL, VPN ve NAT gibi gelişmiş yapılandırmaların oluşturulması için çok faydalı bir araç olduğunu bilmenizi istediğimden tekrarlamaktan hoşlanıyorum. Sizin için tamamlamayı düşündüğüm bir şeydir ve son iki modül bunu ispatlayacaktır.

# Özet

Bu gerçekten eğlenceli bir bölümdü. Hadi, kabul edin. Network Address Translation (NAT) ve onun statik, dinamik ve NAT Overload olarak da bilinen Port Address Translation (PAT) ile nasıl yapılandırıldığı hakkında çok şey öğrendiniz.

Ayrıca hem her NAT türünün bir ağda nasıl kullanıldığını hem de her birinin bir ağda nasıl yapılandırıldığını açıkladım. Artı sizin rahatınız ve kolaylığınız için kitap boyunca yapılandırdığım ağ topluluğunun aynısını kullandım ve ona kolayca NAT Overload'u (PAT) ekledim.

Bazı doğrulama ve hata tespiti komutlarını da inceledim ve sonra NAT'ın kolay ve hızlı bir şekilde yapılandırılması için SDM'in nasıl kullanıldığını göstererek bu bölümü tamamladım.

# Sınav Gereklilikleri

**NAT terimini anlamak:** Daha önce bahsetmediğimden dolayı bu size yeni gelebilir, fakat NAT bazı takma isimlere sahiptir. Endüstride, network network masquerading, IP-masquerading ve OCD ile kuşatılan ve her şeyi hecelemeye zorlanan bunlar için Native Address Translation diyebiliriz. Onu ne şekilde söylemek isterseniz, router veya bir firewall'dan geçtiklerinde, IP paketlerinin kaynak/hedef adreslerinin yeniden yazılması işlemine işaret ederler. Sadece meydana gelen işleme ve onu anlamaya odaklanın.

**NAT'ın üç yöntemini hatırlamak:** Üç yöntem, statik, dinamik ve Port Address Translation (PAT) olarak da bilinen overloading'dir.

**Statik NAT'ı anlamak:** Bu NAT tipi, lokal ve global adresler arasında bire bir eşleşmeyi kabul etmesi için tasarlanmıştır.

**Dinamik NAT'ı anlamak:**Bu versiyon size kayıtlı olmayan bir IP adresini, kayıtlı IP adreslerin havuzunun dışından bir kayıtlı adresle eşleştirme kabiliyeti verir.

**Overloding'i anlamak:** Overloading, farklı port'lar kullanarak, çok sayıda kayıtlı olmayan IP adresini, tek bir kayıtlı IP adresi ile eşleştiren Dinamik NAT'ın bir çeşididir. Bu neden bu kadar özeldir? Çünkü Port Address Translation (PAT) olarak da bilinmektedir.

# Yazılı Lab 11

Bu bölümde, aşağıdaki soruların cevaplarını yazın:

1. Hangi adres çeviri tipi, sadece bir adresin, binlerce host'un global olarak çevrilmesine izin vermesi için kullanılabilir?
2. Hangi komut, NAT çevirilerini, router'ınızda oluyor gibi göstermek için kullanılabilir?
3. Hangi komut size çeviri tablosunu gösterecektir?
4. Hangi komut, tüm NAT kayıtlarınızı çeviri tablosundan silecektir?
5. Her NAT eşleşmesi yaklaşık olarak ne kadar hafıza kullanır?
6. `ip nat translation max-entries` komutunu neden kullanırsınız?
7. Hangi komut hem NAT yapılandırmasının özetini göstermek ve hata tespiti yapmak için hem de aktif çeviri tiplerini saymak ve mevcut bir eşleşmeyi belirtmesi için kullanılabilir?
8. Hangi komutların, NAT adresleri çevirmeden önce router interface'lerinizde kullanılması gerekmektedir?
9. Aşağıdaki çıktıda, hangi tip NAT kullanılmaktadır?
10. `netmask` komutu yerine, _____ ifadesini de kullanabilirsiniz.

# Pratik Lab'lar

Bu lab için bazı basit router'lar kullanacağım, fakat aslında nerdeyse tüm Cisco router çalışacaktır.

Aşağıda bu modüldeki lab'ları bulabilirsiniz:

Lab 11.1: NAT için hazırlanmak

Lab 11.2: Dinamik NAT'ı yapılandırmak

Lab 11.3: PAT'ı yapılandırmak


***Şekil 11.7:*** *Modül 11 pratik lab ağı.*

Pratik lab'larımız için Şekil 11.7'de görülen ağı kullanacağım. Bazı router'lara bağlanıp, bu lab'ı gözden geçirmenizi öneririm. Bu lab'ta, 192.168.10.0 özel IP adresini, 171.16.10.0 bir genel IP adresine çevirmek için Lab_A router'ında NAT'ı yapılandıracaksınız.

Tablo 11.4, kullanacağımız komutları ve her komutun amacını göstermektedir.

**Tablo 11.4:** NAT/PAT Pratik Lab'ı İçin Komut Özetleri

| Komut | Amacı |
|---|---|
| ip nat inside source list acl pool name | Havuzdan, ACL ile eşleşen IP'leri çevirir |
| ip nat inside source static inside_addr outside_addr | Bir iç adresi statik olarak dış adrese eşleştirir. |
| ip nat pool name | Bir adres havuzu oluşturur. |
| ip nat inside | Bir interface'i, iç interface olması için ayarlamak. |
| ip nat outside | Bir interface'i, dış interface olması için ayarlamak. |
| show ip nat translations | Mevcut NAT çevirilerini gösterir. |

# Lab 11.1: NAT İçin Hazırlanmak

Bu lab'da, router'larınızı IP adresi ve RIP routing ile ayarlayacaksınız.

1. Router'larınızı Tablo 11.5'te listelenen IP adresleri ile yapılandırın.

**Tablo 11.5: Router IP Adres Şeması**

| Router | Interface | Ip Adresi |
|---|---|---|
| ISP | S0 | 171.16.10.1/24 |
| Lab_A | S0/2 | 171.16.10.2/24 |
| Lab_A | S0/0 | 192.168.20.1/24 |
| Lab_B | S0 | 192.168.20.2/24 |
| Lab_B | E0 | 192.168.30.1/24 |
| Lab_C | E0 | 192.168.30.2/24 |

Router'larınızı ayarladıktan sonra, router'dan router'a ping atabiliyor olmalısınız. Fakat bir sonraki adıma kadar, çalışan bir routing protokolümüz olmadığından, sadece bir router'dan diğerini doğrulayabilirsiniz, RIP kurulana kadar, ağı doğrulayamazsınız. Dilediğiniz routing protokolünü kullanabilirsiniz. Ben, hazırlanması ve çalışmasının basitliğinden dolayı RIP'i seçtim.

2. Lab_A'da, RIP routing'i ayarlayın, bir passive interface ayarlayın ve default network ayarlayın:

```
Lab_A#config t
Lab_A(config-router)#network 192.168.20.0
Lab_A(config-router)#network 171.16.0.0
Lab_A(config-router)#passive-interface s0/2
Lab_A(config-router)#exit
Lab_A(config)#ip default-network 171.16.10.1
```

   `Passive-interface` komutu, RIP güncellemelerinin ISP'ye gönderilmesini durdurur ve `ip-default network` komutu, internete nasıl çıkacaklarını bilmeleri için diğer router'lara default network'ü yayınlar.

3. Lab_B'de, RIP routing'i yapılandırın:

```
Lab_B#config t
Lab_B(config)#router rip
Lab_B(config-router)#network 192.168.30.0
Lab_B(config-router)#network 192.168.20.0
```

4. Lab_C'de, RIP routing'i yapılandırın, fakat routing tablomuzu ISP'ye göndermenin bir sebebi olmadığından, passive-interface komutunu kullanın.

```
Lab_C#config t
Lab_C(config)#router rip
Lab_C(config-router)#network 192.168.30.0
```

5. ISP router'da, şirket ağına bir default network yapılandırın:

```
ISP#config t
ISP(config)#ip route 0.0.0.0 0.0.0.0 s0
```

6. Şifre istenmeden router'a telnet yapabilecek şekilde ISP router'ını ayarlayın:

```
ISP#config t
ISP(config)#line vty 0 4
ISP(config-line)#no login
```

7. ISP router'ından Lab_C router'ına ve Lab_C router'ından ISP router'ına ping atabildiğinizi kontrol edin. Şayet yapamıyorsanız, ağınızda hata tespiti yapın.

# Lab 11.2: Dinamik NAT'ı Yapılandırmak

Bu lab'ta, Lab_A router'ında dinamik NAT yapılandıracaksınız.

1. Lab_A router'ında, GlobalNet adında bir adres havuzu oluşturun. Havuzun, 171.16.10.50 ile 171.16.10.55 aralığını içermesi gerekmektedir.

```
Lab_A(config)#ip nat pool GlobalNet 171.16.10.50 171.16.10.55
net 255.255.255.0
```

2. Accee-list 1'i oluşturun. Bu list, 192.168.20.0 ve 192.168.30.0 ağlarından trafiğin çevrilmesine için verir.

```
Lab_A(config)#access-list 1 permit 192.168.20.0 0.0.0.255
Lab_A(config)#access-list 1 permit 192.168.30.0 0.0.0.255
```

3. Access list'i oluşturulan havuza eşleştirin.

```
Lab_A(config)#ip nat inside source list 1 pool GlobalNet
```

4. Serial0/0'ı bir iç NAT interface'i olarak ayarlayın.

```
Lab_A(config)#int s0/0
Lab_A(config-if)#ip nat inside
```

5. Serial0/2'ı bir dış NAT interface'i olarak ayarlayın.

```
Lab_A(config-if)#int s0/2
Lab_A(config-if)#ip nat outside
```

6. Lab_C'ye bağlanın. Lab_C router'ından, ISP router'ına telnet yapın.

```
Lab_C#telnet 171.16.10.1
```

7. Lab_B'ye bağlanın. Lab_B router'ından, ISP router'ına telnet yapın.

```
Lab_B#telnet 171.16.10.1
```

8. ISP router'ında `show users` komutunu çalıştırın.(bu, VTY line'larına kimin eriştiğini gösterir)

```
ISP#show users
```

A. Kaynak adresinizi ne gösterir? _____

B. Gerçek kaynak adresiniz nedir? _____

*Bire-bir çeviri olduğuna dikkat edin. Yani, internete çıkmak isteyen her kullanıcı için bir gerçek IP adresiniz olması gerekir.*

NOT

`Show users` çıktısı şöyle görünmelidir:

```
ISP>sh users
   Line       User       Host(s)         Idle        Location
   0 con 0                 idle          00:03:32
   2 vty 0                 idle          00:01:33  171.16.10.50
*  3 vty 1                 idle          00:00:09  171.16.10.51
  Interface   User       Mode            Idle Peer Address
ISP>
```

9. ISP'deki oturumu açık bırakın ve Lab_A'ya bağlanın. (Ctrl+Shift+6'yı kullanın, bırakın ve sonra X'e basın.)

10. Lab_A router'ınıza bağlanın ve `show ip nat translation` komutunu girerek, mevcut çevirilerinize bakın. Şöyle bir şey görmelisiniz:

```
Lab_A#sh ip nat translations
Pro Inside global      Inside local       Outside local
Outside global
--- 171.16.10.50       192.168.30.2       ---                ---
--- 171.16.10.51       192.168.20.2       ---                ---
Lab_A#
```

11. Şayet Lab_A router'ında `debug ip nat`'ı açarsanız ve sonra router'ı ping'lerseniz, NAT işleminin gerçekleştiğini göreceksiniz. Şöyle görünecektir:

```
00:32:47: NAT*: s=192.168.30.2->171.16.10.50, d=171.16.10.1 [5]
00:32:47: NAT*: s=171.16.10.1, d=171.16.10.50->192.168.30.2
```

# Lab 11.3: PAT'ı Yapılandırmak

Bu lab'ta, Lab_A router'ında Port Address Translation'ı (PAT) yapılandıracaksınız. Bire bir çeviri istemediğimiz için ağdaki tüm kullanıcılar için sadece bir IP adresi kullanan PAT'ı kullanacağız.

1. Lab_A router'ında, çeviri tablosunu silin ve dinamik NAT havuzunu kaldırın.

```
Lab_A#clear ip nat translation *
Lab_A#config t
Lab_A(config)#no ip nat pool GlobalNet 171.16.10.50
171.16.10.55 netmask 255.255.255.0
Lab_A(config)#no ip nat inside source list 1 pool GlobalNet
```

2. Lab_A router'ında, tek bir adresle Lammle isminde bir havuz oluşturun. Havuz, sadece 171.16.10.100 IP adresini içermelidir. Aşağıdaki komutu girin:

```
Lab_A#config t
Lab_A(config)#ip nat pool Lammle 171.16.10.l00 171.16.10.100
net 255.255.255.0
```

3. Access-list 2'yi oluşturun. Bu, 192.168.20.0 ve 192.168.30.0 ağlarının çevrilmesine izin verecektir.

```
Lab_A(config)#access-list 2 permit 192.168.20.0 0.0.0.255
Lab_A(config)#access-list 2 permit 192.168.30.0 0.0.0.255
```

4. Access-list 2'yi yeni havuz ile eşleştirin. Overload komutunu kullanarak PAT olmasına izin verin.

```
Lab_A(config)#ip nat inside source list 2 pool Lammle overload
```

5. Lab_C'ye bağlanın ve ISP router'ına telnet yapın. Ayrıca Lab_C'ye bağlanın ve ISP router'ına telnet yapın.
6. ISP router'ında `show users` komutunu çalıştırın. Çıktı şu şekilde olmalıdır:

```
ISP>sh users
    Line       User       Host(s)           Idle       Location
*  0 con 0                idle              00:00:00
   2 vty 0                idle              00:00:39   171.16.10.51
   4 vty 2                idle              00:00:37   171.16.10.50

  Interface  User      Mode                Idle Peer Address

ISP>
```

7. Lab_A router'ından, `show ip nat translations` komutunu kullanın.

```
Lab_A#sh ip nat translations
Pro Inside global  Inside local  Outside local Outside global
tcp 171.16.10.100:11001 192.168.20.2:11001 171.16.10.1:23
171.16.10.1:23
tcp 171.16.10.100:11002 192.168.30.2:11002 171.16.10.1:23
171.16.10.1:23
tcp 171.16.10.100:1024  192.168.20.2:11002 171.16.10.1:23
171.16.10.1:23
```

8. Lab_A router'ında debug IP NAT komutunun açık olduğuna emin olun. Şayet Lab_C router'ından, ISP router'ına ping atarsanız, çıktı şöyle olacaktır:

```
01:12:36: NAT: s=192.168.30.2->171.16.10.100, d=171.16.10.1 [35]
01:12:36: NAT*: s=171.16.10.1, d=171.16.10.100->192.168.30.2 [35]
01:12:36: NAT*: s=192.168.30.2->171.16.10.100, d=171.16.10.1 [36]
01:12:36: NAT*: s=171.16.10.1, d=171.16.10.100->192.168.30.2 [36]
01:12:36: NAT*: s=192.168.30.2->171.16.10.100, d=171.16.10.1 [37]
01:12:36: NAT*: s=171.16.10.1, d=171.16.10.100->192.168.30.2 [37]
01:12:36: NAT*: s=192.168.30.2->171.16.10.100, d=171.16.10.1 [38]
01:12:36: NAT*: s=171.16.10.1, d=171.16.10.100->192.168.30.2 [38]
01:12:37: NAT*: s=192.168.30.2->171.16.10.100, d=171.16.10.1 [39]
01:12:37: NAT*: s=171.16.10.1, d=171.16.10.100->192.168.30.2 [39]
```

# Gözden Geçirme Soruları

NOT *Aşağıdaki sorular, bu bölümün materyallerini anladığınızı test etmek için tasarlanmıştır. İlave sorulara nasıl sahip olacağınızla ilgili daha fazla bilgi için bu kitabın Giriş bölümüne bakın.*

1. Aşağıdakilerden hangileri NAT kullanmanın dezavantajlarındandır? (Üç şık seçin.)
   A. Çeviri, anahtarlama yol gecikmelerine sebep olur.
   B. Yasal olarak register edilen adresleri korur.
   C. Uçtan-uca IP izlenebilirliğinin kaybına sebep olur.
   D. İnternete bağlanıldığında esnekliği artırır.
   E. Belirli uygulamalar, NAT'ın etkin olmasıyla çalışamayacaktır.
   F. Adres çakışmalarını azaltır.
2. Aşağıdakilerden hangileri NAT kullanmanın avantajlarındandır? (Üç şık seçin.)
   A. Çeviri, anahtarlama yol gecikmelerine sebep olur.
   B. Yasal olarak register edilen adresleri korur.
   C. Uçtan-uca IP izlenebilirliğinin kaybına sebep olur.
   D. İnternete bağlanıldığında esnekliği artırır.
   E. Belirli uygulamalar, NAT'ın etkin olmasıyla çalışamayacaktır.
   F. Adres çakışmalarını azaltır.
3. Hangi komut router'ınızdaki gerçek-zamanlı çevirileri gösterecektir?
   A. show ip nat translations
   B. show ip nat statistics
   C. debug ip nat
   D. clear ip nat translations *
4. Hangi komut router'ınızdaki tüm aktif çevirileri gösterecektir?
   A. show ip nat translations
   B. show ip nat statistics
   C. debug ip nat
   D. clear ip nat translations *
5. Hangi komut router'ınızdaki tüm aktif çevirileri silecektir?
   A. show ip nat translations
   B. show ip nat statistics
   C. debug ip nat
   D. clear ip nat translations *
6. Hangi komut NAT yapılandırmasının özetini gösterecektir?
   A. show ip nat translations
   B. show ip nat statistics
   C. debug ip nat
   D. clear ip nat translations *

7. Hangi komut, 30 global adres sağlayacak, Todd isminde dinamik bir havuz oluşturacaktır?
   A. ip nat pool Todd 171.16.10.65 171.16.10.94 net 255.255.255.240
   B. ip nat pool Todd 171.16.10.65 171.16.10.94 net 255.255.255.224
   C. ip nat pool todd 171.16.10.65 171.16.10.94 net 255.255.255.224
   D. ip nat pool Todd 171.16.10.1 171.16.10.254 net 255.255.255.0
8. Aşağıdakilerden hangileri, NAT'ın üç yöntemi olarak kabul edilmektedir?
   A. Statik
   B. IP NAT pool
   C. Dinamik
   D. NAT double-translation
   E. Overload
9. Bir global adres havuzu oluşturulduğunda aşağıdakilerden hangisi `netmask` yerine kullanılabilir?
   A. / (slash notation)
   B. prefix-length
   C. no mask
   D. block-size
10. Router'ınız çeviri yapmıyorsa, aşağıdakilerden hangisi hata tespiti için güzel bir başlangıç noktası olur?
   A. Reboot edin.
   B. Cisco'yu çağırın.
   C. İnterface'lerinizde doğru ayar olup olmadığını kontrol edin.
   D. Debug all komutunu çalıştırın.
11. Aşağıdakilerden hangisi NAT kullanmak için iyi bir sebep olabilir? (Üç şık seçin.)
   A. İnternete bağlanmanız gerekmekte ve host'larınızın, global olarak benzersiz IP adresleri yoktur.
   B. Ağınızı yeniden numaralandırmanızı gerektiren, yeni bir ISP'ye geçtiniz.
   C. Host'larınızın internete bağlanmasını istemiyorsunuz.
   D. Birleştirmek için çoğaltılmış adreslerle iki intranet'e ihtiyacınız vardır.
12. Aşağıdakilerden hangisi çeviriden sonraki bir adres olarak kabul edilir?
   A. Inside local
   B. Outside local
   C. Inside global
   D. Outside global
13. Aşağıdakilerden hangisi çeviriden önceki bir adres olarak kabul edilir?
   A. Inside local
   B. Outside local
   C. Inside global
   D. Outside global

14. Aşağıdakilerden hangisi çeviriden önceki hedef hosts olarak kabul edilir?
    A. Inside local
    B. Outside local
    C. Inside global
    D. Outside global
15. Aşağıdakilerden hangisi çeviriden sonraki dış hedef host'u olarak kabul edilir?
    A. Inside local
    B. Outside local
    C. Inside global
    D. Outside global
16. Özel bir ağda hangi komutu bir interface'e yerleştirirsiniz?
    A. ip nat inside
    B. ip nat outside
    C. ip outside global
    D. ip inside local
17. Hangi komutu, internete bağlı bir interface'e yerleştirirsiniz?
    A. ip nat inside
    B. ip nat outside
    C. ip outside global
    D. ip inside local
18. Port Adress Translation aynı zamanda ne olarak belirtilir?
    A. NAT Fast
    B. NAT Statik
    C. NAT Overload
    D. Overloading Statik

# Gözden Geçirme Sorularının Cevapları

1. A,C,E NAT mükemmel değildir ve bazı ağlarda bazı sorunlara neden olabilir. Fakat birçok ağ için iyidir. NAT, bazı gecikmelere ve hata tespiti sorunlarına neden olabilir ve bazı uygulamalar onunla çalışmayacaktır.
2. B,D,F NAT mükemmel değildir, fakat bazı avantajları vardır. Milyonlarca host'un internete gerçek IP olmaksızın çıkmasına izin vererek, global adresleri korur. Bu, şirket ağlarımızda esneklik sağlar. NAT ayrıca, ağlar çakışmadan, aynı subneti, aynı ağda birden fazla kullanmanıza izin verebilir.
3. C `debug ip nat` komutu, router'ınızda olan gerçek-zamanlı çevirileri gösterecektir.
4. A `show ip nat translations` komutu, tüm aktif NAT kayıtlarını içeren, çeviri tablosunu gösterecektir.
5. D `clear ip nat translations *` komutu, çeviri tablonuzdaki tüm nat kayıtlarını silecektir.
6. B `show ip nat statistics` komutu, hem NAT yapılandırmasının özetini gösterir hem de aktif çeviri tiplerini sayar, mevcut bir eşleşmeyi, kayıpları (bir eşleşme oluşturma girişimine sebep olan) ve sona eren çevirileri belirtir.
7. B `ip nat pool name` komutu, host'ların internete çıkmak için kullanabilecekleri havuzu oluşturur. B şıkkını doğru yapan, 30 host içeren, 171.16.10.65'ten 171.16.10.94'e kadar adres aralığıdır. Fakat mask'ın, 30 host'lada eşleşmesi gerekir ve mask 255.255.255.224'tür. C şıkkı yanlıştır, çünkü havuz ismindeki T küçük harflidir. Havuz isimleri küçük/büyük harf duyarlıdır.
8. A,C,E NAT'ı bir Cisco router'da üç yöntemle yapılandırabilirsiniz: statik, dinamik ve NAT overload (PAT).
9. B netmask komutunun yerine, prefix-length length komutunu kullanabilirsiniz.
10. C NAT'a çeviri servisleri sağlamak için router interface'lerinizde yapılandırılmış `ip nat inside` ve `ip nat outside`'a sahip olmalısınız.
11. A,B,D İnternete bağlanmak istiyorsanız ve host'larınızın global adreslere sahip olmasını istemiyorsanız, NAT'ı kullanmak istersiniz. En yaygın kullanım budur. Fakat B ve D şıkları da doğrudur.
12. C Çeviriden sonra, özel ağdaki host, inside global host olarak kabul edilir.
13. A Çeviriden önce, özel ağdaki host, inside local host olarak kabul edilir.
14. B Çeviriden önce, global ağdaki host, outside local host olarak kabul edilir.
15. D Çeviriden sonra, global ağdaki host, outside global host olarak kabul edilir.
16. A access-list'lerdeki gibi, NAT bir çeviri sağlamadan önce, interface'lerinizi yapılandırmalısınız. İç ağlarınızda, `ip nat inside`, dış ağlarınızda, `ip nat outside` komutunu kullanırsınız.
17. B access-list'lerdeki gibi, NAT bir çeviri sağlamadan önce, interface'lerinizi yapılandırmalısınız. İç ağlarınızda, IP NAT inside, dış ağlarınızda, IP NAT outside komutunu kullanırsınız.
18. C Port address translation için diğer terim, port adres çevirini mümkün kılan bir komut olmasından dolayı, NAT Overload'dur.

# Yazılı Lab 11'in Cevapları

1. NAT Overload olarak da bilinen, Port Address Translation (PAT)
2. debug ip nat
3. show ip nat translations
4. clear ip nat translations *
5. 160 bytes hafıza
6. performans veya politika nedeniyle girişlerin sınırlandırılması gereken durumlarda.
7. show ip nat statistics
8. ip nat inside ve ip nat outside komutları
9. Dinamik NAT
10. prefix-length

# 12 Cisco Wireless Teknolojileri

# 12 Cisco Wireless Teknolojileri

- Wireless Teknolojisine Giriş
- Cisco Unified Wireless Solution
- Kablosuz Ağ topluluğumuzu Yapılandırmak
- Özet
- Sınav Gereklilikleri
- Yazılı Lab 12
- Gözden Geçirme Soruları
- Gözden Geçirme Sorularının Cevapları
- Yazılı Lab 12’nin cevapları

# Cisco Wireless Teknolojileri

Günümüzde kullanılan en yaygın temel wireless LAN'ları (WLAN) anlamak istiyorsanız, hub'lardaki 10BaseT Ethernet'i düşünün. Yani, WLAN'larımız tipik olarak, herkesin, aynı bant genişliğini paylaştığı ve aynı zamanda sadece tek bir kullanıcının haberleştiği half-duplex iletişim kurarlar. Bu çok kötü değil ama yeterince de iyi değildir. Birçok insanın, günümüzde kablosuz ağlara güvenmesinden dolayı, hızla artan ihtiyaçlarımızı yakalayabilmek için daha hızlı gelişmeleri çok önemlidir. İyi haber, bu gerçekten olmaktadır. Cisco, her tür kablosuz bağlantı ile çalışan Cisco Unified Wireless Solution olarak adlandırılan bir çözüm sundu. Ve güvenli çalışıyor!

Bu bölümdeki amacım kablosuz teknolojileri genel olarak tanıtmak değil, Cisco'nun kablosuz teknolojisini tanıtmaktır. Çünkü tahmin edebileceğiniz gibi küçük farklılıklar vardır. Evet, temel kablosuz LAN teknolojileri ve komisyonlarını işleyeceğim, fakat buradaki ana amaç wireless'ı, Cisco'nun gözüyle anlamanız ve Cisco'nun sunduğu çözümleri iyice kavramanızdır.

Cisco'nun Unified Wireless Solution, mobility ve mesh'i kapsar, bu nedenle bu konular üzerinde yoğunlaşacağım. Ayrıca wireless güvenliği ile ilgili tüm önemli konular hakkında da ince bilgiler vereceğim.

Bu bölüm, bölüm 14'ün, yani "Wide Area Network"ün devamı olabilirdi, çünkü kablosuz cep telefonu ağları ve diğer hızlı kablosuz teknolojiler ortaya çıkmaktadır. Fakat bu bölümdeki konuları onları ayrı bir bölümde toplayarak ne kadar önemli olduklarını fark etmenizi istedim. Çünkü wireless'in, wide area network'e (WAN) bir alternatif olarak önemini anlamanızı istedim. Ve merak ediyorsanız evet, Cisco Unified Wireless Solution, Wireless Metropolitan Area Network'leri (WMAN) içermektedir.

*Bu bölüm ile ilgili son güncellemeler için www.lammle.com ve/veya www.sybex.com adreslerine bakınız.*

NOT

## Wireless Teknolojisine Giriş

Tipik 802.11 düzenlemesi kullanan bir sinyali aktarmak, basit bir Ethernet hub'ına oldukça benzer şekilde çalışır: Her ikisi de, iki taraflı haberleşme modelidir. Göndermek ve almak için aynı frekansı kullanırlar ve bu daha önceki bölümlerde açıklandığı gibi half-duplex olarak belirtilir. Wireless LAN'lar (WLAN) radyo dalgaları oluşturan bir antenden havaya yayılan radyo frekanslarını (RF) kullanır. Bu dalgalar, sinyal gücü azalacak şekilde su, duvar ve metal yüzeyler tarafından emilebilir, kırılabilir veya yansıtılabilir. Bu çevresel faktörlerle çevrili doğal hassasiyetten dolayı, wireless'ın kablolu ağların sağlayabildiği sağlamlılığın aynısını asla sunamayacağı çok açıktır. Fakat bu hala, wireless kullanmayacağız anlamına gelmemektedir. Bana inanın, kesinlikle kullanacağız!

Aktarım gücümüzü artırabilir ve büyük aktarım uzaklığına ulaşabiliriz, fakat böyle yapmak bazı bozulmalara neden olabilir, bu nedenle onun çok dikkatli ayarlanması gerekir. Yüksek frekanslar kullanarak, yüksek veri hızına ulaşabiliriz. Fakat bu maalesef azalmış aktarım mesafesinden fedakarlıkla olur. Şayet daha düşük frekans kullanırsak, düşük veri hızında daha uzağa aktarım yapabiliriz. Şunun sizin için netleşmesi gerekir ki, uygulayacağınız tüm farklı WLAN tiplerini anlamak, sizin kullandığınız durumun özel gerekliliklerini en iyi karşılayan LAN çözümünü oluşturmak için zorunludur.

Ayrıca şu da önemlidir 802.11 düzenlemeleri o kadar gelişti ki, kullanıcının herhangi bir lisans ve işletim ücreti olmaksızın kurulum yapıp, çalıştırması serbestliğini sağlamak için birçok ülkede lisanslama gerekmemektedir. Yani, bir üretici, ürünlerini geliştirebilir ve onları yerel bir bilgisayar mağazasında veya nerede isterse satabilir.

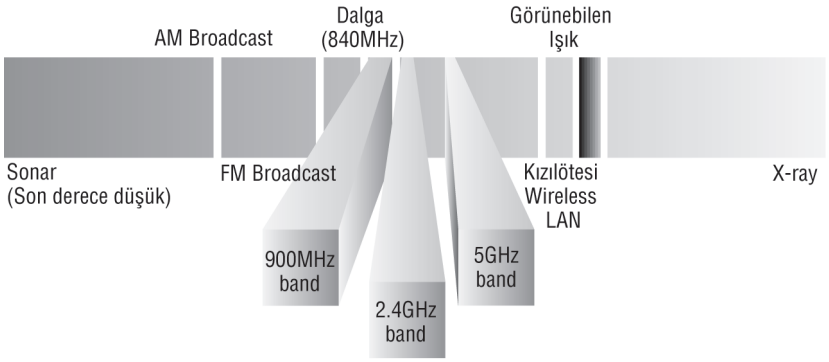
Çeşitli kurumlar wireless cihazların, frekansların, standartların ve frekans spektrumlarının kullanımını yönetmeye yardımcı olmak için çok uzun zamandır uğraşmaktadır. Tablo 12.1, dünya genelinde wireless standartlarının oluşturulması, sağlanması ve hatta uygulanmasına yardımcı olan mevcut kurumları göstermektedir.

**Tablo 12.1:** Wireless Kurumları ve Standartları

| Kurum | Amacı | Web Sitesi |
|---|---|---|
| Institute of Electrical and Electronics Engineers (IEEE) | Kullanılmaya hazır standartlar oluşturmak ve sağlamak. | www.ieee.org |
| Federal Communications Commission (FCC) | U.S'de wireless cihazların kullanımı düzenlemek. | www.fcc.gov |
| European Telecommunications Standards Institute (ETSi) | Avrupa'daki yaygın standartları geliştirmek için yetkili. | www.etsi.org |
| Wi-Fi Alliance | WLAN birlikte işlerliliğini geliştirmek ve test etmek. | www.wi-fi.com |
| WLAN Association (WLANA) | WLAN'lar hakkında tüketici bilincini geliştirir ve öğretir. | www.wlana.org |

WLAN'ların radyo frekansları üzerinden aktarılmasından dolayı, AM/FM radyo gibi yapıları yönetmek için kullanılan bazı kanun maddeleriyle düzenlenmektedirler. Kablosuz LAN cihazlarının kullanımını, Federal Communications Commission (FCC) düzenler, buradan Institute of Electrical and Electronics Engineers (IEEE) alır ve FCC'nin kamusal kullanım için serbest bıraktığı frekansların temelinde standartlar oluşturur.

FCC, kamusal kullanım için üç lisanssız bant yayınlamıştır: 900MHz, 2.4GHz ve 5.7GHz. 900MHz ve 2.4GHz bantları, endüstriyel, bilimsel ve tıbbi bantlar olarak belirtilir. 5-GHz bant, Unlicensed National Information Infrastructure (UNII) bandı olarak da bilinir. Şekil 12.1 lisanssız bantların, RF spektrumunda nerede bulunduklarını göstermektedir.




***Şekil 12.1:*** *Lisanssız frekanslar.*

Bundan dolayı, Şekil 12.1'de gösterilen üç umumi bandın dışında bir aralıkta wireless yayın yapmayı seçerseniz, bunun için FCC'den özel bir lisans almanız gerekmektedir. FCC, üç frekans aralığını kamusal kullanıma açınca, birçok üretici, günümüzde en yaygın kullanılan kablosuz ağ olan 802.11b/g ile marketlere dağıtılan çok sayıda ürünü sağlamaya başlayabildi.

Wi-Fi Alliance, farklı üreticiler tarafından önerilen 802.11 ürünleri arasında birlikte çalışabilirlik için belgelendirme sağladı. Access point'lerinizin tamamını aynı üreticiden almanızın tamamıyla daha kolay olduğuyla ilgili kişisel tecrübeme rağmen, bu belgelendirme çok çeşitli ürünleri alan kullanıcılar için bir çeşit güven bölgesi sağladı.

Mevcut U.S. kablosuz LAN pazarında, Institute of Electrical and Electronics Engineers (IEEE) tarafından geliştirilen ve sürdürülen bazı onaylı işlevsel standartlar ve taslaklar vardır. Gelin bu standartlara bir göz atalım ve sonra en yaygın kullanılan standartların nasıl çalıştıklarını konuşalım.

## 802.11 Standartları

"Ağlar arası İletişim" başlıklı bölüm 1'den yola çıkarsak, kablosuz ağ kurulumu, kendi 802 standart grubuna sahiptir (ethernet'in komisyonunun 802.3 olduğunu hatırlayın). Wireless, 802.11 ile başlar ve sonra 802.16 ve 802.20 gibi, faal ve geleceği parlak çeşitli standart grupları vardır. Ve hiç tereddüt yoktur ki, cep telefonu ağları, wireless geleceğimizde çok önemli faktörlerden olacaktır. Fakat şimdi, 802.11 standart komisyon ve alt komisyonlarına odaklanıyoruz.

IEEE 802.11; 1 ve 2Mbps'da ilk orijinal ve standartlaşmış WLAN'dır. 2.4GHz radyo frekansında çalışır ve 802.11b'nin ortaya çıktığı 1999'a kadar, çok fazla ürün görmememize rağmen, 1997'de onaylanmıştır. Tablo 12.2'de listelenen tüm komiteler, bağımsız dökümanlar olan 802.11F ve 802.11T dışındaki orijinal 802.11 standart revizyonlarıdır.

**Tablo 12.2:** 802.11 Komiteler ve Alt Komiteler

| Komite | Amacı |
|---|---|
| IEEE 802.11a | 54Mbps, 5GHz standart |
| IEEE 802.11b | 5.5 ve11Mbps'ı desteklemesi için 802.11'e ilaveler |
| IEEE 802.11c | IEEE 802.1D standartlarını içeren, Bridge çalışma yöntemleri |
| IEEE 802.11d | Uluslar arası dolaşım genişlemeleri |
| IEEE 802.11e | Servis kalitesi |
| IEEE 802.11F | Inter-Access Point Protocol |
| IEEE 802.11g | 54Mbps, 2.4GHz standardı (802.11b ile geriye dönük uyumlu) |
| IEEE 802.11h | 5Ghz'de Dynamic Frequency Selection (DFS) ve Transmit Power Control (TPC). |
| IEEE 802.11i | Gelişmiş güvenlik |
| IEEE 802.11j | Japon ve U.S. genel güvenliği için ilaveler |
| IEEE 802.11k | Radyo kaynak ölçüm geliştirme |
| IEEE 802.11m | Standardın bakımı; ufak tefek şeyler |
| IEEE 802.11n | Daha yüksek throughput, MIMO (multiple input, multiple output antennas) kullanımını geliştirir |
| IEEE 802.11p | Wireless Access for the Vehicular Environment (WAVE) |
| IEEE 802.11r | Hızlı dolaşım |
| IEEE 802.11s | Extended Service Set (ESS) Mesh Networking |
| IEEE 802.11T | Wireless Performance Prediction (WPP) |
| IEEE 802.11u | 802-olmayan ağlarla iletişim (örneğin, cep telefonu) |
| IEEE 802.11v | Wireless network yönetimi |
| IEEE 802.11w | Korumalı yönetim frame'leri |
| IEEE 802.11y | U.S'de 3650–3700 çalışması |

Şimdi popüler 802.11 WLAN'ların en önemlilerini tartışalım.

## 2.4GHz (802.11b)

Menüde ilk sırada 802.11b standardı var. En yaygın kullanılan wireless standardıydı. Maksimum 11Mbps veri hızı kullanan ve 2.4GHz'de çalışan lisanssız radyo bandıdır. 802.11b standardı, çoğu uygulama için çalışan 11Mbps veri hızını oldukça iyi bulan üretici ve tüketiciler tarafından yaygın şekilde benimsenmiştir. Fakat şimdi 802.11b'nin ağabeyi vardır (802.11g). Artık hiç kimse gidip 802.11b kartı veya access point satın almaz. Aynı paraya, 10/100 Ethernet kart alabilecekken, niye 10Mbps Ethernet kart alasınız?

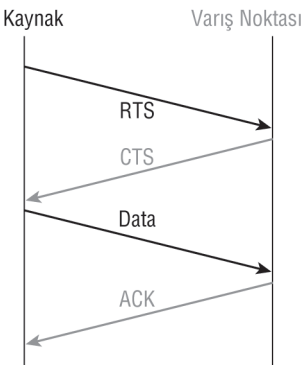
Tüm Cisco 802.11 WLAN ürünleriyle ilgili ilginç olan, taşınırken, veri hızı değişim kabiliyetine sahip olmalarıdır. Bu, 11Mbps'da çalışan kişiye, 5.5Mbps, 2Mbps ve son olarak access point'ten en uzak mesafeyle hala haberleşen 1Mbps'a kaymasına izin verir. Dahası, bu hız değişimi, bağlantı kaybı ve kullanıcıyla etkileşim olmaksızın gerçekleşir. Hız değişimi ayrıca, aktarım tabanlı olmaktadır. Bu önemlidir, çünkü access point'in, her istemcinin lokasyonuna bağlı olarak farklı hızlarda birçok kullanıcıyı destekleyebileceği anlamına gelir.

802.11b ile ilgili problem, Data Link katmanını nasıl ele alacağıyla ilgilidir. RF spektrumunda bu problemi çözmek için CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance) denilen Ethernet collision algılama yöntemi geliştirildi. Bunu Şekil 12.2'de inceleyin.

Host'ların access point'lerle (AP) iletişim kurması gerektiğinden dolayı CSMA/CA aynı zamanda Request To Send, Clear To Send (RTS/CTS) olarak da adlandırılır. Her paket gönderildiğinde, alındığına dair bir RTS/CTS onayının alınması gerekir. Ağır işleyen bu proses yüzünden, bu işlemin gerçekten de işe yaradığına inanmak oldukça güçtür.

## 2.4GHz (802.11g)

802.11g standardı Haziran 2003'de imzalandı. 802.11b ile uyumludur. 802.11g standardı aynen 802.11a gibi maksimum 54Mbps veri hızı sağlar ancak 802.11b gibi 2.4GHz aralığında çalışır.




***Şekil 12.2:*** *802.11b CSMA/CA.*

802.11b/g'nin aynı 2.4GHz lisanssız bantta çalışmasından dolayı, 802.11g'ye geçiş, 802.11b kablosuz altyapı ile şirketler için maddi olarak karşılanabilir bir seçimdir. 802.11b ürünlerinin, 802.11g'ye yazılım yükseltmesi yapılamayacağını unutmayın. Bu sınırlamanın sebebi, daha yüksek veri hızında taşıması için 802.11g radyolarının, farklı bir chipset kullanmalarıdır.

Fakat hala, 802.11g ürünleri, Ethernet ve Fast Ethernet gibi çoğu ağdaki 802.11b ürünleri ile beraber çalışmaktadır. Yine de, örneğin Ethernet'in tamamıyla tersine, 802.11g kartlı çalışan dört ve 802.11b kartlı çalışan bir kullanıcınız varsa, aynı access point'e bağlı herkes, 802.11b CSMA/CA çalışmaya mecbur tutulur. Bu nedenle, performansı optimize etmek için tüm access point'lerinizde, sadece 802.11b kullanma modunu kapatmanız tavsiye edilir.

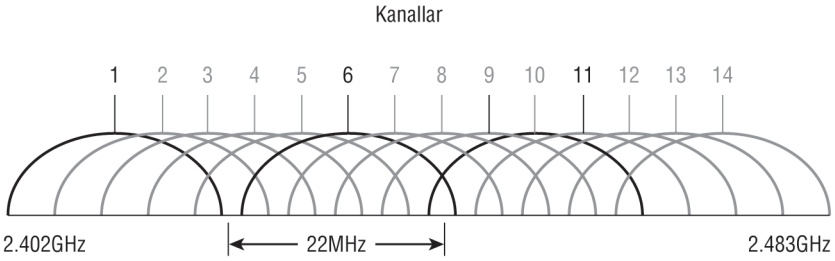
Biraz daha açıklamak gerekirse, 802.11b, 802.11g ve 802.11a tarafından kullanılan Orthogonal Frequency Division Multiplexing (OFDM) modülasyonu kadar güçlü olmayan, Direct Sequence Spread Spectrum (DSSS) olarak bilinen bir modülasyon tekniği kullanır. OFDM kullanan 802.11g istemcileri, 802.11b istemcileri ile aynı hızlarda çok daha iyi performans sağlarlar. 802.11g istemcilerin, 802.11b hızlarında (11, 5.5, 2 ve 1Mbps) çalıştığında, aslında 802.11b'nin kullandığıyla aynı modülasyonu kullandığını hatırlayın.

Şekil 12.3, FCC'nin, 2.4GHz hızında yayınladığı her biri 22Mhz genişliğinde, 14 farklı kanalı göstermektedir.

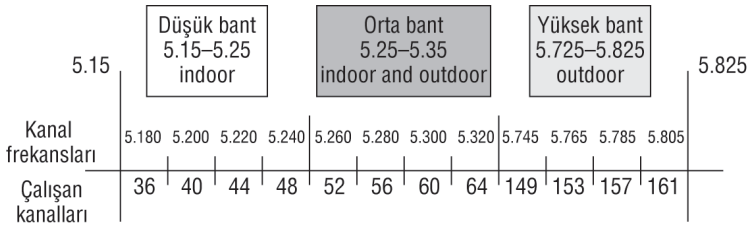
U.S.'de çakışma olmaksızın, kanal1, 6 ve 11 ile yapılandırılabilir sadece 11 kanal vardır. Bu, sinyal karışımı olmadan aynı alanda üç access point'e sahip olmanıza izin verir.

## 5GHz (802.11a)

IEEE, 802.11a standardına, 1999'da onay verdi, fakat ilk 802.11a ürünleri, 2001'e kadar marketlerde görünmedi. Ve onlar çok pahalıydı. 802.11a, 12 çakışmayan frekans kanalıyla, maksimum 54Mbps veri hızı sağlayan bir standarttır. Şekil 12.4, UNI bantlarını göstermektedir.




***Şekil 12.3:*** *ISM 2.4GHz kanalları.*




***Şekil 12.4:*** *UNII 5GHz bant, 12 çakışmayan kanala sahiptir (U.S.).*

Farklı frekanslarda olduklarından, 5GHz radyo bandında çalışarak, mikrodalga fırınlar, kablosuz telefonlar ve Bluetooth cihazlar gibi 2.4GHz bantta çalışan cihazlardan etkilenmez. Bu nedenle, ağınız sadece upgrade olan bölümünden ibaret olmadığından, her şeyin mükemmel bir uyum içinde birlikte çalışmasını beklemeyin. Fakat yine de üzülmeyin, her iki tip ağda çalışacak çok sayıda dual-radyo cihazları vardır. 802.11a için pozitif özellik, aynı fiziksel ortamda 802.11b kullanıcılarından etkilenmeden çalışabilmesidir.

802.11b radyolarına benzer şekilde tüm 802.11a ürünleri, taşınırken veri hızı değişimi kabiliyetine sahiptir. 802.11a ürünleri 54Mbps'da çalışan birine 48Mbps, 36Mbps, 24Mbps, 18Mbps, 12Mbps, 9Mbps ve son olarak AP'tan en uzaktaki nokta ile hala haberleşmek için 6Mbps'a geçmeye izin verir.

Ayrıca 802.11h olarak belirtilen, 802.11a düzenlemesinin bir uzantısı vardır.

## 5GHz (802.11h)

FCC, Şubat 2004'te 11 yeni kanal ekledi ve 2008'de, üreticilerin daha fazla 802.11a 5GHz ürünü yayınlamasına bağlı olarak bu kanalları kullanmaya başlamıştır. Yani, 23 adet çakışma olmayan kanala sahip olabileceğiz. 802.11h düzenlemesinin parçası olan, iki yeni 5GHz radyo özelliği vardır: Transmit Power Control (TPC) ve Dynamic Frequency Selection (DFS).

**DFS:** Bu güzel özellik, aktarımdan önce, hem 5GHz bandın bölümlerinde hem de 802.11a'da çalışması kabul edilen radar sinyalleri için bir cihazın çalışma aralığını sürekli görüntüler. Şayet DFS, bir radar sinyali algılarsa, kullanılan kanalı terk edecektir veya WLAN'da çakışma olmasından korumak için kanalı kullanılamaz olarak işaretleyecektir.

**TPC:** Uzun süredir, cep telefonu sanayi tarafında kullanılıyor olsa da, bu teknolojinin bazı yeni maharetli kullanımları vardır. İstemci makinesinin adaptörünü ve access point'in aktarım gücünü, farklı aralıklarda ayarlayabilirsiniz. Bu, birçok sebepten çok kullanışlı bir özelliktir. Bunlardan biri, access point'in aktarım gücünü 5mW'a ayarlamak hücre hızını düşürür. Şayet, yüksek-yoğunluklu kullanımlı sıkışık bir alana sahipseniz, çok iyi çalışmaktadır.

TPC, istemci ve access point'lerin haberleşmesini mümkün kılmayı da içeren birçok avantaja sahiptir. Yani, istemci makine aktarım gücünü dinamik olarak ayarlayabilir, böylece access point ile bağlantısını korumak için sadece yeterli miktarda enerji kullanır, pil gücünü korur, artı komşu WLAN hücrelerindeki çakışmayı azaltır.

## 802.11 Karşılaştırmaları

Cisco'ya özgü cihazlara geçmeden önce, 802.11a, b ve g ile ilgili olumlu ve olumsuz yönleri listeleyen Tablo 12.3'e bakalım.

**Tablo 12.3:** 802.11 Karşılaştırması

| 802.11b | 802.11g | 802.11a (h) |
|---|---|---|
| 2.4GHz | 2.4GHz | 5GHz |
| En yaygını | Daha yüksek throughput | En yüksek throughput |
| 11Mpbs'e kadar | *54Mbps'e kadar | 54Mbps'e kadar |
| DSSS | DSSS/OFDM | OFDM |
| 3 çakışma olmayan kanal | 3 çakışma olmayan kanal | 23'e kadar çakışma olmayan kanal |
| **Hücre başına yaklaşık 25 kullanıcı | Hücre başına yaklaşık 20 kullanıcı | Hücre başına yaklaşık 15 kullanıcı |
| Çoklu yol ile sınırlı uzaklık | 802.11b istemcileri tarafından düşürülen throughput | Daha düşük market yaygınlığı |

**11Mbps ve altı hızlarda 802.11b çalıştığında, Direct Sequence Spread Spectrum çalışır.*

***Bu Cisco'nun genel prensibidir. Hücre başına kullanıcıların gerçek sayısının birçok etkene bağlı olduğunu bilin.*

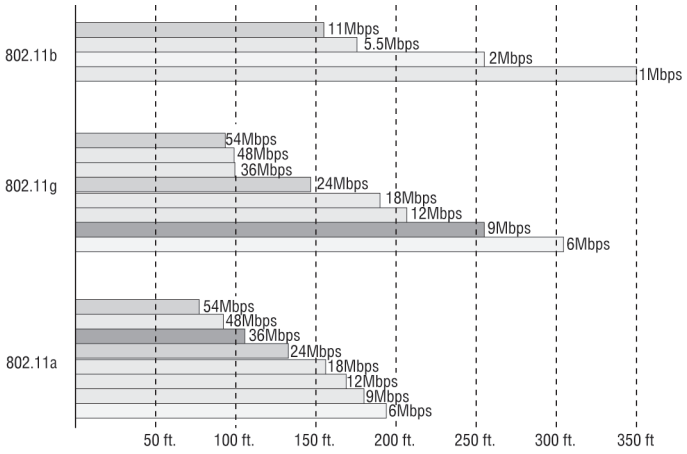
Şimdi, her 802.11 standardının hız karşılaştırmasını ve bir faktör olarak bina içi açık-ofis ortamını kullanan farklı hızları gösteren Şekil 12.5'e bakalım.

802.11a ve 802.11g'nin tam 54Mbps avantajına sahip olmak için 15 ile 30 metre, hatta istemci ile access point arasında engeller varsa, daha bile az uzaklıkta olmanız gerektiğini görebilirsiniz.

Uzak mesafelerde, yüksek hızlara sahip olmak için kullanacağımız, bir IEEE 802.11 standardından daha bahsetmek istiyorum.

## 2.4GHz/5GHz (802.11n)

802.11n,Multiple-Input Multiple-Output (MIMO) ekleyerek, önceki 802.11 standartları üzerine kurulmuştur. Veri throughput'unu arttırmak için çoklu alıcı ve vericiler çalıştırır. 802.11n, sekize kadar antene sahip olabilir, fakat günümüz access point'lerinin çoğu, dört tane kullanır. Bunlar bazen akıllı anten olarak belirtilir ve dört tane kullanırsanız ikisi eşzamanlı aktarmak, ikisi de almak için kullanılacaktır. Bu kurulum 802.11a/b/g den daha yüksek bir veri hızı sağlayacaktır.




***Şekil 12.5:*** *802.11 standartlarının hız karşılaştırması.*

Aslında pazarlama elemanları yaklaşık 250Mbps sağlayacağını iddia etmektedirler, fakat kişisel olarak onu satın almam. Bunun gerçek throughput seviyelerimiz olabileceğine inanmıyorum. Söyledikleri doğru olsa bile, hepimiz, internete çıkmak için 1 veya 2Mbps kablo veya DSL bağlantısına sahipken nasıl yardımcı olabilir?

802.11n standardının henüz onaylanmadığını ve 2008'de belli bir zamana, belki de daha sonrasına kadar beklenmediğini unutmayın. Yani, raflardaki ürünler tescillidir ve pre-N ürünler olarak belirtilirler.

Tüm bunları aklımızda tutarak, büyüyen wireless pazarına Cisco çözümlerine bir bakalım.

# Cisco Unified Wireless Solution

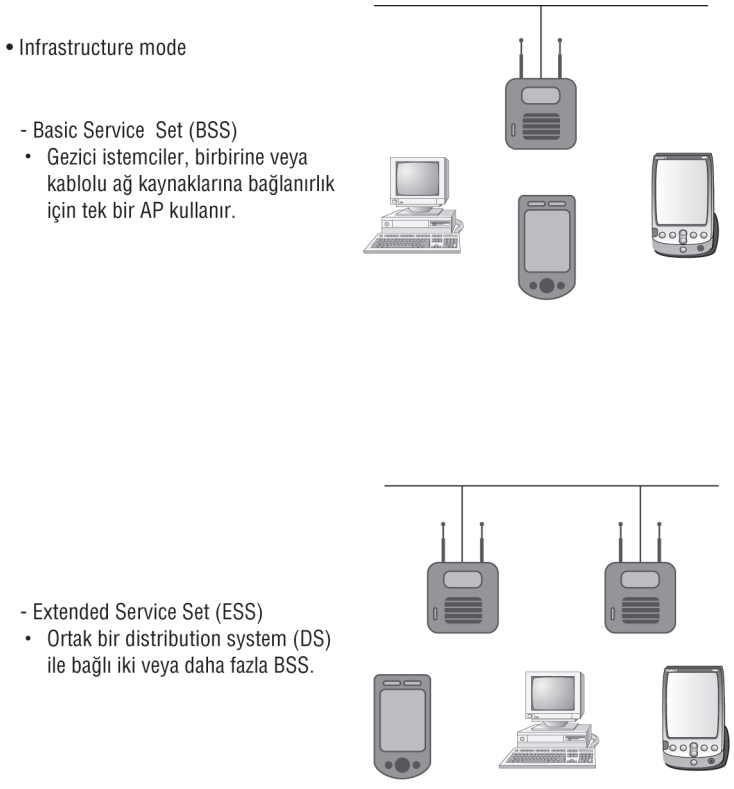
*Dürüst olmak gerekirse, beyniniz doluysa ve CCNA Composite sınavınız için çok çalıştıysanız, muhtemelen bu bölümü atlarsınız. Fakat bunu yapmadan önce, emin olun ve Cisco CCNA Composit 640-802 sınav konularındaki son pürüzler için www.lammle.com internet adresini kontrol edin.*

NOT

IEEE 802.11a/b/g ve n teknolojilerini destekleyen ürünlerin kapsamıyla, Cisco aslında, bina içi ve dışı wireless LAN hakkında oldukça güzel ve etkili çözümler önermektedir.

Bu ürünler; access point'leri, wireless controller'ları, wireless LAN istemci adaptörlerini, güvenlik ve yönetim sunucularını, wireless yönetim cihazlarını, wireless tümleşik switch ve router'ları, hatta anten ve aksesuarları bile içermektedir.

2000 yılından beri birçok firma, kendi ana ağlarındaki basit access point'lere güvendiler ve onları, ağlarında kullanıcıların dolaşımına izin veren bir alt yapıya bağladılar. Şekil 12.6, bir access point olan veya dolaşım maksadıyla, tamamı aynı Service Set Identifier (SSID) kullanan çoklu access point'lere sahip olacağınız genişletilmiş bir servis düzeneği şeklinde tipik bir ağı göstermektedir.




***Şekil 12.6:*** *Tipik altyapı ağı.*

Burada her iki yapılandırmada da AP'lerin her birinin bir root AP olarak yapılandırıldığını görürüz. Şayet bölüm 4'teki, iki wireless router'ım (871W ve R2) ve 1242AP'nin yapılandırmalarına geri dönüp bakarsak, üçünün de, root olarak yapılandırıldığını görürüz. Basit olarak bunun anlamı, "wireless istemcisi, bana bağlan ve ihtiyaçlarına (kablolu kaynaklara) sahip ol" demektir. Şayet AP'ler root olmasalardı, sadece, root bir cihaza, repeater olarak bağlanabilirlerdi. Root olmayan cihazlar istemcileri, bridge'leri, repeater access point'leri ve workgroup bridge'leri içermektedir.

Fakat bir dakika durun, bu, IT Cretaceous çağı gibiydi. Fakat nerdeyse 2008' geldiğimiz bu zamanda kesinlikle öyle değil. Kapsamlı, tümleşik bir WLAN çözümü sağlayan Cisco Unified Wireless Solution'e sahibiz. Bu, akıllı Cisco IP'leri ve özellikle AP'leri desteklemesi için tasarlanan Cis-

co WLAN controller'ları içeren yeni bir teknolojidir. Bu çözüm, controller web ara yüzü boyunca, controller'ın kendisi veya Cisco's Wireless Control System'den (WCS) yönetilir.

Bu tür ağlar hakkındaki asıl güzel şey, başlangıç kurulumundan sonra, yapılandırma istememesidir. Yani, bina içi veya bina dışı ortamdaki bir AP'ye bağlanabilirsiniz ve AP, controller bilgisine bağlı olarak kendisisini otomatik olarak yapılandıracaktır. Kanal için çakışmayı ve sinyal karışmasını bile kontrol edecektir ve onu, çakışma olmayan bir kanal olarak atayacaktır. Ne kadar güzel, değil mi? Daha önce özellikle belirttiğim gibi kendi bölgesinde çakışan bir kanal tespit ederse, sinyal karışmasını sınırlandırmak için aktarım seviyesini düşürecektir. Cisco, bunu otomatik RF kontrolü olarak belirtir.

Fakat her şey mükemmel değildir. Bu ürün ailesi fakirler için değildir, çünkü zulanızdakileri ortaya çıkarmanız gerekebilir. Emin olmak için birden fazla AP'ye sahip olmanız gerekecektir. Minimum alışveriş listeniz, bir Cisco 1020 AP ile bir bina içi çözüm için bir ve bina dışı için bir controller, bir Cisco 1520 AP ile bir controller olmalıdır. Ve bunların minimum ihtiyaçlar olduğunu hatırlayın. Gerçek hayatta muhtemelen daha fazlasına ihtiyaç duyulacaktır. Sınıflarımda 1020 ve 1520 AP kullanıyorum ve onları iki controller tipiyle birlikte kitap yazmak için kullanıyorum (kesinlikle, işler hale getirmek için kullanabileceğim minimum sayıda cihaz). AP'lerin fiyatları makuldür ve bu Cisco için olağandır. Onların fiyatları, ürün model numarasını izler. Controller'ların fiyatları oldukça yüksektir, çok pahalıya mal olurlar. Umut ediyorum ki, siz bu kitabı okurken, gökkubbe altında bir yerlerde bedeli düşecektir. Her zaman bir ümit vardır!

Sadece eğlence için limitsiz fonla harcama yaparak iyi ölçülerde bir ağa sahip olduğunuzu düşünelim. İlk olarak, en az iki controller alalım (iyilerinin tanesi yaklaşık 20.000$'dır). Neden iki? Çünkü her AP'den gelen paket, kablolu ağa veya tekrar geri wireless ağa yerleşmesi için, controller'a gitmek zorundadır. Controller, onu enkapsüle eden Lightweight Access Point Protocol'e (LWAPP) bağlı olarak paketin gidişatına karar verir. (LWAPP hakkında birazdan daha çok bahsedeceğim). Sizin bunların en az iki tanesine ihtiyaç duymanızın sebebi, onlardan birinin arızalanması talihsizliğinin olması ihtimalidir. Tek hata noktasına sahip olacak şekilde tasarım yapmanız için deli olmanız gerekir. Tamamıyla akıllı davranmak için iki adet alırsınız ve bu tür ağlar için gereken yedeklemeyi sağlarsınız.

Tek hata noktası gibi şeyleri de görünce, controller'larınızı da yönetebilmeniz gerekir. Cisco, tek bir ara yüzden tüm WLAN'ı yönetmek için Wireless Control System (WCS) GUI'ye sahiptir. Bu, cihaz lokasyonlarının detayına, ağ istatistiklerinin eğilimine ve ağ kapsamına bazı detaylı kavrayış sağlar.(paranın mesele olmadığını hatırlayın!)

NOT *Cisco web sayfasından, WCS'in 30 günlük demo'sunu indirebilirsiniz.*

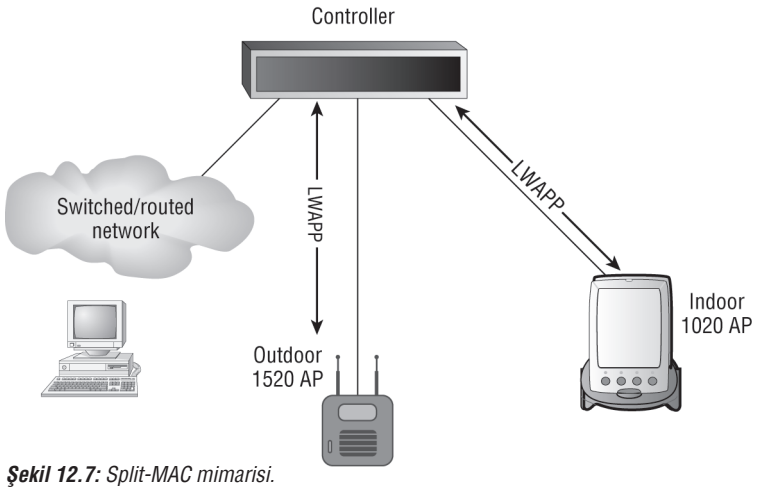
Cisco WLAN controller'ın, WCS'deki kapsamlı araçlarla yönetilebilen AP'ler tarafından toplanan veriyi analiz etmesinden dolayı, aslında WCS'e sahip olmanıza gerek yoktur. Fakat zengin olduğunuzdan, kendinizi bu WCS'e bağlayacaksınız.

İşleri sizin için biraz daha güçleştirmek için, controller'lar, gigabit interface'lerle gelecektir. Yani, AP bağlantılarınız için 10/100 portlara ve controller'larınıza bağlanmak için bir gigabit porta sahip bir switch'e ihtiyacımız var. Her iki tip porta sahip, minimum 3560 (veya daha iyisi) switch olması muhtemelen en iyisi olacaktır. Böylece VLAN'lar arası routing'de sağlayabilirsiniz.

## Split-MAC Mimarisi

Evet, kulağa biraz garip geliyor, fakat bu garip isim, aslında oldukça güzel bir özelliktir. İki cihaz, AP ve merkezi Cisco WLAN controller arasındaki 802.11 protokolün işlemini ayırıyoruz. Şekil 12.7, her lokasyonda olan işleyişin nasıl ayrıldığını göstermektedir.

Şekil 12.7'de, 1520 AP ve 1020 AP, controller'a direkt bağlı görünse de, birincisi 10/100'ü gigabit'e dönüştürmesi için bir switch'le bağlanmaya ihtiyaçları olması, ikincisi de controller'ların, sadece LWAPP-uyumlu porttan gelen LWAPP paketlerini iletmesinden dolayı, onlar direkt bağlı olamazlar. Yani, bir LWAPP paketini alıp, onu IP verisi gibi, LWAPP olmayan bir ağa iletmek istiyorsanız, bir router'a ihtiyacınız vardır. Yüksek kapasiteli bir uç switch, routing'i sağlayabilir.




***Şekil 12.7:*** *Split-MAC mimarisi.*

AP, gerçek zamanlı ihtiyaçlara sahip protokol parçaları sağlar:

- Havadan bir frame aktardığında, frame, bir istemci ve AP arasında frame değişim anlaşması.
- Uyarıcı frame'leri aktarmak.
- Güç koruma işlemlerinde, istemciler için frame'lerin belleklenmesi ve aktarılması.
- İstemcilerden gelen frame isteklerini incelemekten sorumlu olmak
- Alınan inceleme istekleri uyarılarını, controller'a iletmek.
- Her alınan frame ile controller'a, gerçek zamanlı sinyal kalite bilgisi sağlamak.
- Her radyo kanalının gürültü, sinyal karışması ve diğer WLAN'ların görüntülenmesi.
- Diğer AP'lerin varlığının görüntülenmesi.
- VPN/IPSec istemcileri dışındaki durumlarda şifreleme ve şifre çözme.

Kalan tüm işlevsellik, Cisco WLAN controller'da ele alınır. Bu nedenle zaman hassasiyeti bir endişe oluşturmaz. Aşağıda, WLAN controller'da sağlanan bazı MAC-katman özellikleri vardır:

- 802.11 kimlik doğrulaması
- 802.11 association ve reassociation (mobility)
- 802.11 frame çevrimi ve bridging

Şayet Appliance moddaki bir Cisco Wireless Controller arızalanırsa, onun kapanan Cisco AP'leri, başka Cisco Wireless Controller için ağı tarayacaktır. Bir çevrim içi Cisco Wireless Controller, kalan bir AP portuna sahip olduğunda, yönetim ara yüzü, otomatik tespit etmek, ilişkilendirmek ve olabildiğince çok Cisco AP ile haberleşmek için, Cisco AP toplama mesajları için ağı dinler.

*Aslında split-MAC mimarisi, protokolün gerçek zamanlı bölümlerini kullanan Cisco LWAPP-tabanlı AP ve zaman duyarlı olmayan öğeleri kullanan WLAN controller arasındaki 802.11 protokol paketlerinin ayrılmasına izin verir.*

NOT

## MESH ve LWAPP

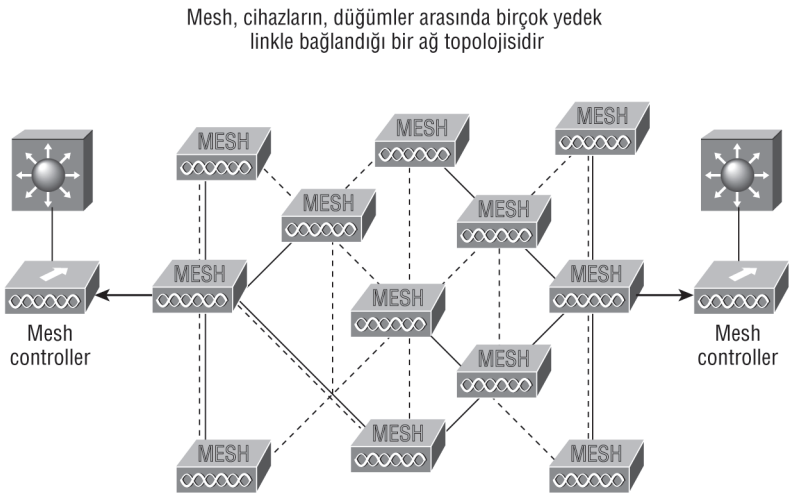
Birçok üretici bir mesh hiyerarşik tasarıma geçtikçe ağlar büyüyecek ve bu ağlarda çok gelişmemiş access point'ler kullanılmış olacağından, bu basit access point'lerin, WLAN sitemleriyle haberleşmelerini yönetecek standart bir protokole ihtiyacımız olacaktır. Bu tamamıyla, Internet Engineering Task Force'un (IETF) son taslak düzenlemelerinden biri olan Lightweight Access Point Protocol (LWAPP) tarafından üstlenilen bir roldür.

LWAPP ile büyük birden fazla üreticinin cihazı olan wireless ağları, maksimum kapasite ve arttırılmış esneklikle düzenlenebilmektedir. Bu neredeyse doğrudur. Hiç kimse, Cisco ve Motorola ağlarını aynı şirkette yayınlamaz ve kendini beğenerek oturup, "bu çok güzel oldu" demez. Cisco, Cisco'dur, Motorola, Cisco değildir. Onlar, sözüm ona aynı IETF protokolleri çalıştırsalar da, standartları aynı yollarla anlıyor gibi görünmemektedirler. Aslında, birlikte çok iyi çalışmazlar.

Bu nedenle, sadece Cisco kullandığımızı farz edelim. (Zaten limitsiz bir bütçemiz var, hepsini niye Cisco koymayalım ki, bu bir Cisco kitabı değil mi?)

Düğümler, verileri en yakın düğümlerden, yönetilebilir kablolu bağlantıya, çok uzak kişilere aktarmak için repeater'lar gibi davranırlar. Bu, bir ağın, özellikle de engebeli veya zor bölgelerde gerçekten uzak mesafelere yayılmasına sebep olur. Şekil 12.8, wireless bağlanırlıkla, bir alanı bütünüyle kapsaması için Cisco 1520 AP'ler kullanan büyük bir mesh ağı göstermektedir.

Artı, mesh ağlar, aşırı derecede güvenli olmaktadır. Her düğüm, imkan dahilinde birçok farklı düğüme bağlandığından, onlardan birisi, donanım arızası veya başka sebepten, ağdan düştüğünde, komşuları kolayca başka route bulacaktır. Böylece, daha fazla düğüm ekleyerek, fazladan kapasite ve hata toleransına sahip olursunuz.




***Şekil 12.8:*** *Tipik Geniş mesh bina dışı ortamı.*

AP düğümleri arasındaki kablosuz mesh bağlantıları bir radyo sinyali tarafından oluşturulur. Bu, tek bir düğümden diğer düğümlere birçok uygun yol sağlar. Mesh network boyunca kullanılan yollar, trafik yükleri, radyo şartları veya trafik önceliklerine bağlı olarak değişebilir.

Cisco LWAPP-uyumlu mesh access point'ler, Cisco Mesh Networking Solution'da yerleştirilmiş bir Cisco Wireless LAN Controller kullanılarak yapılandırılır, görüntülenir ve çalıştırılır. Onlar, bir controller'ı geçmek zorundadırlar, bundan dolayı, yedekli controller'a sahip olmak kesin bir ihtiyaçtır.

Mesh ağlarda kullanılan birkaç terimi açıklayalım.

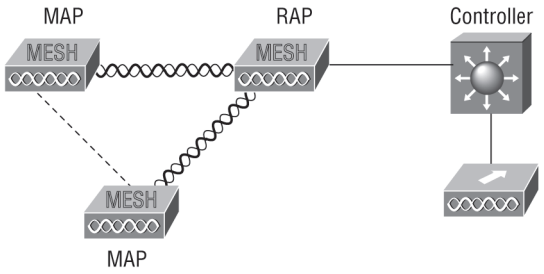
**Root Access Point (RAP):** Bu access point, kablolu ağlara veya sunuculara root ya da kablolu ağ için gateway olarak bağlanırlar. RAP'lar, Cisco Wireless LAN Controller'a kablolu bir bağlantıya sahiptir. Komşu Mesh AP'lerle haberleşmek için backhaul wireless interface kullanırlar.

**Mesh Access Point (MAP):** Mesh AP'ler tipik olarak, çatı veya kulelerde olan uzak AP'lerdir ve bir 5GHz backhaul üzerine 32 MAP'e bağlanabilir. Bootup esnasında, bir access point, kablolu ağa bağlıysa, bir RAP olmaya çalışacaktır. Tersine, bir RAP, kablolu ağa bağlantısını kaybederse, bir MAP olmaya çalışacaktır ve bir RAP arayacaktır.

Tipik bir mesh network, Şekil 12.9'da görünen cihazları içerecektir.

Şekil 12.9'da, altyapıya bağlı bir RAP ve hem birbirine hem de RAP boyunca controller'a bağlı MAP'leri görebilirsiniz.

Fakat tamamıyla bitirmedik henüz. Wireless güvenliğine geçmeden önce, bir mesh terimini daha açıklamak istiyorum: AWPP.




***Şekil 12.9:*** *Cisco mesh network'ünde bulunan makineler.*

## AWPP

Her AP, özellikle kablosuz ortamlar için Cisco tarafından tasarlanan yeni bir protokol olan Adaptive Wireless Path Protocol (AWPP) çalıştırır. Bu protokol, RAP'lerin, RAP'den geçerek kablolu ağlara tekrar en iyi yolu belirlemek için birbirleriyle haberleşmelerini sağlar. En iyi yol kurulunca AWPP, topolojinin değişmesi ya da şartların link gücünün azalmasına sebep olduğu durumlarda, RAP'a alternatif route'ları oluşturmak için arka planda çalışmaya devam eder.

Bu protokol mesh'in otomatik yapılandırma ve iyileştirme yapabilmesi için belirli radyo sinyal karışımlarını ve özelliklerini düşünür. AWPP aslında, wireless ortamla ilgili tüm öğeleri hesaba katma kabiliyetine sahiptir. Böylece, mesh ağın işlevselliği bozulmaz ve sürekli iletişim sağlayabilir.

Bu wireless ortamının gerçekten ne kadar dinamik olduğu göz önüne alınırsa oldukça güçlüdür. İnterface olduğunda veya AP'ler eklenip çıkartıldığında, Adaptive Wireless Path Protocol, rooftop AP'lere (RAP) geri dönüş yollarını tekrar yapılandırır. Yüksek dereceli dinamik kablosuz ortama karşılık AWPP, route'ları azaltmak için yapışkanlık faktörü kullanır. Bu, bir kamyonun mesh'den geçerken, geçici kesilmeye sebep olması gibi bir olayın, mesh'in gereksiz değişmesine sebep olmamasını garantiye alır.

*Cisco Mesh ve Mobility (1500/3200 Integration) ürün ailesiyle pratik eğitim için www.globaltraining.com internet adresine bakın.*

NOT

## Wireless Güvenliği

Varsayılan olarak wireless güvenliği, access point'lerde ve kullanıcılarda yoktur. Orijinal 802.11 komitesi, bir gün wireless host'larının, sınırlı ortam host'larının sayısını aşacağını hayal edememişler. Fakat bu yönde olduğumuz gerçektir. Ayrıca maalesef, IPv4 protokolleri ile olduğu gibi, mühendis ve bilim adamları, bir şirket ortamında kullanmak için yeteri kadar sağlam güvenlik standartları eklememişlerdir. Böylece, güvenli bir kablosuz ağ oluşturmak için isteklerimizde bize yardımcı olması amacıyla, tescilli çözüm eklentilerini geliştirmekle sorumlu tutulduk. Dünyamız karmaşık bir yer, bu nedenle güvenlik çözümlerimizde olacaktır.

Orijinal 802.11 standartlarına eklenen standart temel güvenliği ve günümüz sorunlarıyla ilgili, güvenli bir wireless network oluşturmamızı mümkün kılması için bu standartların neden zayıf ve yetersiz olduklarını tartışarak başlamak için iyi bir yerdeyiz.

### Open Access

Tüm Wi-Fi Sertifikalı LAN ürünleri, güvenlik özellikleri kapalı olarak open access modda gönderilir. Serbest erişim veya güvenlik olmaması kafeteryalar, kolej kampüsleri ve belki havaalanları gibi kamusal yerler için uygun ve makul olabilir. Fakat şirket organizasyonları için kesinlikle bir alternatif değildir ve hatta özel ev ağınız için bile uygun değildir.

Güvenliğin, şirket ortamlarındaki kurulumlarınız sırasında wireless cihazlarda aktif olması gerekmektedir. Şaşıracaksınız ama bazı şirketler, herhangi bir WLAN güvenlik özelliğini aktif hale getirmez. Açıkçası, bunu yapan firmalar, ağlarını büyük bir riske maruz bırakırlar.

Ürünlerin serbest erişim hakkıyla gelmesi nedeniyle, bilgisayarlar hakkında hiçbir şey bilmeyen herhangi biri, access point alabilir, onu kablo veya DSL modemine takabilir ve kolayca ağa bağlanabilir. Yapılması, basit ve kolaydır.

## SSID'ler, WEP ve MAC Adres Kimlik Doğrulaması

Orijinal 802.11 tasarımcılarının basit güvenlik oluşturmak için yaptıkları, Service Set Identifiers (SSID), açık veya shared-key authentication, statik Wired Equivalency Protocol (WEP) ve seçmeli Media Access Control (MAC) authentication kullanılmasını içermektedir. Çok şey yapıyor gibi geliyor, fakat bunların hiç birisi, ciddi güvenlik çözümü önermez. Bunların hepsi belki genel ev kullanımı için uygun olabilir. Fakat biz yine de onları inceleyeceğiz.

SSID, kablosuz ağ oluşturan bir WLAN sistemindeki cihazlar için genel ağ ismidir. Bir SSID, SSID'si olmayan bir istemci cihazın erişimini engeller. Varsayılan olarak bir access point SSID'sini, yayın alanında saniyede çok sayıda broadcast ile yollar. SSID broadcast işlemi kapalı olsa bile, kötü bir adam, ağı görüntüleyerek ve istemcinin access point'e cevabını bekleyerek, SSID'yi tespit edebilir. Neden? Çünkü ister inanın ister inanmayın, orijinal 802.11 şartnamesine uygun düzenlenen bu bilgi açık gönderilmek zorundadır.

IEEE 802.11 komitesi tarafından belirlenen iki kimlik doğrulama tipi vardır: Open ve shared-key authentication. Open authentication, sadece doğru SSID sağlamaz. Fakat günümüzde en yaygın kullanılandır. Shared-key authetication ile access point istemci cihaza, bir challenge-text paketi gönderir. İstemci sonra, doğru Wired Equivalency Protocol (WEP) ile şifrelemeli ve access point'e geri dönmelidir. Doğru anahtar olmaksızın, kimlik doğrulaması başarısız olacaktır ve istemcinin, access point ile ilişki kurmasına izin verilmeyecektir. Fakat shared-key authentication hala güvenli kabul edilmez. Çünkü bunun üstesinden gelmek için her davetsiz misafirin yapması gereken, clear-text ve bir WEP key ile şifrelenmiş kimlik sorgusunu bulmak ve sonra WEP key'in şifresini çözmektir. Clear-text kimlik sorgusundan dolayı, günümüz WLAN'larında shared-key kullanılmaz.

Open authentication ile bir istemci, kimlik doğrulamasını tamamlayıp bir access point ile bağlantı kurabilse dahi, WEP kullanımı, istemcilerin doğru WEP anahtarına sahip olmadıkça, access point'den veri alıp göndermelerini engeller. Bir WEP anahtarı, 40 veya 128 bit'ten oluşur. Genellikle access point'de bir ağ yöneticisi tarafından tanımlanır ve tüm istemciler bu access point ile haberleşirler. Statik WEP anahtarları kullanıldığında, bir ağ yöneticisi, WLAN'daki her cihazda aynı anahtarın girilmesi ile ilgili çalışma yapmalıdır. Günümüzün çok büyük şirket wireless ağlarında yönetimsel olarak imkansız olduğundan, bunun zorluklarını biliyoruz.

Son olarak, istemci MAC adresleri, her access point'e statik olarak girilebilir ve onların hiç birisi, fitler tablosundaki MAC adreslerinin erişimine izin verilmedikçe, görünmezler. Kulağa hoş geliyor, ancak tüm MAC adres bilgileri şifresiz gönderilmek zorundadır. Bedava wireless sniffer programına sahip herhangi bir kişi, access point'e gönderilen istemci paketlerini okuyabilir ve MAC adreslerini ele geçirebilir.

WEP doğru yönetildiği takdirde çalışabilir. Fakat basit statik WEP anahtarları, ona ilave olarak çalışan bazı tescilli düzeltmeler olmadan, günümüz şirket ağlarında geçerli bir seçenek değildir. Şimdi biraz bunlardan bahsedelim.

## WPA veya WPA 2 Pre-Shared Key

Şimdi bir yerlere ulaşıyoruz. Bu, düzenlemelere bir eklenti olan, basit güvenlik yönteminin diğer türü olmasına rağmen, WPA ve WPA2 Pre-Shared Key (PSK), şimdiye kadar belirttiğimiz diğer basit wireless güvenlik yöntemlerinden daha iyi bir wireless güvenliğidir. Basit olarak tanımladım.

PSK, istemci makine ve access point'de bir şifre veya kod (passphrase olarak da belirtilir) tanımlaması kullanılarak, kullanıcıları doğrular. Bir istemci sadece, access point şifresi ile kendi şifresi eşleştiği takdirde ağa erişim hakkına sahip olur. PSK ayrıca, aktarılan her paket için bir şifreli anahtar üretmek için kullanılan anahtarlama araçları da sağlar. Statik WEP'ten daha güvenliyken, PSK hala, statik WEP'teki özelliklere sahiptir. PSK, istemci makinesinde tutulur ve istemci makine, kaybolduğunda veya çalındığında, anahtarın bulunması kolay olmasa da, bir şekilde ele geçebilir. Harflerin, numaraların ve harf ve numara dışındaki karakterlerin karışımı olan güçlü bir PSK şifresi kullanılması özellikle önerilir.

Wi-Fi Protected Access (WPA), eskiden WECA olarak bilinen Wi-Fi Alliance tarafından, 2003'de geliştirilen bir standarttır. WPA, WLAN'ların kimlik doğrulaması ve şifrelenmesi için bir standart sağlar. 2003 senesi de dahil o zamana kadar mevcut bilinen güvenlik problemlerini çözmesi tasarlanmıştır. Bu, iyi reklam edilmiş AirSnort ve man-in-the-middle WLAN ataklarıyla da ilgilenir.

WPA, IEEE 802.11i standardı için bir basamaktır ve şifreleme haricinde çok sayıda ortak bileşen kullanır. 802.11i, AES-CCMP şifreleme kullanır. WPA mekanizmaları, mevcut donanım üreticileri tarafından çalıştırılabilir şekilde tasarlanmıştır. Yani, kullanıcıların, sistemlerindeki WPA'i, sadece ürün bilgisi/yazılım değişikliğiyle kullanabilmeleri gerekir.

*IEEE 802.11i standardı, WPA tarafından onaylanmıştır ve WPA versiyon 2 olarak tanımlanmıştır.*

NOT

### Cisco Unified Wireless Network Security

Cisco Unified Wireless network pek çok yeni Cisco geliştirmelerini sunar ve her bir kullanıcıya her oturumda karşılıklı kimlik doğrulaması yoluyla access control, veri gizliliği sağlayan Wi-Fi Protected Access (WPA) ve Wi-Fi Protected Access 2'yi (WPA2) destekler. Servis kalitesi (QoS) ve mobility, zengin firma uygulamaları bütünü sağlayabilmek için bu çözümde birleştirilmiştir.

Cisco Unified Wireless network şunları sağlar:

**WLAN'ler İçin Güvenli Bağlantı:** Transfer edilen verinin gizliliğini korumak için yönetilebilir, otomatik değişen güçlü dinamik şifreleme anahtarları

- WPA-TKIP, MIC, paket anahtarlarının hash edilmesi ve broadcast anahtar dönüşümü gibi şifreleme geliştirmeleri içerir.
- WPA2-TKIP, veri şifrelemesi için altın standarttır.

**WLAN'lar İçin Güven ve Özdeşlik:** Uygun istemcilerin, güvensiz ve yetkisiz access point'lerden ziyade, sadece güvenli olanlarla ilişkilendirilmesini sağlamaya yardım eden güçlü bir WLAN access control'dür. IEEE 802.1X, bir çeşit Extensible Authentication Protocol (EAP) çeşidi, bir Remote Authentication Dial-In User Service (RADIUS) ve bir Authentication, Authorization, and Accounting (AAA) sunucusu kullanan, çift taraflı kimlik doğrulaması yardımıyla her kullanıcı için, oturum başına sağlanmaktadır. Özetle aşağıdakiler sağlanmaktadır:

- Piyasadaki istemci işletim sistemleri, istemci cihazları ve 802.1X kimlik doğrulama türlerinin en geniş aralığı.
- Tüm kimlik doğrulama girişimleri için RADIUS hesap kayıtları

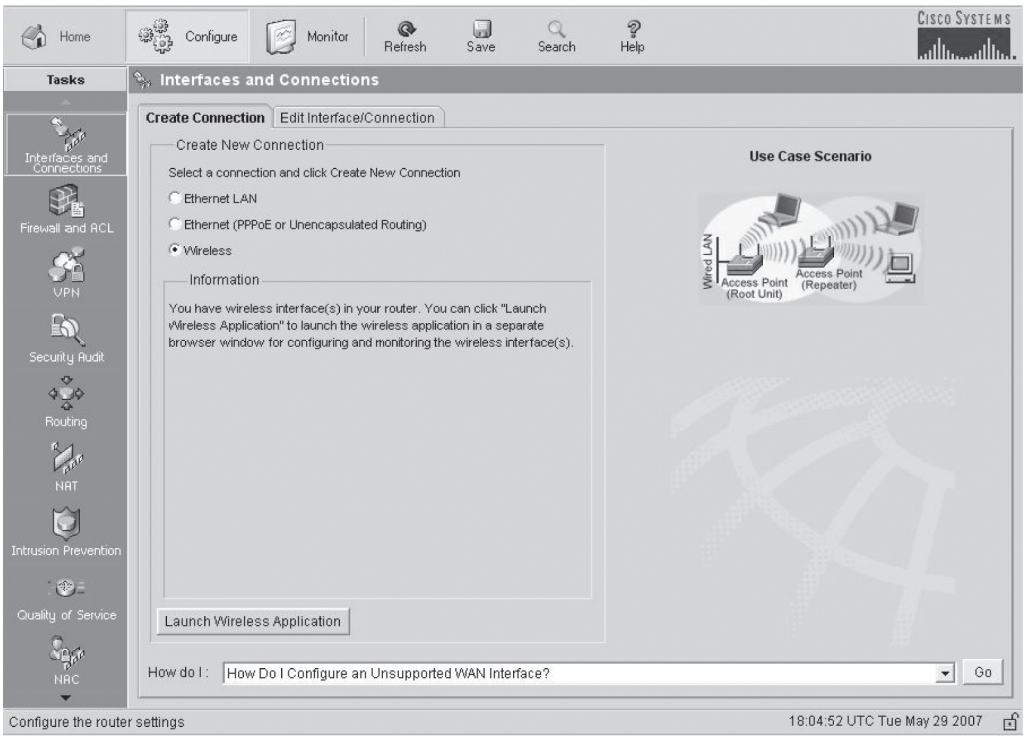
**WLAN'lar İçin Tehdit Koruması:** Intrusion Prevention System (IPS) yardımıyla, güvensiz access point'ler, ağ atakları ve yetkisiz erişim belirlenmesi, WLAN NAC ve gelişmiş lokasyon servisleri Sağlanır. Cisco IPS, IP yöneticilerinin, RF ortamını sürekli taramasına, güvensiz access point ve yetkisiz girişimleri belirlenmesine, binlerce cihazın eş zamanlı izlenmesine ve ağ ataklarının azaltılmasına olanak verir. NAC, ağ kaynaklarına erişmeye çalışan, PC, laptop, sunucu ve PDA'ler gibi kablolu ve kablosuz uç noktadaki cihazların, güvenlik tehditlerinden yeterince korunmasını sağlamaya yardımcı olması için tasarlanmıştır. NAC, kuruluşların, ağa gelen tüm cihazların kontrol ve analiz edebilmesine izin verir.

Şimdi bazı wireless cihazları yapılandıralım.

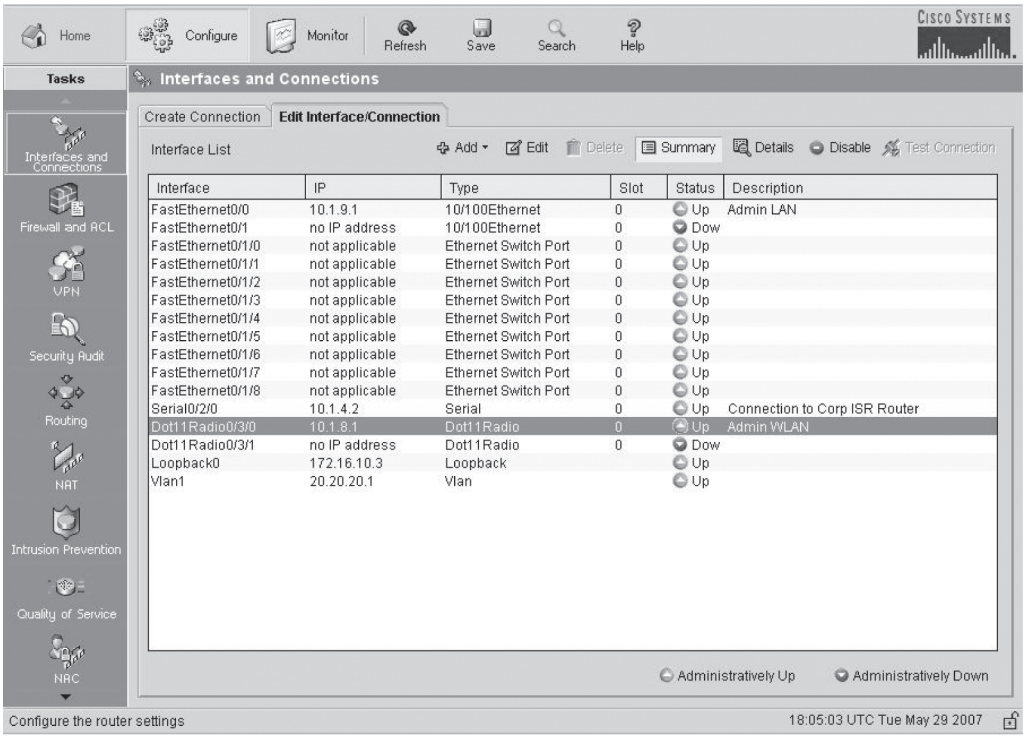
## Kablosuz Ağ Topluluğumuzu Yapılandırmak

SDM ile yapılandırma, her hangi bir güvenlik yöntemi kullanıyorsanız (tabi ki kullanmalısınız), wireless konfigürasyonu için en kolay yöntemdir. Bir access point'i aktif hale getirmek için tüm yapmanız gereken, sadece onu açmaktır. Şayet router'ınızda wireless kart kullanıyorsanız, bölüm 4'te gösterdiğim şekilde onu yapılandırmanız gerekir.

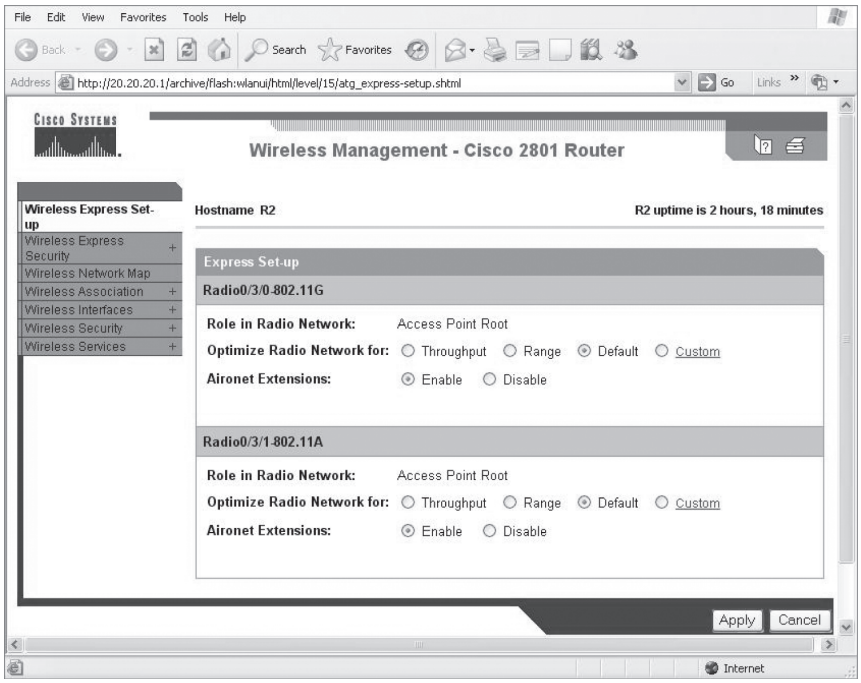
Aşağıda, slot3'te kurduğum wireless kartı yapılandırabildiğimi gösteren R2 router'ımın ekran çıktısını görebilirsiniz.




Aslında SDM'in kendisinden yapabileceğiniz çok fazla şey yoktur. Fakat Edit Interface/Connection sekmesine ve sonra Summary'e tıklarsam, hem interface'i aktif ve pasif yapabilirdim hem de Edit butonuna basarsam, NAT'ı, access list'leri ve interface'e, bazı ayarları eklememe izin verecektir.

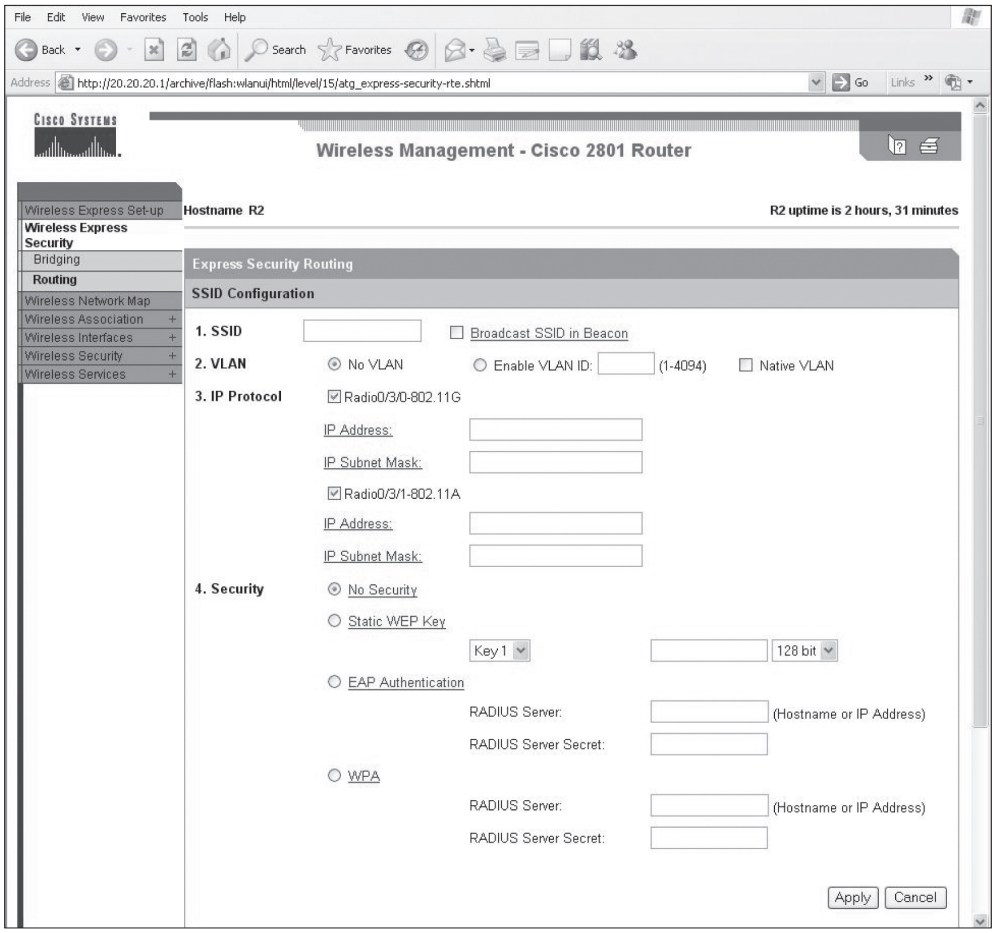



Hem bu bölümün ilk ekranında gösterilen Create Connection ekranından ya da ikinci ekrandaki Edit butonuna tıkladığınızda görünen ekrandan, Launch Wireless Application'a tıklayabilirsiniz. Bu, Exspress Set-up'dan wireless cihazınızın yapılandırıldığı, yeni bir HTTP ekranını açacaktır.

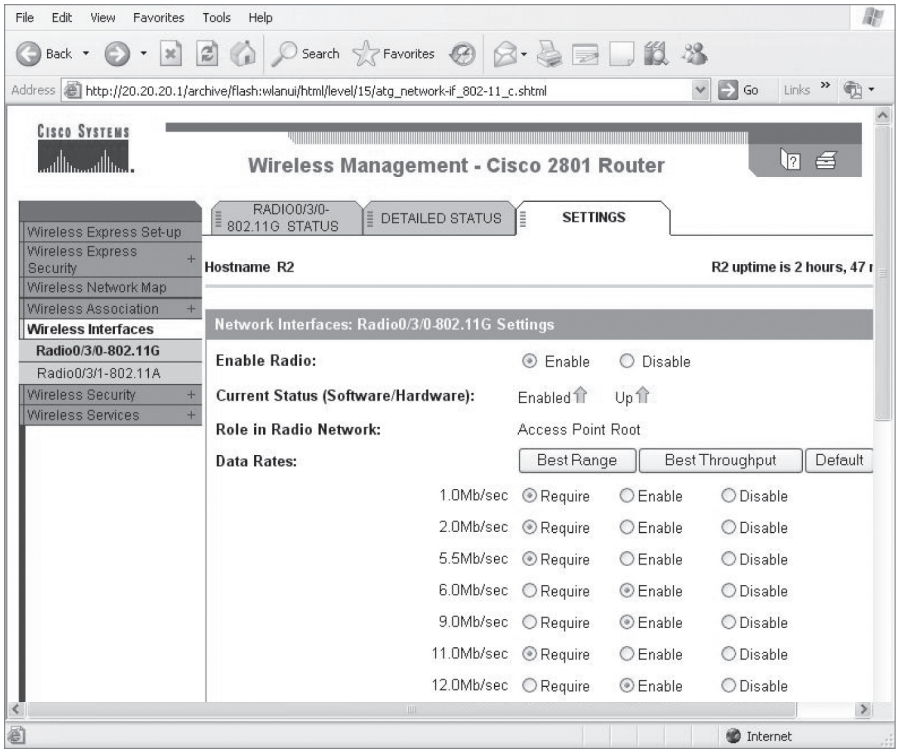



Bu, (bizim 1241 AP gibi) bir access point'a HTTP yazdığınızda görebileceğiniz ekranla aynıdır. SDM, istatistikler sağlamak, wireless interface'lere sahip bir router'da wireless configuration moda erişim sağlamak ve görüntüleme için wireless interface'lerle kullanılacaktır. Bundan dolayı, zor yapılandırmalar için CLI kullanmak zorunda değiliz.

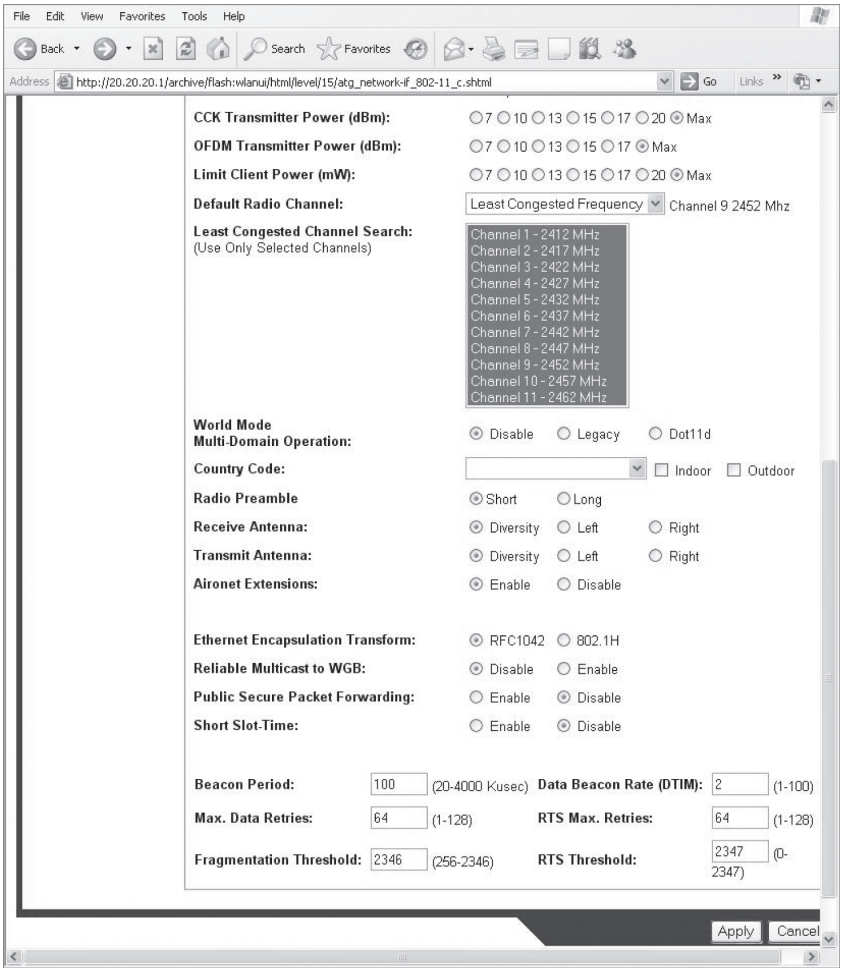
Sadece basit bazı bilgileri buradan yapılandırabilirsiniz. Fakat sonraki Wireless Express Security ekranından, wireless AP'i bridging veya routing moddan yapılandırabilirsiniz.



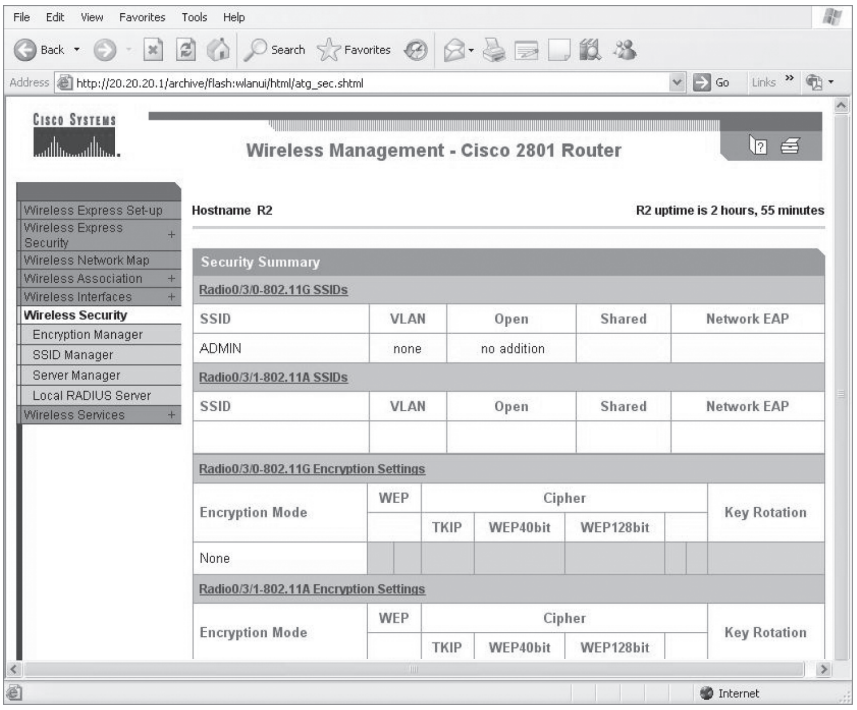
Bir sonraki ekran, wireless interface'leri veya basit ayarları gösterir.




Aşağıdaki ekran görüntüsü, Wireless Interfaces ekranının ikinci kısmıdır.



Wireless Security başlığı altında, HTTP yönetimi vardır. Şifreleme yapabilir, SSID'leri ekleyebilir ve Radius sunucu ayarlarını yapabilirsiniz.




1241AG AP'ye HTTP ile bağlanırsak, şu ekranı göreceğiz.

ISR router'larımızda bulacağımız AP'lere çok benzemektedir ve aynı cihazları ve güvenlik ayarlarını yapılandırabiliriz.

# Özet

Wireless ağlar olmadan bir dünya hayal etmek çok güçtür. Cep telefonlarından önce ne yapıyorduk acaba?

Bu bölüme, wireless ağların çalışmasının temellerini keşfederek başladık.

Burayı hızlıca geçtikten sonra, sizi wireless RF ve IEEE standartlarının temelleriyle tanıştırdım. Daha sonra 802.11'in başlangıcından, şimdiki ve gelecekteki standartlara nasıl değişim gösterdiğini tartıştık ve onları oluşturan alt komisyonlardan bahsettim.

Tüm bunlar bizi wireless güvenliğine, (daha çok, çoğu bölümü için) güvensizliği konusuna getirdi. Bu bizi, Cisco'nun bu çıkmaz için geliştirdiği çözüme yöneltti: detaylı olarak incelediğimiz Cisco Unified Wireless Solution.

Kablosuz ağlarımızı ve onunla ilgili cihazları yapılandırmak için SDM kullanarak bölümü tamamladık.

# Sınav Gereklilikleri

**IEEE 802.11a şartnamesini anlamak:** 802.11a, 5GHz spektrumunda çalışır ve 802.11h düzenlemesi kullanırsanız, 23 çakışma olmayan kanala sahip olursunuz. 802.11a, bir access point'den sadece 15 metre'den daha yakınsanız, 54Mbps'a kadar çalışabilir.

**IEEE 802.11b şartnamesini anlamak:** IEEE 802.11b 2.4GHz'de çalışır ve üç çakışma olmayan kanala sahiptir. Maksimum 11Mbps veri hızı ile uzak mesafelere ulaşabilir.

**IEEE 802.11g şartnamesini anlamak:** IEEE 802.11g, 802.11b'nin abisidir ve oda 2.4GHz'de çalışır. Şayet bir access point'ten 30 metreden daha yakın mesafedeyseniz, 54Mbps'a kadar veri hızına ulaşabilir.

# Yazılı Lab 12

Bu bölümde aşağıdaki wireless sorularına cevaplar verin:

1. IEEE 802.11b'nin veri hızı nedir?
2. IEEE 802.11g'nin veri hızı nedir?
3. IEEE 802.11a'nin veri hızı nedir?
4. IEEE 802.11b'nin frekans aralığı nedir?
5. IEEE 802.11g'nin frekans aralığı nedir?
6. IEEE 802.11a'nin frekans aralığı nedir?
7. Cisco Unified Wireless Solution'da, split-MAC mimarisi nedir?
8. IEEE 802.11h düzenlemesi, IEEE 802.11a'ya hangi iki eklentiyi ilave eder?
9. Hangi IEEE komitesi, WPA tarafından onaylanmıştır ve WPA2 olarak tanımlanmaktadır?
10. IEEE 802.11a temel standardı, kaç çakışma olmayan kanala sahiptir?

(Yazılı Lab 12'nin cevapları, bu bölüm için gözden geçirme sorularına cevaptan sonra bulunabilir.)

# Gözden Geçirme Soruları

NOT *Aşağıdaki sorular, bu bölümün materyallerini anladığınızı test etmek için tasarlanmıştır. İlave sorulara nasıl sahip olacağınızla ilgili daha fazla bilgi için bu kitabın Giriş bölümüne bakın.*

1. Cisco Unified Wireless Solution'da, split-MAC mimarisi nedir?
   A. Split-MAC mimarisi, forward/filter tablosu oluşturmak için MAC adresleri kullanır ve collision domain'lerini ayırır.
   B. Split-MAC mimarisi, iki cihaz tarafından işleme izin vermek için, AP ve controller arasındaki 802.11 protokol paketlerinin bölünmesine izin verir.
   C. Split-MAC mimarisi, wireless ağda MAC adresi ve kablolu ağda IP adresi kullanır.
   D. Split-MAC mimarisi, forward/filter tablosu oluşturmak için MAC adresleri kullanır ve broadcast domain'lerini ayırır.
2. IEEE 802.11b standardının frekans aralığı nedir?
   A. 2.4Gbps
   B. 5Gbps
   C. 2.4GHz
   D. 5GHz
3. IEEE 802.11a standardının frekans aralığı nedir?
   A. 2.4Gbps
   B. 5Gbps
   C. 2.4GHz
   D. 5GHz
4. IEEE 802.11g standardının frekans aralığı nedir?
   A. 2.4Gbps
   B. 5Gbps
   C. 2.4GHz
   D. 5GHz
5. 802.11h ile kaç adet çakışma olmayan kanal mevcuttur?
   A. 3
   B. 12
   C. 23
   D. 40
6. 802.11g ile kaç adet çakışma olmayan kanal mevcuttur?
   A. 3
   B. 12
   C. 23
   D. 40

7. 802.11b ile kaç adet çakışma olmayan kanal mevcuttur?
   A. 3
   B. 12
   C. 23
   D. 40
8. 802.11a ile kaç adet çakışma olmayan kanal mevcuttur?
   A. 3
   B. 12
   C. 23
   D. 40
9. 802.11a standardının maksimum veri hızı nedir?
   A. 6Mbps
   B. 11Mbps
   C. 22Mbps
   D. 54Mbps
10. 802.11g standardının maksimum veri hızı nedir?
    A. 6Mbps
    B. 11Mbps
    C. 22Mbps
    D. 54Mbps
11. 802.11b standardının maksimum veri hızı nedir?
    A. 6Mbps
    B. 11Mbps
    C. 22Mbps
    D. 54Mbps
12. 802.11a için maksimum veri hızıyla maksimum uzaklık nedir?
    A. Yaklaşık 19–23 metre
    B. Yaklaşık 27-30 metre
    C. Yaklaşık 46 metre
    D. 60 metrenin üzerinde
13. 802.11g için maksimum veri hızıyla maksimum uzaklık nedir?
    A. Yaklaşık 19–23 metre
    B. Yaklaşık 27-30 metre
    C. Yaklaşık 46 metre
    D. 60 metrenin üzerinde

14. 802.11b için maksimum veri hızıyla maksimum uzaklık nedir?
    A. Yaklaşık 19–23 metre
    B. Yaklaşık 27-30 metre
    C. Yaklaşık 46 metre
    D. 60 metrenin üzerinde
15. 802.11b için en düşük veri hızıyla, maksimum uzaklık nedir?
    A. Yaklaşık 30 metre
    B. Yaklaşık 53 metre
    C. Yaklaşık 90 metre
    D. Yaklaşık 105 metre
16. 802.11a için en düşük veri hızıyla, maksimum uzaklık nedir?
    A. Yaklaşık 30 metre
    B. Yaklaşık 53 metre
    C. Yaklaşık 90 metre
    D. Yaklaşık 105 metre
17. 802.11g için en düşük veri hızıyla, maksimum uzaklık nedir?
    A. Yaklaşık 30 metre
    B. Yaklaşık 53 metre
    C. Yaklaşık 90 metre
    D. Yaklaşık 105 metre
18. 18. Cisco's Unified Wireless Solution, mesh bir çözüm sağlar. Cisco çözümü çalıştırmak için hangi cihazları almanız gerekmektedir? (İki şık seçin.)
    A. WCS
    B. Controller
    C. Access point
    D. Bridge
19. Access point'inize bağlandınız ve AP'niz root olarak ayarlandı. Extended Service Set ID'nin anlamı nedir?
    A. Birden fazla access point'a sahipsiniz ve bir dağıtım sistemiyle bağlı aynı SSID'dedirler.
    B. Birden fazla access point'a sahipsiniz ve bir dağıtım sistemiyle bağlı ayrı SSID'dedirler.
    C. Çoklu access point'a sahipsiniz ve onlar fiziksel olarak ayrı binalarda bulunmaktadır.
    D. Çoklu access point'a sahipsiniz, onlardan birisi, repeater access point'dir.
20. Bir Cisco mesh ağa sahipsiniz. Hangi protokol, çoklu AP'lerin, düğümler arasında birçok yedekli bağlantıyla birleşmesine izin verir?
    A. LWAPP
    B. AWPP
    C. STP
    D. IEEE

# Gözden Geçirme Sorularının Cevapları

1. B Split-MAC mimarisi, Cisco LWAPP-tabanlı AP'ler arasındaki 802.11 protokol paketlerinin bölünmesine izin verir. Gerçek zamanlı protokol bölümlerini ve WLAN controller'ı kullanır ve bu zaman duyarlılığı olmayan öğeleri ele alır.
2. C IEEE 802.11b ve IEEE 802.11b standartlarının ikisi de 2.4 GHz RF hızında çalışır.
3. D IEEE 802.11a standartlarının ikisi de 2.4 GHz RF hızında çalışır.
4. C IEEE 802.11b ve IEEE 802.11b standartlarının ikisi de 2.4 GHz RF hızında çalışır.
5. C IEEE 802.11h standardı, 802.11a standardının 12 çakışma olmayan kanalına, 11 ilave kanal sağlar ve toplam 23 çakışma olmayan kanal olur.
6. A IEEE 802.11g standardı, üç çakışma olmayan kanal sağlar.
7. A IEEE 802.11b standardı, üç çakışma olmayan kanal sağlar.
8. D IEEE 802.11a standardı, 12'ye kadar çakışma olmayan kanal sağlar.
9. D IEEE 802.11a standardı, maksimum 54Mbps'a kadar veri hızı sağlar.
10. D IEEE 802.11g standardı, maksimum 54Mbps'a kadar veri hızı sağlar.
11. B IEEE 802.11b standardı, maksimum 11Mbps'a kadar veri hızı sağlar.
12. A IEEE 802.11a standardı, maksimum 54Mbps'a kadar veri hızı sağlar. Fakat access point'e 19-23 metre kadar yakın olması gerekmektedir.
13. B IEEE 802.11g standardı, maksimum 54Mbps'a kadar veri hızı sağlar. Fakat access point'e 27-30 metre kadar yakın olması gerekmektedir.
14. C IEEE 802.11b standardı, maksimum 11Mbps'a kadar veri hızı sağlar. Şartlara göre, 45 metre veya biraz daha uzakta olabilirsiniz.
15. D IEEE 802.11b standardı, minimum 1Mbps'da veri hızı sağlar. Fakat yaklaşık 105 metreye kadar, en uzun uzaklığa sahiptir.
16. B IEEE 802.11a standardının minimum veri hızı 1Mbps'dır. Fakat yaklaşık 53 metre uzaklığa kadar erişebilir.
17. C IEEE 802.11g standardının minimum veri hızı 6Mbps'dır. Fakat yaklaşık 90 metre uzaklığa kadar erişebilir.
18. B,C Cisco Unified Wireless Solution, çok güzel bir üründür. Gelişmiş cihazlar almanız gerekmektedir. Cisco yönetime sahip access point'ler ve controller, Unified Wireless Solution çalışması için almanız gereken cihazlardır.
19. A Extended Service Set ID, birden fazla access point'e sahipsiniz, onların hepsi aynı SSID'ye sahip ve kullanıcıları dolaşabilmesi için, hepsi birbiriyle aynı VLAN'da ya da dağıtım sisteminde bağlanmışlar demektir.
20. B Bir mesh'deki her AP, Adaptive Wireless Path Protocol (AWPP) çalışır. Bu protokol, RAP'lere, RAP yardımıyla, kablolu ağa geri dönüşte en iyi yolu belirlemek için diğerleriyle haberleşme izni verir.

# Yazılı Lab 12'nin cevapları

1. 11Mbps
2. 54Mbps
3. 54Mbps
4. 2.4GHz
5. 2.4GHz
6. 5GHz
7. Split-MAC mimarisi, Cisco LWAPP-tabanlı AP'ler arasındaki 802.11 protokol paketlerinin bölünmesine izin verir. Gerçek zamanlı protokol bölümlerini ve WLAN controller'ı kullanır ve bu zaman duyarlılığı olmayan öğeleri ele alır.
8. 5GHz radyo için iki yeni özellik vardır: Transmit Power Control (TPC) ve Dynamic Frequency Selection (DFS).
9. IEEE 802.11i standardı, WPA tarafından onaylanmıştır ve WPA version 2 olarak tanımlanmaktadır.
10. 12

# 13 Internet Protocol Version 6 (IPv6)

# 13 Internet Protocol Version 6 (IPv6)

- IPv6'ya Neden İhtiyaç Duyuyoruz?
- IPv6 Kullanımı ve Faydaları
- IPv6 Adreslemesi ve Terimleri
- Bir Ağ topluluğunda IPv6 Nasıl Çalışır?
- IPv6 Routing Protokolü
- IPv6'ya Geçiş
- 6to4 Tunneling
- Ağ topluluğumuzda IPv6 Yapılandırmak
- Özet
- Sınav Gereklilikleri
- Yazılı Lab 13
- Gözden Geçirme Soruları
- Gözden Geçirme Sorularının Cevapları
- Yazılı Lab 13'ün Cevapları

# Internet Protocol Version 6 (IPv6)

Umarım, Internet Protocol version 6'nın (IPv6) gerekli ayrıntılarını öğrenmeye hazırsınızdır.

Şimdiye kadar IPv4 konusunda güçlü bir kavrayışa sahip olmuş olmalısınız. Fakat yinede tekrar hatırlamaya ihtiyacınız varsa, bölüm 3'e bakabilirsiniz. Şayet IPv4 ile ilgili adres problemleri konusunda her şey net değilse, bölüm 11'i tekrar gözden geçirmelisiniz.

İnsanlar, IPv6'yı gelecek nesil Internet Protokolü olarak belirtmektedir ve IPv4'ün kaçınılmaz, adres tükenme krizine cevap olarak geliştirilmiştir. Muhtemelen IPv6 hakkında daha önceden bir şeyler duymuş olabilirsiniz. Atası olan IPv4'ün kapasitesi, onunla mukayese edilince önemsizdir. Onun, sonunda tamamıyla tarihe gömülecek olmasının sebebi budur.

IPv6 başlığı ve adres yapısı tamamıyla gözden geçirilmiştir ve IPv4'de ikinci olarak düşünülen ve ilave olan özelliklerin çoğu, IPv6'nın tamamıyla aktif standartlarıdır. IPv6, gerçekten iyi donatılmış, dengelenmiş ve internetin çılgın isteklerini karşılamaya hazırdır.

Bu bölümün sizin için kolay olmasına çalışacağıma söz veriyorum. Hatta kendinizi eğleniyor dahi bulabilirsiniz. Ben gerçekten eğleniyorum. IPv6; oldukça karmaşık, üstün, yenilikçi ve özelliklerle dolu olmasından dolayı, yeni bir Lamborghini ve gelecekle ilgili büyüleyici bir romanın garip bir kombinasyonu gibi beni hayran bırakır. Bu bölümü sakin bir sürücü gibi tecrübe edeceğinizi umuyorum.

*Bu bölüm ile ilgili son güncellemeler için www.lammle.com ve/veya www.sybex.com sitelerine bakın.*

NOT

## IPv6'ya Neden İhtiyaç Duyuyoruz?

Haberleşmeye ihtiyacımız var ve mevcut sistemimiz artık yeterince başarılı değildir. Bu sorunun kısa cevabıdır. Bu, atlı posta sisteminin, uçak postasıyla yarışamaması gibi bir şeydir. Bant genişliği ve IP adreslerini korumanın yeni yolunu bulmak için ne kadar zaman ve efor harcadığımızı düşünün. Kötüleşen adres kıtlığıyla baş edebilmek için mücadelemizde, Variable Length Subnet Mask'ı (VLSM) bile geliştirdik.

Ağa bağlanan insan ve cihaz sayısının her geçen gün arttığı bir gerçektir. Bu tamamıyla kötü bir şey değildir. Her zaman daha fazla insanı haberleştirmek için yeni ve heyecanlı bazı yöntemler buluyoruz, bu da güzel bir şey. Aslında bu temel insan ihtiyacıdır. Fakat hava durumu, her zaman tamamıyla mavi gökyüzünü ve havanın güneşli olduğunu söylemez. Çünkü bu bölümün girişinde ima ettiğim gibi şimdilik, haberleşme kabiliyetimizin bağımlı olduğu IPv4'ün kullanacağımız adresleri tükenecektir. IPv4, teoride 4.3 milyar adrese sahiptir ve bunların hepsini kullanamadığımızı biliyoruz. Aslında cihazlara atanabilmiş yaklaşık 250 milyon adres vardır. Classless Inter-Domain Routing (CIDR) ve Network Address Translation'ın (NAT) kullanımı, adreslerin kaçınılmaz azalmasını uzatmaya yardımcı olmaktadır. Fakat biz onları tüketeceğiz ve bu birkaç yıl içinde olacaktır. Çin güç bela çevrim içi olmaktadır ve olmak isteyen çok büyük nüfusun ve şirketlerin olduğunu biliyoruz. Tüm bu sayıları bize veren raporlar vardır, fakat tamamıyla emin olmanız gereken, dünyada bugün 6.5 milyar insan olduğu ve ortalama bu nüfusun %10'unun İnternet'e bağlandığıdır.

Bu istatistik, onlarla birlikte kullanılan diğer cihazları geçin, herkesin bir bilgisayara bile sahip olamadığı, IPv4 kapasitesi ile ilgili kötü gerçeği haykırır. Benim birden fazla bilgisayarım var, sizin de olması kuvvetle muhtemeldir. Ve telefon, laptop, oyun konsolu, fax cihazı, router, switch ve her gün kullandığımız diğer cihazlara sahip bile değilim. Böylece, adreslerimiz tükenmeden önce bir şeyler yapmamız gerektiği, yoksa tanıdıklarımızla haberleşme kabiliyetimizi yitireceğimizi netleştirdiğimi düşünüyorum. Ve bu sebep ile IPv6 uygulamasıyla ilgili çalışmalar yapıldı.

## IPv6 Kullanımı ve Faydaları

IPv6'yı bu kadar mükemmel yapan nedir? IPv6 gerçekten yaklaşan sorunumuza derman olacak mıdır? Hepsi güzel sorular, hatta daha fazlasını da düşünebilirsiniz. Tabi ki zamanla test edil-

miş, iyi bilinen değişime direnme sendromlu insan grupları olacaktır. Fakat onları dinlemeyin. Eğer seneler önce öyle yapmış olsaydık, postalarımızın at sırtında yerine ulaşması için haftalar, hatta aylarca bekliyor olurduk. Cevabımız kesinlikle evettir. IPv6, bize sadece çok sayıda adres sağlamaz (3.4 ¥ 10^38 = kesinlikle yeterli), ona geçmek için gerekli efora, maliyete ve zamana değecek tümleşik gelen birçok özellik sunar. Bu bölümün sonlarında, "IPv6'ya Geçiş" bölümünde, tüm bu çabalardan bahsedeceğim. Orada, versiyon4'ten versiyon6'ya dönüşüm için gerekli bazı geçiş tiplerinden bahsedeceğim. Ve size söz veriyorum, geçişin çok sayıda avantajının, aleyhte görüşleri bastıracağını keşfedeceksiniz.

Hem günümüz ağları hem de internet, IPv4 oluşturulduğunda, düşünülmeyen, çok sayıda öngörülmeyen gereksinimlere sahiptir. Bir grup eklentileyle onları gidermeye çalıştık. IPv6 gelişti ve bu özelliklerin çoğunu bir standart olarak içerdi. Bu güzel yeni standartlardan birisi, bölüm 14'de bahsedeceğim, uçtan-uca güvenlik sağlayan bir özellik olan IPSec'dir. Diğer güzel bir özellik, mobility'dir (taşınabilirlik) ve isminin belirttiği gibi bir cihaza, bağlantısı kesilmeden, bir ağdan diğerine dolaşım izni verir.

İlk olarak, bir IPv6 paketindeki başlık, alanların yarısına sahiptir ve onlar 64 bit olarak sıralıdırlar. IPv4 ile mukayese edildiğinde ışık hızında işlem hızı sağlar. IPv4 başlığına eklemek için kullanılan bilgilerin çoğu atılmıştır ve şimdi onu veya bir bölümünü, temel başlık alanlarını izleyen opsiyonel, ilave başlık formunda başlığa geri yerleştirmek için seçebilirsiniz.

Ve tabi ki, zaten bahsettiğimiz yeni bir adres uzayı vardır (3.4x10^38). Fakat onları nereden elde edebiliriz? Bu adres çoğalması bir yerlerden mi gelmek zorundadır? IPv6, büyük ölçüde geniş adres uzayı sağlar bize, yani gerçekçi olarak dört kat daha büyüktür. Bir IPv6 adresi 128 bit uzunluğundadır. Sakın üzülmeyin, adresi parça parça ayıracağım ve "IPv6 Adreslemesi ve Deyimleri" bölümünde hepsinin neye benzediğini göstereceğim. Şimdilik sadece ilave alanların, adres uzayında daha fazla hiyerarşik kademeye ve daha esnek adres mimarisine izin verdiğini söylememe izin verin. Adreslerin, çok daha verimli şekilde bir araya gelmesinden dolayı, routing'in daha etkili ve ölçeklenebilir yapılmasını sağlar. IPv6 ayrıca, host ve ağlar için çoklu adreslere izin vermektedir. Bu özellikle, availability isteyen kuruluşlar için önemlidir. Artı, yeni IP versiyonu, multicast haberleşmenin yaygın kullanımını da içermektedir (birçok cihaza veya seçili bir gruba gönderi yapabilen bir yapı) Bu, haberleşmelerin daha özel olmasından dolayı, ağlardaki verimliliği de arttıracaktır.

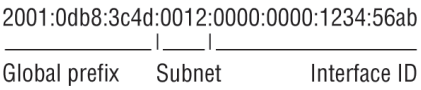
IPv4, broadcast'leri bol miktarda kullanır. Buda birçok probleme sebep olur. Bunlardan en kötüsü, aşırı miktarda iletilen broadcast'in tüm ağı çalışamaz hale getirmesi ve tüm bant genişliğini tüketmesine sebep olan, güçlü broadcast fırtınasıdır. Broadcast trafiğiyle ilgili diğer kötü durum, ağdaki tüm cihazların çalışmasını durdurmasıdır. Bir broadcast gönderildiğinde her makine, broadcast onunla ilgili olsun ya da olmasın, ne yapıyorlarsa onu durdurmak zorundadır.

IPv6'de broadcast gibi bir şey yoktur. Çünkü o, multicast trafiği kullanmaktadır. Ayrıca iki farklı haberleşme yöntemi daha vardır: IPv4'de olduğu gibi unicast ve yeni olan anycast. Anycast iletişim aynı adresin birden fazla makineye yerleştirilmesine izin verir. Böylece trafik, bu yolla adreslenmiş tek cihaza gönderildiğinde aynı adresi paylaşan en yakın host'a yönlendirilir. Bu daha başlangıç, farklı haberleşme türlerini, "Adres Tipleri" bölümünde daha detaylı göreceğiz.

# IPv6 Adreslemesi ve Terimleri

IP adreslerinin yapılandırılması ve kullanılmasının anlaşılması IPv4 ile nasıl önemli ise IPv6'da da hayati önem taşır. IPv6'nın 128 bit olduğunu ve IPv4 adresten daha uzun olduğunu zaten okudunuz. Hem de adreslerin kullanılabileceği yeni yöntemlerden dolayı, IPv6'nın yönetilmesinin daha karmaşık olacağını tahmin edersiniz. Fakat endişe etmeyin, söylediğim gibi esaslarını böleceğim ve adresin neye benzediğini, onu nasıl yazabileceğinizi ve yaygın kullanışlarını göstereceğim. İlk olarak biraz garip gelebilir, fakat öğrendikten sonra, onu iyice kavrayacaksınız.

Şimdi, bölümlerine ayrılmış, örnek bir IPv6 adresi gösteren Şekil 13.1'e bakalım.




***Şekil 13.1:*** *IPv6 adres örneği.*

*Router'larımızı yapılandırdığımızda, bu bölümde daha sonra "Ağ Topluluğumuzda, IPv6'yı Yapılandırmak" bölümünde onu kullanacağımızdan, bu subnet ID'yi nerede bulabileceğinizi hatırlayın.*

**NOT**

Şimdi görebileceğiniz gibi adres tamamıyla daha büyüktür, fakat başka ne fark vardır? İlk olarak, dört yerine sekiz adres grubuna sahip olduğuna ve bu grupların, nokta yerine iki nokta üst üste ile ayrıldıklarına dikkat edin. Hey bir dakika durun, bu adreste harflerde mi var? Evet, adres, aynı MAC adresinde olduğu gibi, hexadecimal formatta gösterilmektedir. Bu nedenle, bu adresin sekiz 16-bit hexadecimal iki noktayla ayrılmış bloklarının olduğunu söyleyebilirsiniz. Çok uzundur ve muhtemelen adresi henüz, yüksek sesle söylemeyi denememişsinizdir!

Size işaret etmek istediğim diğer bir konu, test ağınızı, IPv6 ile kurduğunuzda, bir IPv6 cihaza HTTP bağlantısı kurmak için bir web browser kullanırsanız, adresi browser'a, tamamını köşeli parantez içinde yazmanız gerekmektedir. Niçin? İki nokta üst üste, zaten port numarasını belirtmek için browser tarafında kullanılmaktadır. Adresi köşeli parantezle kullanmazsanız, browser isteği doğru teşhis edemeyecektir.

Bunun nasıl göründüğüyle ilgili örnek şöyledir:

```
http://[2001:0db8:3c4d:0012:0000:0000:1234:56ab]/default.html
```

Şayet yapabilseniz, bir hedefi belirtmek için, isimleri tercih edersiniz. (`www.lammle.com` gibi). Kesinlikle bir sorun olacak gibi görünmesine rağmen, bazen zor şartlara dayanmak ve adresi yazmamız gerektiğini kabul etmemiz gerekmektedir. Öyleyse, IPv6 çalıştığımızda, DNS'in çok önemli olacağı çok aşikardır.

## Kısaltılmış İfade

Bu uzun adresi yazdığımızda bize yardımcı olacak bazı incelikler vardır. Bir tanesi, özetlemek için adresin bölümlerini atlayabilmenizdir. Fakat bunu yapmak için bazı kurallara uymak zorundasınız. İlk olarak, ayrılmış blokların her birinde öndeki sıfırları atabilirsiniz. Bunu yaptıktan sonra, önceki örnekteki adres şöyle görünür:

```
2001:db8:3c4d:12:0:0:1234:56ab
```

Bu iyi bir gelişmedir. En azından, fazladan bu sıfırları yazmak zorunda değiliz. Peki, içinde sıfır haricinde bir şey olmayan bloğun tamamı ne olacak? Onların, en azından bir kısmını yok edebiliriz. Tekrar örneğimize bakarsak, onların yerine iki tane iki nokta üst üste yazarak, iki sıfır bloğunu atabiliriz. Adres şimdi şöyle olacaktır:

```
2001:db8:3c4d:12::1234:56ab
```

Çok güzel! Tamamı sıfır olan blokların yerine iki adet iki nokta üst üste yazdık. Bunun için izlemek zorunda olduğunuz kural, bir adreste sadece bitişik bir sıfır bloğunu yerleştirebileceğinizdir. Yani, adresim, dört sıfır bloğuna sahipse ve onların hepsi ayrıldıysa, onların tamamını yerleştiremem. Sadece bitişik bir blok yerine, iki nokta üst üste koyabileceğiniz kuralını hatırlayın. Şu örneği inceleyin:

```
2001:0000:0000:0012:0000:0000:1234:56ab
```

Ve şunu yapamayacağınızı bilin:

```
2001::12::1234:56ab
```

Onun yerine, yapabileceğinizin en iyisi şudur:

```
2001::12:0:0:1234:56ab
```

Yukarıdaki örneğin en iyi olmasının sebebi; diğer örnekte iki sıfır bloğunu atarsak, adrese bakan cihazın, sıfırların geri nereye konacağını bilme şansının olmamasıdır. Aslında router, yanlış adrese bakacak ve "iki bloğu, ilk çift iki nokta üst üste setine yerleştireyim mi veya üç bloğu, ilk sete ve bir bloğu ikinci sete yerleştireyim mi?" diyecektir. Ve router'ın ihtiyacı olan bilgi orada olmadığından, o sürekli devam edecektir.

## Adres Çeşitleri

Kimin veya en az kaç cihazın konuştuğunu belirten, IPv4'ün unicast, broadcast ve multicast adreslerine hepimiz aşinayız. Fakat belirttiğim gibi, IPv6 bu üçlüye anycast'i ekler. Broadcast'ler, bildiğimiz gibi, sıkıcı verimsizliklerinden dolayı, IPv6'dan çıkartılmışlardır.

Üç adresleme çeşidi ve haberleşme yönteminin ne olduğunu anlayalım.

**Unicast:** Unicast için adreslenen paketler, tek bir interface'e teslim edilir. Yük dengelemesi için çoklu interface'ler aynı adresi kullanır. Birkaç farklı interface çeşidi vardır. Fakat onları burada anlatmaya gerek yoktur.

**Global unicast adresler:** Bunlar sizin tipik genel olarak yönlendirilebilir adreslerinizdir ve IPv4'dekilerle aynıdır.

**Link-local adresler:** Route edilemedikleri anlamında, IPv4'deki özel adresler gibidir. Onları, route edilmeyecek, fakat hala lokal olarak dosya ve servisleri paylaşması ve erişmesi gereken küçük bir LAN oluşturmak veya toplantılar için geçici bir LAN gönderme kabiliyeti sağlayan faydalı bir araç alarak düşünün.

**Unique local adresler:** Bu adresler, routing olmayan amaçlar için tasarlanmıştır, fakat onlar nerdeyse global olarak benzersizdir. Bu nedenle, onların çakışması olasılık dışıdır. Uniqe local adresler, site-local adreslerinin yerine konmak için tasarlanmışlardır. Bu nedenle, nerdeyse çoklu lokal ağlarda route edilebilirken, bir site boyunca iletişime izin veren IPv4 özel adreslerinin yaptıklarını yaparlar. Site-local adreslerine, Eylül 2004 gibi itiraz edildi.

**Multicast:** IPv4'de olduğu gibi bir multicast adrese gönderilen paketler, multicast adres tarafından tespit edilen tüm interface'lere ulaştırılır. Bazen insanlar onları, one-to-many adres olarak belirtir. Daima FF ile başladıklarından, IPv6'daki multicast adreslerini fark etmek gerçekten kolaydır. Multicast işlemiyle ilgili detaya, "Bir Ağ Topluluğunda IPv6 Nasıl Çalışır?" bölümünde gireceğim.

**Anycast:** Multicast adresleri gibi bir anycast adresi, çoklu interface'leri tanımlar. Fakat büyük bir farklılık vardır: Bir anycast paketi, tek bir adrese teslim edilir (aslında, routing uzaklığıyla tanımlı, bulduğu ilk adrese). Tek bir adresi birden fazla interface'e atayabileceğiniz için bu adres özeldir. Onları, one-to-one-of-many adresler olarak belirtebilirsiniz, fakat kolay olması için sadece anycast olarak belirtin.

IPv4'de olduğunu bildiğinizden, IPv6'da, belirli kullanım için rezerve edilmiş başka özel adres olup olmadığını merak ediyorsunuzdur. Evet, onlardan çok vardır. Şimdi onları inceleyelim.

## Özel Adresler

Çok sık kullanacağınızdan, hatırlamanız için kesinlikle işaretlemeniz gereken bazı adres ve adres aralıklarının listesini vereceğim. Onların tamamı, özel kullanım için rezerve edilmiştirler veya özeldirler. Fakat IPv4'ün tersine, IPv6 bize büyük bir adres zenginliği sağlar. Bu nedenle burada birkaçını rezerve etmenin kimseye zararı olmaz.

**0:0:0:0:0:0:0:0:** Eşittir :: Bu IPv4'ün 0.0.0.0'ına eşdeğerdir ve stateful bir konfigürasyon kullandığınızda, tipik olarak, bir host'un kaynak adresidir.

**0:0:0:0:0:0:0:1:** Eşittir ::1. IPv4'deki 127.0.0.1'in eşdeğeridir.

**0:0:0:0:0:0:192.168.100.1:** Bu, bir IPv4 adresinin, karışık IPv6/IPv4 ağ ortamında yazılma şeklidir.

**2000::/3:** Global unicast adres aralığı.

**FC00::/7:** Uniqe local unicast aralığı.

**FE80::/10:** Link-local unicast aralığı.

**FF00::/8:** Multicast aralığı.

**3FFF:FFFF::/32:** Örnek ve dökümantasyon için rezerve edilmiştir.

**2001:0DB8::/32:** Bu da örnek ve dökümantasyon için rezerve edilmiştir.

**2002::/16:** Geçiş sistemi olan, 6to4 ile kullanılır. Belirtilmiş tünellere gerek duymadan, bir IPv4 ağı boyunca, IPv6 paketlerine izin veren yapıdır.

"IPv6'ya Geçiş" bölümünde bunu daha detaylı işleyeceğiz. Şimdi IPv6'nin bir ağ topluluğunda nasıl çalıştığını göstermeme izin verin. IPv4'ün nasıl çalıştığını tamamıyla biliyoruz. Bakalım yeni neler vardır.

# Bir Ağ topluluğunda IPv6 Nasıl Çalışır?

IPv6'nın iyi noktalarını keşfetme zamanı geldi. Bir host'a nasıl adres verileceğini ve bir ağdaki kaynak ve diğer host'ları bulma kabiliyetini göstererek başlamak için güzel olacaktır.

Cihazların, bazen stateless autoconfiguration olarak belirtilen, kendilerini otomatik olarak adresleme kabiliyetini de göstereceğim. Diğer bir autoconfiguration çeşidi, stateful'dur. Stateful autoconfiguration'ın, IPv4'ün kullandığıyla çok benzer şekilde, bir DHCP kullandığını unutmayın. Ayrıca, IPv6'da, Internet Control Message Protocol (ICMP) ve multicast'in nasıl çalıştığını da göstereceğim.

## Autoconfiguration

Autoconfiguration, çok faydalı bir çözümdür. Çünkü bir ağdaki cihazların kendilerini, link-local unicast adresle adreslemesine izin verir. Bu proseste ilk olarak, router'dan prefix bilgisi öğrenilir ve sonra cihazın kendi interface adresi, interface ID'si olarak eklenir. Peki, bu interface ID'sini nereden alır? Bir Ethernet ağındaki her cihazın, fiziksel bir MAC adrese sahip olduğunu ve bunların, interface ID için kullanıldığını biliyorsunuz. Fakat IPv6'daki interface ID'nin 64 bit ve MAC adresinin, sadece 48 bit olmasından dolayı, ilave 16 bit nereden gelmektedir? MAC adresine, ilave bit'ler ortaya takviye edilir. O, FFFE ile takviye edilir.

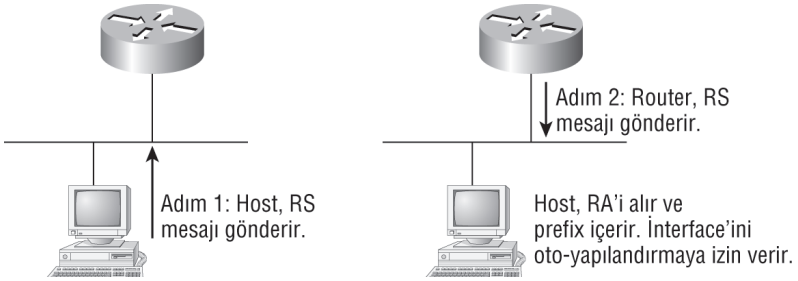
Örnek olarak, 0060.d673.1987 şeklinde bir MAC adresi olan cihazımız olsun. Takviye edildikten sonra, şöyle görünecektir: 0260.d6FF.FE73.1987.

Peki, adresin başındaki 2 nereden geldi? Başka bir iyi soru. Takviye işlemini (değiştirilmiş eui-64 format olarak belirtilir), adresin, lokal veya global olarak benzersiz olduğunu belirtmek için, bir bit'i değiştirir. Ve değişen bit, adresteki yedinci bit'tir. 1 değerindeki bit'in anlamı, global olarak benzersiz, 0'ın anlamı da lokal olarak benzersizdir. Örneğimize bakarak, bunun global veya lokal olarak benzersiz olduğunu söyleyebilir misiniz? Şayet global olarak benzersiz olarak cevaplarsanız, doğrudur. Bu, host makinelerinizin adreslenmesinde size zaman kazandıracaktır. Çünkü bunu yapmak için router'la haberleşirler.

Autoconfiguration çalıştırmak için bir host, basit iki adımlı bir işlemden geçer:

1. İlk olarak, interface'ini yapılandırması için host'un, (bir IPv4 adresinin network bölümüne benzer) prefix bilgisine ihtiyaç vardır. Bu nedenle, o, bir router solicitation (RS) isteği gönderir. Bu RS, daha sonra her router'ın multicast adresine, multicast olarak gönderilir. Gönderilen asıl bilgi, bir ICMP mesaj tipidir ve ağdaki her şey gibi, bu ICMP mesajı, onu tanımlayan bir numaraya sahiptir. RS mesajı, ICMP type 133'dür.
2. Router, bir router advertisement (RA) yardımıyla, cevap olarak gerekli prefix bilgisiyle geri döner. Bir RA mesajı ayrıca, her düğümün multicast adresine gönderilen bir multicast paketi olmaktadır ve ICMP type 134'dür. RA mesajları, periyodik olarak gönderilirler, fakat host, RS'i, acil bir cevap için gönderir. Bu yüzden, ihtiyacı olanı almak için, sonraki planlanmış RA'e kadar beklemek zorunda değildir.

Bu iki adım, Şekil 13.2'de gösterilmektedir.




***Şekil 13.2:*** *IPv6 autoconfiguration için iki adım.*

Bu autoconfiguration tipi ayrıca stateless autoconfiguration olarak bilinir. Çünkü daha fazla bilgi almak için diğer cihazlarla bağlantıya geçmez veya bağlanmaz. Birazdan DHCPv6'dan bahsederken, stateful autoconfiguration kullanacağız.

Cisco router'ların, IPv6 ile nasıl yapılandırıldığına bir bakalım.

## Cisco Router'ları, IPv6 ile Yapılandırmak

Bir router'da IPv6'yı etkinleştirmek için, `ipv6 unicast-routing` global konfigürasyon komutunu kullanabilirsiniz:

```
Corp(config)#ipv6 unicast-routing
```

Varsayılan olarak, IPv6 trafik forwarding etkin değildir, bu nedenle bu komutu kullanmak için onu etkinleştirin. Ayrıca, muhtemelen tahmin edebileceğiniz gibi IPv6'da varsayılan olarak interface'ler de etkin değildir. Bu nedenle her interface'e teker teker gidip onları aktif hale getirmemiz gerekmektedir.

Bunu yapmanın çeşitli yolları vardır, fakat en basit yolu, interface'e bir adres eklemektir. Bunu yapmak için `ipv6 address <ipv6prefix>/<prefix-length> [eui-64]` interface konfigürasyon komutunu kullanırsınız.

İşte bir örnek:

```
Corp(config-if)#ipv6 address 2001:db8:3c4d:1:0260.d6FF.FE73.1987/64
```

128-bit global IPv6 adresini belirtebilir veya `eui-64` seçeneğini kullanabilirsiniz. Eui-64 formatının, cihazın MAC adresini kullanmasına ve interface ID'si yapmak için ona takviye yapmasına izin verdiğini hatırlayın. Şunu kontrol edin:

```
Corp(config-if)#ipv6 address 2001:db8:3c4d:1::/64 eui-64
```

Sadece link-local adresler kullanması için bir router'ı yapılandırmakta `ipv6 enable interface` konfigürasyon komutunu kullanın:

*Sadece bir link-local adresine sahipseniz, sadece bu lokal subnet ile haberleşebileceğinizi hatırlayın.*

NOT

```
Corp(config-if)#ipv6 enable
```

Şimdi IPv6 kullanımı için bir DHCP yapılandırarak, stateful IPv6'ya derinlemesine girelim.

## DHCPv6

DHCPv6, IPv6 için desteklediği yeni adresleme sistemi farklılığıyla, v4'de DHCP'nin çalıştığına oldukça benzer şekilde çalışır. Ve belki şaşırtıcı gelebilir, fakat autoconfiguration'ın sağlamayıp, DHCP'nin hala sağladığı bazı seçenekler vardır. Ciddi söylüyorum, örneğin IPv4 yoluyla DHCP'nin bize daima sağladığı DNS sunucusu, domain ismi veya diğer seçeneklerin çoğundan kesinlikle bahsedilmemektedir. Bu, çoğu zaman, IPv6'da hala DHCP kullanıyor olmamızın gerçek sebebidir.

IPv4'de başladığında bir istemci, gerekli olan bilgileri vermesi amacıyla bir sunucu bulmak için bir DHCP discover mesajı gönderir. Fakat IPv6'da, ilk olarak RS ve RA işlemlerinin olduğunu hatırlayın. Şayet ağda bir DHCPv6 sunucusu varsa, istemciye geri gelen RA, DHCP'nin kullanım için uygun olup olmadığını söyleyecektir. Şayet router bulunamazsa istemci, bir DHCP solicit mesajı ile cevaplandırılacaktır. Bir solicit mesajı, tüm server ve relay DHCP agent'ları anlamına gelen ff02::1:2 kaynak adresine sahip bir multicast mesajıdır.

Cisco IOS'da, DHCPv6 için desteğin olduğunu bilmek güzeldir. Fakat bu, stateless DHCP sunucusu ile sınırlıdır. Yani, herhangi bir adres havuz yönetimi sunmaz, artı bu adres yapılandırmaları için uygun seçenekler, DNS, domain ismi ve sadece SIP server ile sınırlıdır.

Yani, hem ilave gerekli bilgileri sağlayıp dağıtacak hem de adres atamalarını yönetecek başka bazı sunuculara kesinlikle ihtiyacınız olacaktır.

Burada, konfigürasyonun, stateless DHCP'yi nasıl aradığını görebilirsiniz. IPv4 ile yapılandırdığımızda yaptığımıza oldukça yakındır:

```
Router1(config)#ipv6 dhcp pool ?
WORD  DHCP pool name
Router1(config)#ipv6 dhcp pool test
Router1(config-dhcp)#?
IPv6 DHCP configuration commands:
  default           Set a command to its defaults
  dns-server        DNS servers
  domain-name       Domain name to complete unqualified host names
  exit              Exit from DHCPv6 configuration mode
  no                Negate a command or set its defaults
  prefix-delegation IPv6 prefix delegation
  sip               SIP Servers options
Router1(config-dhcp)#dns-server ?
  Hostname or X:X:X:X::X  Server's name or IPv6 address
Router1(config-dhcp)#domain-name lammle.com
Router1(config-dhcp)#prefix-delegation ?
  X:X:X:X::X/<0-128>  IPv6  x:x::y/<z>
  aaa                 Acquire prefix from AAA
```

```
  pool                IPv6 prefix pool
Router1(config-dhcp)#prefix-delegation pool ?
  WORD  IPv6 prefix pool
Router1(config-dhcp)#prefix-delegation pool test ?
  lifetime  Configure prefix lifetimes
  <cr>
Router1(config-dhcp)#prefix-delegation pool test lifetime ?
  <60-4294967295>  Valid lifetime (seconds)
  at               Expire prefix at a specific time/date
  infinite         Infinite valid lifetime
Router1(config-dhcp)#prefix-delegation pool test lifetime 3600 ?
  <60-4294967295>  Preferred lifetime (seconds)
  infinite         Infinite preferred lifetime
Router1(config-dhcp)#prefix-delegation pool test lifetime 3600
3600 ?
  <cr>
Router1(config-dhcp)#prefix-delegation pool test lifetime 3600
3600
```

IPv4 ile DHCP'deki gibi, bir lifetime ayarlamanıza gerek olmadığına dikkat edin.

Şimdi yapılandırılmış bir havuza sahibim. IPv4'ten hareketle onu bir interface'e atamam gerekmektedir:

```
Router1(config)#int fa 0/0
Router1(config-if)#ipv6 dhcp server ?
  WORD  Name of IPv6 DHCP pool
Router1(config-if)#ipv6 dhcp server test
```

Şimdi, fa0/0 interface'imize uygulanmış, tamamıyla yapılandırılmış bir DHCPv6 sunucumuz vardır.

## ICMPv6

IPv4, hedef ulaşılamaz gibi hata mesajları ve Ping ve Traceroute gibi hata tespiti fonksiyonları gibi birçok şey için ICMP kullandı. ICMPv6 hala bunları yapmaktadır. Fakat öncekilerin tersine, v6, ayrı bir katman4 protokolü olarak uygulanmaz. O, IPv6'nın tümleşik bir parçasıdır ve uzantı bir başlık olarak, temel IPv6 başlık bilgisinden sonra taşınmaktadır. Ve ICMPv6 başka bir güzel özellik getirir. Path MTU discovery denilen bir ICMPv6 işlemiyle, IPv6'nın herhangi bir fragmentasyon yapmasını engeller.

Bu şöyle çalışır: Bir bağlantının kaynak düğümü, yerel linkin MTU boyutuna eşit bir paket gönderecektir. Bu paket, hedefine doğru giderken, mevcut paketin boyutundan daha küçük bir MTU'ya sahip herhangi bir link, aradaki router'ın, kaynak makineye "paket çok büyük" mesajı göndermesi için zorlayacaktır. Bu mesaj, kaynak düğüme, kısıtlayıcı linkin kabul edeceği maksimum boyutun ne olacağını ve kaynaktan, geçebilecek yeni bir paket göndermesini istediğini söylemektedir. Bu proses, kaynak düğümün yeni yolun MTU'sunu kullanmasıyla, sonunda hedefe erişinceye kadar devam eder. Böylece, veri paketinin kalanı aktarıldığında, onlar parçalanmaktan korunacaktır.

ICMPv6 şimdi, yerel linkteki diğer cihazların adreslerini bulma görevinden sorumludur. Address Resolution Protocol, IPv4 için bu fonksiyonu çalıştırmak için kullanıldı. Fakat ICMPv6'da ismi Ne-

ighbor Discovery olarak değiştirildi. Bu proses, solicited node adresi denilen bir multicast adresi kullanarak tamamlanır ve ağa bağlandıklarında, tüm host'lar bu multicast gruba katılır. Onların IPv6 bölümleri (sağdan 24bit uzaklıkta olan), FF02:0:0:0:0:1:FF/104 multicast adresinin sonuna eklenmektedir. Bu adres sorgulandığında, ilgili host, onun katman2 adresine gönderilecektir. Cihazlar, ağdaki diğer komşu cihazları, nerdeyse aynı yolla bulup, izleyebileceklerdir. Daha önce RA ve RS mesajlarından bahsettiğimde ve onların, adres bilgilerini istemek ve almak için, multicast trafiği kullandıklarını söylediğimde, özellikle neighbor discovery'nin, ICMPv6'nın bir fonksiyonu olduğunu söylemiştim.

IPv4'de IGMP protokolü, host cihazın yerel router'a bir multicast gruba katıldığını ve bu grup için trafiği almak istediğini söylemesine izin verir. Bu IGMP fonksiyonu, ICMPv6 tarafından değiştirildi ve prosesin adı, multicast listener discovery olarak değiştirildi.

# IPv6 Routing Protokolü

Daha önce işlediğimiz routing protokollerinin çoğu, IPv6 ağlarda kullanmak için upgrade edilmektedir. Ayrıca öğrendiğimiz fonksiyon ve yapılandırmaların çoğu, şimdikilerle nerdeyse aynı şekilde kullanılacaklardır. Broadcast'in, IPv6'dan çıkartıldığını bilin. Tamamıyla broadcast trafiği kullanan bir protokol, nesli tükenmiş kelaynak kuşuna benzeyecektir. Fakat kelaynakın tersine, bu bant genişliği-oburu, performans yok edici küçük cine hoşça kal demek güzel olacaktır.

V6'da hala kullanacağımız routing protokolleri, yeni bir isme ve görünüşe sahip olacaktır. Şimdi onların bazılarından bahsedelim.

Listenin başındaki, RIPng (next generation)'dir. Bir süredir IT sektöründe olanlarınızın RIP olarak bildiği bu protokol, küçük ağlarda oldukça iyi çalışır. Tükenmemesi ve hala IPv6'da kullanılmasının birçok sebebi vardır. Ve hala EIGRPv6'ya sahibizdir. Çünkü protokol-bağımsız modüllere sahipti ve tüm yapmamız gereken, IPv6 için yeni bir modül eklememizdir. OSPFv3, bir yazım hatası değildir, gerçekten v3'tür. IPv4 için OSPF, v2'di. IPv6 için upgrade olunca, OSPFv3 oldu.

## RIPng

Dürüst olmak gerekirse, RIPng'nin temel özelliği, RIPv2 ile aynıdır. O hala distance-vector bir protokoldür. Maksimum 15 hop sayısına sahiptir ve split horizon, poison reverse ve diğer döngü engelleme mekanizmaları kullanır. Fakat şimdi, UDP port 521'i kullanıyor.

Güncellemelerini göndermek için de hala multicast kullanır. Fakat IPv6'da, transport adres için FF02::9 kullanır. Bu, RIPv2'de multicast adresi, 224.0.0.9 olduğundan, gerçekten güzel bir özelliktir. Yeni IPv6 multicast aralığında, adresin sonunda yine 9 vardır. Aslında, çoğu routing protokolü, bu gibi IPv4 özelliklerinin bir kısmını korumaktadır.

Fakat tabi ki yeni versiyonda farklılıklar vardır. Router'ların, router tablolarındaki her hedef ağ için komşu router'larının next-hop adreslerini tuttuklarını biliyoruz. Farklılık; RIPng ile router, bu next-hop adresleri bir global adresle değil de link-local adres kullanarak izler.

RIPng (ve tüm IPv6 routing protokollerinin bu durum için) ile değişikliklerin en büyüğü muhtemelen, ağ yayınlarını, router configuration modda `network` komutu yerine, interface configuration moddan bir `network` komutu ile aktif kılmak veya ayarlamaktır. Eğer router configuration moda gitmeden, onu bir interface'de direkt olarak aktif kılarsanız ve bir RIPng prosesi başlatırsanız, yeni bir RIPng prosesi sizin için başlayacaktır. Bu şöyle olacaktır:

```
Router1(config-if)#ipv6 rip 1 enable
```

Bu komutta gördüğünüz 1, çalışan RIPng prosesini tanımlayan bir etikettir ve söylediğim gibi, bu bir RIPng prosesini başlatacaktır, böylece router configuration moda girmeniz gerekmeyecektir.

Fakat şayet, router configuration modda, redistribution gibi başka bir şeyi yapılandırmak için girerseniz, bunu hala yapabilirsiniz. Bunu yaparsanız, router'ınızda şöyle görünecektir:

```
Router1(config)#ipv6 router rip 1
Router1(config-rtr)#
```

Sadece RIPng'nin, bağlı ağlara route için interface'i aktif etmek için kullandığınız `network` komutu kullanması yerine network'ün kendisini kullanması farklılığıyla, IPv4'le neredeyse aynı çalıştığını hatırlayın.

## EIGRPv6

RIPng'de olduğu gibi EIGRPv6, önceki IP4 versiyonu ile nerdeyse aynı çalışır. EIGRPv6'dan önce EIGRP'nin sağladığı özelliklerin çoğu hala geçerli olacaktır.

EIGRPv6, aynı link-state özelliklerine sahip, gelişmiş bir distance-vector protokolüdür. Hello'lar kullanan komşu bulma prosesi hala vardır ve bize, Diffusing Update Algorithm (DUAL) kullanan, döngü olmayan hızlı convergence sağlayan güvenli transport protokol ile güvenli haberleşme sağlayacaktır.

Hello paketleri ve güncellemeler, multicast aktarım kullanarak gönderilir ve RIPng ile EIGRPv6'nın multicast adresi, neredeyse aynı kaldı. IPv4'de, 224.0.0.10'du, IPv6'da FF02::A (A, hexadecimal gösterimde 10'a eşittir).

Fakat iki versiyon arasında farklılıklar vardır. Özellikle, RIPng'deki gibi, `network` komutu kullanımı yoktur ve yayınlanacak network ve interface, interface configuration moddan aktif hale getirilmelidir. EIGRPv6'da, routing protokolünü etkin kılmak için, hala router configuration modu kullanmanız gerekmektedir. Çünkü routing prosesinin, bir interface'de olduğu gibi, `no shutdown` komutu ile aktif hale getirilmesi gerekmektedir.

EIGRPv6 için yapılandırma şöyle görünecektir:

```
Router1(config)#ipv6 router eigrp 10
```

Bu örnekteki 10, hala autonomous system (AS) numarasıdır. İstemci `(config-rtr)` olarak değişmiştir ve buradan, `no shutdown` çalıştırmalısınız:

```
Router1(config-rtr)#no shutdown
```

Redistribution gibi diğer seçenekler de bu moddan ayarlanabilir.

Şimdi, interface'e gidelim ve IPv6'yı aktif hale getirelim:

```
Router1(config-if)#ipv6 eigrp 10
```

`Interface` komutundaki 10 yine, configuration modda etkinleştirilen AS numarasını işaret eder.

## OSPFv3

OSPF'in yeni versiyonu, routing protokollerinin, IPv4 versiyonlarının birçok benzerliğine sahip olma eğilimini devam ettirmektedir.

OSPF altyapısı aynı kalmaktadır. O hala, tüm ağ topluluğunu veya autonomous system'i area'lara bölerek hiyerarşi oluşturan, bir link-state routing protokolüdür. Multi-area OSPF, en azından şimdilik, CCNA konularının kapsamı dışındadır. Fakat bölüm 7'de değindiğimiz bazı seçenekler, biraz farklılık gösterecektir.

OSPFv2'de, routerID (RID), router'a atanmış (ya da onu siz atayabilirsiniz) en yüksek IP adresi ile belirlenmektedir. Versiyon3'de siz, area ID, RID ve link-state ID atarsınız. Bunların hepsi 32-bit'tir, IPv6 adreslerin, 128 bit olmasından dolayı, artık IP adresi kullanarak bulunamazlar. OSPF

paket başlıklarından IP adres bilgilerinin silinmesiyle birlikte, bu değerlerin nasıl atanacağıyla ilgili değişiklikler, OSPF'in yeni versiyonunu, neredeyse Network katman protokolü boyunca route edilir hale getirmiştir.

Adjacency ve next-hop attribute'ları şimdi, link-local adresleri kullanırlar ve OSPFv3, güncelleme ve acknowledgment'larını göndermek için hala multicast trafik kullanır. (OSPF router'lar için FF02::5 ve OSPF-designated router'lar için FF02::6). Bu yeni adresler, sırasıyla 224.0.0.5 ve 224.0.0.6'nin yerine kullanılmaktadır.

Diğer daha az esnek IPv4 protokolleri, OSPFv2'nin sağladığı, OSPF'e belirli network ve interface'lerin atanması kabiliyetini sağlamaz. Bununla beraber, hala router configuration mod altında yapılandırılır. OSPFv3 ile bahsettiğimiz diğer IPv6 routing protokollerinde olduğu gibi interface ve onlara bağlı network'ler, interface configuration modda, interface'de direkt olarak yapılandırılır.

OSPFv3 yapılandırması, şöyle olacaktır:

```
Router1(config)#ipv6 router osfp 10
Router1(config-rtr)#router-id 1.1.1.1
```

Summarization ve redistribution gibi bazı yapılandırmaları, router configuration moddan çalıştırmalısınız. Fakat OSPFv3'ü, interface'den yapılandırıyorsak, OSPFv3'ü yapılandırmaya bile gerek yoktur.

Interaface yapılandırılması tamamlandığında, router konfigürasyon prosesi otomatik olarak eklenir ve interface yapılandırması şöyle görünür:

```
Router1(config-if)#ipv6 ospf 10 area 0.0.0.0
```

Böylece, her interface gidip bir process ID ve area atarsak, tamamlamış oluruz.

Şimdi, IPv4'den IPv6'ya nasıl geçiş yapılacağına geçme ve onu öğrenme zamanı geldi.

# IPv6'ya Geçiş

IPv6'nın nasıl çalıştığı ve ağlarımızda çalışması için onu nasıl yapılandırabileceğimiz konusunda çok şey konuştuk. Fakat bunu yapmanın bize maliyeti ne olacaktır? Ve bu gerçekte bizlere ne kadar çalışma yükü gerektirecektir? Emin olmak için iyi sorular, fakat cevapları herkes için aynı olmayacaktır. Aslında cevap kendinize nasıl bir bütçe desteği bulacağınız, nasıl bir altyapıya sahip olduğunuzla ilgilidir. Açıkça, eski router ve switch'lerinizi kaldırıp ,onların hepsini, IPv6 uyumlu olacak şekilde upgrade etmeniz gerekiyorsa, bu oldukça fazla miktarda paranız olmasıyla mümkündür. Bu, sunucu ve bilgisayar işletim sistemlerini içermemektedir. Tabi ki biraz onların da masrafı da olacaktır. Güzel haber, bir şeylerin atılmasına izin vermeksizin, uzun yıllardır birçok işletim sistemi ve ağ cihazlarının IPv6 uyumlu olmasıdır. Şu ana kadar bu özelliklerin tamamını kullanmıyorduk.

Şimdi, çalışma yükü ve zaman ile ilgili diğer soruya gelince. Bu oldukça zaman alıcı ve yoğun olabilir. Tüm sisteminizi elden geçirmeniz ve her şeyin düzgün çalıştığına emin olmanız zaman alacaktır. Bir de, çok sayıda cihazın olduğu büyük bir ağdan bahsediyorsanız, gerçekten uzun bir zaman alacaktır. Fakat paniğe kapılmayın, geçiş stratejileri, uzun süreli bir uyumu kabul edecek şekilde oluşturulmuştur. Bizim için uygun geçiş stratejilerinden en önemli üçünü göstereceğim. İlki, bir cihazın, mevcut haberleşmelerin ve yeni IPv6 iletişimlerin eşzamanlı çalışmasını devam ettirme kabiliyetine sahip olacak şekilde, hem IPv4 hem de IPv6 protokol yığını çalışmasına izin veren, dual stacking olarak bilinen yöntemdir. Diğer strateji, 6to4 tunneling yaklaşımıdır. Bu, uzaktaki diğer IPv6 ağlarına, IPv4 bir ağ üzerinden erişen, IPv6 ağlarınız için kullanılan seçimdir. Üçüncüsü, sadece eğlence için, size sürpriz olacak.

## Dual Stacking

Bizim için en kolay yol olduğundan, en yaygın geçiş stratejisidir. Cihazlarımızın hem IPv4 hem de IPv6 kullanarak haberleşmelerine izin verir. Dual stacking, ağdaki cihaz ve uygulamalarınızı, istediğiniz zaman upgrade etmenize izin verir. Ağdaki host ve cihazlar zamanla upgrade edildikçe, haberleşmeleriniz IPv6 üzerinden olacaktır ve sonunda her şey IPv6 çalışacaktır. Sonunda ihtiyacınız olmayan eski IPv4 protokol yığınlarının hepsini atarsınız.

Ayrıca bir Cisco cihazda dual stacking yapılandırmak inanılmaz kolaydır. Tüm yapmanız gereken, IPv6 forwarding'i aktif hale getirmek ve zaten IPv4 ile yapılandırılmış interface'e, bir adres atamaktır. O da şöyle olacaktır:

```
Corp(config)#ipv6 unicast-routing
Corp(config)#interface fastethernet 0/0
Corp(config-if)#ipv6 address 2001:db8:3c4d:1::/64 eui-64
Corp(config-if)#ip address 192.168.255.1 255.255.255.0
```

Fakat dürüst olmak gerekirse, muhtemelen kısa bir zaman alacağından, tek bir protokol olarak IPv6 çalışmaya başlamadan önce, farklı tunelling tekniklerini anlamak gerçekten iyi bir fikirdir.

# 6to4 Tunneling

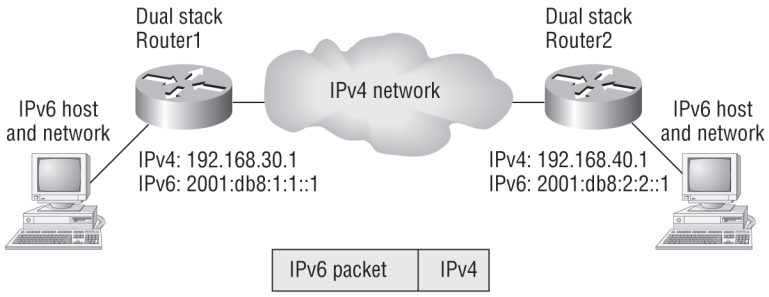
6to4 tunelling, IPv6 veriyi hala IPv4 çalışan bir ağ üzerinden taşımak için gerçekten kullanışlıdır. IPv6 subnetlere veya diğer uçları tamamıyla IPv6 olan ağlara sahip olmanız ve bu ağların birbirleriyle mutlaka haberleşmesi gerekeceği durumlar olacaktır. Çok karmaşık değil, fakat bunun bir WAN veya kontrol edemeyeceğiniz başka bir ağ üzerinden olduğunu düşündüğünüzde, bu biraz kötü olabilir. Peki, prosesin tamamını kontrol edememek ile ilgili ne yaparız? IPv6 trafiği, IPv4 network üzerinden taşıyacak bir tünel oluşturun. Yapılacak olan budur.

Tunelling kavramı, zor anlaşılır değildir ve tüneller oluşturmak, aslında düşündüğünüz kadar zor değildir. Olay aslında, ağda mutlu şekilde dolaşan IPv6 paketinin yakalanması ve onun önüne bir IPv4 başlığı konmasıdır. Akıntıya tekrar bırakılmadan önce, balığın yüzüne yara bandı yapıştırmak haricinde, bir nevi, balığın yakalanması ve salınmasına benziyor.

Bunun nasıl olduğunu, Şekil 13.3'e bakarak görebilirsiniz.

Bunu yapmak için, birkaç adet dual-stacked router'a ihtiyacımız olacak. Şimdi bu router'lar arasında bir tünel yerleştirmek için küçük bir konfigürasyon eklememiz gerekmektedir. Her router'a, tünelin nerede başladığını ve onun nerede sonlanmasını istediğimizi söylememiz gerekiyor. Şekil 13.3'ü referans alarak, her router'da tünel yapılandıracağım:

```
Router1(config)#int tunnel 0
Router1(config-if)#ipv6 address 2001:db8:1:1::1/64
Router1(config-if)#tunnel source 192.168.30.1
Router1(config-if)#tunnel destination 192.168.40.1
Router1(config-if)#tunnel mode ipv6ip

Router2(config)#int tunnel 0
Router2(config-if)#ipv6 address 2001:db8:2:2::1/64
Router2(config-if)#tunnel source 192.168.40.1
Router2(config-if)#tunnel destination 192.168.30.1
Router2(config-if)#tunnel mode ipv6ip
```




***Şekil 13.3:*** *Bir 6to4 tünel oluşturmak.*

IPv6 paketi, bir IPv4 paketi içinde enkapsüle ediliyor.

Şimdi IPv6 ağlarımız, IPv4 network üzerinden haberleşebilmektedir. Bunun kalıcı bir yapılandırma olduğu anlamına gelmediğini söylemek zorundayım. Sizin son hedefiniz, komple IPv6 uçtan uca bir ağın çalışması olmalıdır.

Buradaki önemli bir nokta, şayet dolaştığınız IPv4 ağı bir NAT çeviri noktasıysa bizim yeni oluşturduğumuz tünel enkapsülasyonunu kıracaktır. Yıllar geçmesine rağmen NAT, özel protokolleri ve dinamik bağlantıları çalıştırabilecek şekilde birçok defa upgrade edilmiştir. Bu upgrade'lerden biri olmadan, NAT çoğu bağlantıyı yıkıcı olarak görünür ve çoğu NAT uygulamasında bu geçiş stratejisi olmadığından, bu tam bir problemdir.

Bu ufak problemi çözmek amacıyla Teredo denilen bir çözüm vardır. O, tüm tünel trafiğinizin, UDP paketlerine yerleştirilmesine izin verir. NAT, UDP paketlerinde etkili değildir, bu nedenle diğer protokol paketlerinde olduğu gibi parçalanmayacaklardır. Böylece Teredo ile ve paketleriniz UDP örtüleri altında gizleneceği için NAT tarafından kolayca, canlı ve iyi şekilde işlenecektir.

## NAT-PT

Muhtemelen IPv6'nın NAT'a sahip olmadığını duymuşsunuzdur. Ve bu neredeyse doğrudur. Kendi başına IPv6 bir NAT uygulamasına sahip değildir. NAT protocol translation (NAT-PT) olarak bilinen bir çeviri stratejisi olduğundan, bu sadece bir teknik detaydır. Bunun sadece son çare olarak kullanılacak bir yaklaşım olduğunu, iyi bir çözüm olmadığını da bilin. Onunla, IPv4 host'larınız, sadece diğer IPv4 host'larıyla haberleşebilir ve IPv6 host'ları da, diğer IPv6'larıyla görüşürler. Bununla ne demek istiyorum? Bir tunelling yaklaşımıyla, IPv6 paketlerini aldık ve onları IPv4 paketi olarak gizledik. NAT-PT ile enkapsülasyon yoktur. Kaynak paketin verisi, bir IP tipinden çıkartılır ve yeni hedef IP tipiyle tekrar paketlenir. NAT-PT'nin yapılandırılabilmesi, CCNA konularının kapsamı dışında olmasına rağmen, bunu size açıklamak istiyorum. IPv4 için NAT ile olduğu gibi, onu çalıştırmanın birkaç yolu vardır.

Statik NAT-PT, tek bir IPv4 adresinin, tek bir IPv6 adresine eşleştirilmesini sağlar (statik NAT gibi geliyor kulağa). Ayrıca IPv6 adresleriyle bire-bir eşleşme sağlamak için bir IPv4 adres havuzu kullanan, Dinamik NAT-PT de vardır. Son olarak, çoklu IPv6 adresini, tek bir IPv4 adresine ve bir port numarasına many-to-one eşleştirmesi sağlayan, Network Address Port Translation (NAPT-PT) vardır

Görebileceğiniz gibi; IPv4'de yaptığımız gibi genel bir adresi, özel IPv6 adresine çevirmek için NAT'ı kullanmıyoruz. Bunun kesinlikle son çare olarak kullanılması gerekmektedir. Çoğu durumda, bu yapılandırmanın getireceği baş ağrısı ve sistem yükü olmadan, bir tunelling yaklaşımı çok daha iyi çalışacaktır.

# Ağ Topluluğumuzda IPv6 Yapılandırmak

Bu bölümde, bu kitap boyunca kullandığım, birbirine bağlı beş router olan ağ topluluğunu yapılandıracağım. Fakat bazı proseslerin basit ve kolay anlaşılmasını devam ettirmek için 871W router'a veya R1, R2 ve R3 router'lara bağlı LAN ve WLAN ağlarına IPv6 eklemeyeceğim. Şimdi Corp, R1, R2 ve R3 router'larına IPv6 ekleyerek başlayalım. Daha sonra RIP ve OSPF routing protokolleri ekleyeceğim ve bazı doğrulama komutları çalıştırarak bu modülü tamamlayacağım.

Her zamanki gibi, Corp router ile başlayacağım:

```
Corp#config t
Corp(config)#ipv6 unicast-routing
Corp(config)#int f0/1
Corp(config-if)#ipv6 address 2001:db8:3c4d:11::/64 eui-64
Corp(config-if)#int s0/0/0
Corp(config-if)#ipv6 address 2001:db8:3c4d:12::/64 eui-64
Corp(config-if)#int s0/0/1
Corp(config-if)#ipv6 address 2001:db8:3c4d:13::/64 eui-64
Corp(config-if)#int s0/1/0
Corp(config-if)#ipv6 address 2001:db8:3c4d:14::/64 eui-64
Corp(config-if)#int s0/2/0
Corp(config-if)#ipv6 address 2001:db8:3c4d:15::/64 eui-64
Corp(config-if)#^Z
Corp#copy run start
Destination filename [startup-config]?[enter]
Building configuration...
[OK]
Corp#
```

Yukarıdaki yapılandırmada her interface'in subnet adresini kısmen değiştirdim. Routing tablosuna bir göz atalım:

```
Corp#sh ipv6 route
IPv6 Routing Table - 12 entries
Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP
       U - Per-user Static route I1 - ISIS L1, I2 - ISIS L2, IA - ISIS
       interarea, IS - ISIS summary O - OSPF intra, OI - OSPF inter,
       OE1 - OSPF ext 1, OE2 - OSPF ext 2
       ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
C   2001:DB8:3C4D:11::/64 [0/0]
     via ::, FastEthernet0/1
L   2001:DB8:3C4D:11:21A:2FFF:FE55:C9E9/128 [0/0]
     via ::, FastEthernet0/1
C   2001:DB8:3C4D:12::/64 [0/0]
     via ::, Serial0/0/0
L   2001:DB8:3C4D:12:21A:2FFF:FE55:C9E8/128 [0/0]
     via ::, Serial0/0/0
```

```
C   2001:DB8:3C4D:13::/64 [0/0]
     via ::, Serial0/0/1
L   2001:DB8:3C4D:13:21A:2FFF:FE55:C9E8/128 [0/0]
     via ::, Serial0/0/1
C   2001:DB8:3C4D:14::/64 [0/0]
     via ::, Serial0/1/0
L   2001:DB8:3C4D:14:21A:2FFF:FE55:C9E8/128 [0/0]
     via ::, Serial0/1/0
C   2001:DB8:3C4D:15::/64 [0/0]
     via ::, Serial0/2/0
L   2001:DB8:3C4D:15:21A:2FFF:FE55:C9E8/128 [0/0]
     via ::, Serial0/2/0
L   FE80::/10 [0/0]
     via ::, Null0
L   FF00::/8 [0/0]
     via ::, Null0
Corp#
```

Her interface'deki iki adres nedir? Biri, bağlı C'yi, diğeri L'yi göstermektedir. Bağlı adres, her interface'de yapılandırdığım IPv6 adresidir ve L, otomatik olarak atanan linkl-local adresidir. Link-local adresindeki FF:FE'nin, eui-64 adresi oluşturmak için adrese dahil edildiğine dikkat edin.

R1 router'ına geçmeden önce bir şey daha var. İnterface'leri adreslemeden önce, her biri için farklı bir subnet numarası eklediğime dikkat edin. Ayrıca onların neredeyse, IPv4 özel adreslerimle eşleştiğine dikkat edin. Bu yolla, yönetimin kolay olmasını sağladım. Şimdi R1'i yapılandıralım:

```
R1#config t
R1(config)#ipv6 unicast-routing
R1(config)#int s0/0/0
R1(config-if)#ipv6 address 2001:db8:3c4d:12::/64 eui-64
R1(config-if)#int s0/0/1
R1(config-if)#ipv6 address 2001:db8:3c4d:13::/64 eui-64
R1(config-if)#^Z
R1#show ipv6 route
IPv6 Routing Table - 6 entries
[codes cut]
C   2001:DB8:3C4D:12::/64 [0/0]
     via ::, Serial0/0/0
L   2001:DB8:3C4D:12:21A:6DFF:FE64:9B2/128 [0/0]
     via ::, Serial0/0/0
C   2001:DB8:3C4D:13::/64 [0/0]
     via ::, Serial0/0/1
L   2001:DB8:3C4D:13:21A:6DFF:FE64:9B2/128 [0/0]
     via ::, Serial0/0/1
L   FE80::/10 [0/0]
     via ::, Null0
```

```
L   FF00::/8 [0/0]
     via ::, Null0
R1#
```

Linkin her iki tarafında da tamamıyla aynı subnet IPv6 adresini kullandığıma dikkat edin. Gelin R2 ve R3 router'larını yapılandıralım ve IPv6'yı ekleyelim:

```
R2#config t
R2(config)#ipv6 unicast-routing
R2(config)#int s0/2/0
R2(config-if)#ipv6 address 2001:db8:3c4d:14::/64 eui-64
R2(config-if)#do show ipv6 route
IPv6 Routing Table - 4 entries
C   2001:DB8:3C4D:14::/64 [0/0]
     via ::, Serial0/2/0
L   2001:DB8:3C4D:14:213:60FF:FE20:4E4C/128 [0/0]
     via ::, Serial0/2/0
L   FE80::/10 [0/0]
     via ::, Null0
L   FF00::/8 [0/0]
     via ::, Null0
R2(config-if)#
Looking good! Let's go to R3:
R3#config t
R3(config)#ipv6 unicast-routing
R3(config)#int s0/0/1
R3(config-if)#ipv6 address 2001:db8:3c4d:15::/64 eui-64
R3(config-if)#do show ipv6 route
IPv6 Routing Table - 4 entries

C   2001:DB8:3C4D:15::/64 [0/0]
     via ::, Serial0/0/1
L   2001:DB8:3C4D:15:21A:6DFF:FE37:A44E/128 [0/0]
     via ::, Serial0/0/1
L   FE80::/10 [0/0]
     via ::, Null0
L   FF00::/8 [0/0]
     via ::, Null0
R3(config-if)#
```

Tekrar Corp router'ından R1, R2 ve R3 router'larına linklerin her iki tarafında da tamamıyla aynı subnet IPv6 adresini kullandığıma dikkat edin. Şimdi routing protokolleri eklemeye başlayalım.

## RIPng Yapılandırmak

Bu gerçekten basit bir bölümdür. Tüm yapmamız gereken, router'larımızın tamamındaki interface'lere gitmek ve bir komut yazmaktır:

```
Corp#config t
Corp(config)#int f0/1
Corp(config-if)#ipv6 rip ?
  WORD  User selected string identifying this RIP process
Corp(config-if)#ipv6 rip 1 enable
Corp(config-if)#int s0/0/0
Corp(config-if)#ipv6 rip 1 enable
Corp(config-if)#int s0/0/1
Corp(config-if)#ipv6 rip 1 enable
Corp(config-if)#int s0/1/0
Corp(config-if)#ipv6 rip 1 enable
Corp(config-if)#int s0/2/0
Corp(config-if)#ipv6 rip 1 enable
```

Gelin R1 router'ıda yapılandıralım:

```
R1#config t
R1(config)#int s0/0/0
R1(config-if)#ipv6 rip 1 enable
R1(config-if)#int s0/0/1
R1(config-if)#ipv6 rip 1 enable
```

R2 config:

```
R2#config t
R2(config)#int s0/2/0
R2(config-if)#ipv6 rip 1 enable
```

R3 config:

```
R3#config t
R3(config)#int s0/0/1
R3(config-if)#ipv6 rip 1 enable
```

IPv6 routing tablolarımı ve yapılandırmalarımı doğrulama zamanıdır.

## RIPng'i Doğrulamak

Genelde kullanılan `show ip route` komutuyla başlayacağım. R3 router çıktısı aşağıdadır:

```
R3#sh ipv6 route
R   2001:DB8:3C4D:11::/64 [120/2]
     via FE80::21A:2FFF:FE55:C9E8, Serial0/0/1
R   2001:DB8:3C4D:12::/64 [120/2]
     via FE80::21A:2FFF:FE55:C9E8, Serial0/0/1
R   2001:DB8:3C4D:13::/64 [120/2]
     via FE80::21A:2FFF:FE55:C9E8, Serial0/0/1
R   2001:DB8:3C4D:14::/64 [120/2]
     via FE80::21A:2FFF:FE55:C9E8, Serial0/0/1
```

```
C   2001:DB8:3C4D:15::/64 [0/0]
     via ::, Serial0/0/1
L   2001:DB8:3C4D:15:21A:6DFF:FE37:A44E/128 [0/0]
     via ::, Serial0/0/1
L   FE80::/10 [0/0]
     via ::, Null0
L   FF00::/8 [0/0]
     via ::, Null0
R3#
```

Administrative distance ve hop count içeren, normal IPv4 RIP tablosu gibi görünüyor. 11, 12, 13, 14 ve 15 subnet'lerini görebiliyorum.

Birkaç doğrulama komutuna daha bakalım:

```
R3#sh ipv6 protocols
IPv6 Routing Protocol is "connected"
IPv6 Routing Protocol is "static"
IPv6 Routing Protocol is "rip 1"
  Interfaces:
    Serial0/0/1
  Redistribution:
    None
R3#
```

`show ipv6 protocols` komutuyla çok fazla bilgi alamadık. Gelin `show ipv6 rip` komutunu deneyelim:

```
R3#sh ipv6 rip
RIP process "1", port 521, multicast-group FF02::9, pid 60
     Administrative distance is 120. Maximum paths is 16
     Updates every 30 seconds, expire after 180
     Holddown lasts 0 seconds, garbage collect after 120
     Split horizon is on; poison reverse is off
     Default routes are not generated
     Periodic updates 44, trigger updates 19
  Interfaces:
    Serial0/0/1
  Redistribution:
    None
```

Administrative distance'ın hala 120 olduğunu, artı multicast group, maximum path'leri ve timer'ları görebiliyoruz. Öyleyse, gelin devam edelim ve `show ipv6 interface s0/0/1` komutu ile başlayarak, iki doğrulama komutunu daha deneyelim:

```
R3#sh ipv6 interface serial 0/0/1
Serial0/0/1 is up, line protocol is up
  IPv6 is enabled, link-local address is FE80::21A:6DFF:FE37:A44E
  Global unicast address(es):
    2001:DB8:3C4D:1:21A:6DFF:FE37:A44E, subnet is 2001:
DB8:3C4D:1::/64 [EUI]
  Joined group address(es):
    FF02::1
    FF02::2
    FF02::9
    FF02::1:FF37:A44E
  MTU is 1500 bytes
  ICMP error messages limited to one every 100 milliseconds
  ICMP redirects are enabled
  ND DAD is enabled, number of DAD attempts: 1
  ND reachable time is 30000 milliseconds
  Hosts use stateless autoconfig for addresses.
```

Bu da bize bazı güzel bilgiler sağlamaktadır. Fakat durun, en faydalısı şimdi geliyor: `debug ipv6 rip` komutu:

```
R3#debug ipv6 rip
*May 24 18:31:11.959: RIPng: Sending multicast update on
Serial0/0/1 for 1
*May 24 18:31:11.959:        src=FE80::21A:6DFF:FE37:A44E
*May 24 18:31:11.959:        dst=FF02::9 (Serial0/0/1)
*May 24 18:31:11.959:        sport=521, dport=521, length=32
*May 24 18:31:11.959:        command=2, version=1, mbz=0, #rte=1
*May 24 18:31:11.959:        tag=0, metric=1, prefix=2001:
DB8:3C4D:1::/64
*May 24 18:40:44.079: %LINEPROTO-5-UPDOWN: Line protocol on
Interface
   Serial0/0/0, changed state to down
*May 24 18:31:24.959: RIPng: response received from
   FE80::21A:2FFF:FE55:C9E8 on Serial0/0/1 for 1
*May 24 18:31:24.959:        src=FE80::21A:2FFF:FE55:C9E8
(Serial0/0/1)
*May 24 18:31:24.959:        dst=FF02::9
*May 24 18:31:24.959:        sport=521, dport=521, length=32
*May 24 18:31:24.959:        command=2, version=1, mbz=0, #rte=1
*May 24 18:31:24.959:        tag=0, metric=16,
   prefix=2001:DB8:3C4D:12::/64
*May 24 18:31:24.959: RIPng: 2001:DB8:3C4D:12::/64, path
   FE80::21A:2FFF:FE55:C9E8/Serial0/0/1 unreachable
*May 24 18:31:24.959: RIPng: 2001:DB8:3C4D:12::/64, expired, ttg
is 120
```

```
*May 24 18:31:24.959: RIPng: Triggered update requested
*May 24 18:31:25.959: RIPng: generating triggered update for 1
*May 24 18:31:25.959: RIPng: Suppressed null multicast update on
   Serial0/0/1 for 1
```

Şimdi bu ilginçtir. Kullanılan kaynak ve hedef port'larının 521 (evet hala UDP kullanıyoruz) ve network/subnet12'nin erişilemez olduğunu görüyoruz. RIPng'nin, hala bazı temel IPv4 RIP özelliklerine sahip olduğunu görebiliriz. Gelin gidelim ve rouer'larımıza, OSPFv3'ü ekleyelim.

## OSPFv3'ü Yapılandırmak

RIPng yapılandırması gibi ağ topluluğumuzda, OSPF'i aktif hale getirmek için tüm yapmamız gereken çalışmasını istediğimiz interface'lere gitmektir.

Corp yapılandırması şöyledir:

```
Corp#config t
Corp(config)#int f0/1
Corp(config-if)#ipv6 ospf 1 ?
  area  Set the OSPF area ID
Corp(config-if)#ipv6 ospf 1 area 0
Corp(config-if)#int s0/0/1
Corp(config-if)#ipv6 ospf 1 area 0
Corp(config-if)#int s0/1/0
Corp(config-if)#ipv6 ospf 1 area 0
Corp(config-if)#int s0/2/0
Corp(config-if)#ipv6 ospf 1 area 0
Corp(config-if)#^Z
Corp#
```

Bu çok kötü değildi, aslında, IPv4'den biraz kolaydı. Gelin diğer üç router'ımızı yapılandıralım:

```
R1#config t
R1(config)#int s0/0/1
R1(config-if)#ipv6 ospf 1 area 0
R1(config-if)#
*May 24 19:24:55.279: %OSPFv3-5-ADJCHG: Process 1, Nbr 172.16.10.2 on
   Serial0/0/1 from LOADING to FULL, Loading Done
```

R1 Corp router ile adjacent oldu. Bir ilginçlik de, IPv4 RID'in, OSPFv3 adjacent değişikliğinde kullanılıyor olmasıdır.

```
R2#config t
R2(config)#int s0/2/0
R2(config-if)#ipv6 ospf 1 area 0
R2(config-if)#
*May 24 19:27:31.399: %OSPFv3-5-ADJCHG: Process 1, Nbr 172.16.10.3 on
   Serial0/1/0 from LOADING to FULL, Loading Done
```

Adjacency olduğumuz görünmektedir. Tek router kaldı, ondan sonra bazı doğrulamalar uygulayacağız:

```
R3#config t
R3(config)#int s0/0/1
R3(config-if)#ipv6 ospf 1 area 0
R3(config-if)#
*May 24 19:29:07.231: %OSPFv3-5-ADJCHG: Process 1, Nbr 172.16.10.4 on
   Serial0/2/0 from LOADING to FULL, Loading Done
```

Ağımızı doğrulamadan bile, her şey çalışıyor gibi geliyor bana. Fakat hala doğrulama yapmak zorundayız.

## OSPFv3'ü Doğrulamak

Genellikle yapıldığı gibi `show ipv6 route` komutu ile başlayacağım:

```
R3#sh ipv6 route
IPv6 Routing Table - 7 entries
O   2001:DB8:3C4D:11::/64 [110/65]
     via FE80::21A:2FFF:FE55:C9E8, Serial0/0/1
O   2001:DB8:3C4D:13::/64 [110/128]
     via FE80::21A:2FFF:FE55:C9E8, Serial0/0/1
O   2001:DB8:3C4D:14::/64 [110/128]
     via FE80::21A:2FFF:FE55:C9E8, Serial0/0/1
C   2001:DB8:3C4D:15::/64 [0/0]
     via ::, Serial0/0/1
L   2001:DB8:3C4D:15:21A:6DFF:FE37:A44E/128 [0/0]
     via ::, Serial0/0/1
L   FE80::/10 [0/0]
     via ::, Null0
L   FF00::/8 [0/0]
     via ::, Null0
R3#
```

Mükemmel, 12 hariç tüm subnetleri görebiliyorum. (12, arızalı olduğundan göremiyoruz). Gelin `show ipv6 protocols` komutuna bir bakalım:

```
R3#sh ipv6 protocols
IPv6 Routing Protocol is "connected"
IPv6 Routing Protocol is "static"
IPv6 Routing Protocol is "rip 1"
  Interfaces:
    Serial0/0/1
  Redistribution:
    None
IPv6 Routing Protocol is "ospf 1"
  Interfaces (Area 0):
    Serial0/0/1
  Redistribution:
      None
```

Sonraki komut için daha fazla bağlantı görebilmek için tekrar Corp router'ına gitmek istiyorum: `show ipv6 ospf neighbor`.

```
Corp#sh ipv6 ospf neighbor
Neighbor ID     Pri   State   Dead Time  Interface ID  Interface
172.16.10.4       1   FULL/  - 00:00:36    6        Serial0/2/0
172.16.10.3       1   FULL/  - 00:00:33    16       Serial0/1/0
172.16.10.2       1   FULL/  - 00:00:30    6        Serial0/0/1
Corp#
```

Bekleyin, debugging komutlarını çalıştırmamız gerekmektedir. Onlardan ikisini kullanacağım: `debug ipv6 ospf packet` ve `debug ipv6 ospf hello` (neredeyse IPv4'de kullandığımızla aynı komutlar):

```
Corp#debug ipv6 ospf packet
  OSPFv3 packet debugging is on
Corp#
*May 24 19:38:12.283: OSPFv3: rcv. v:3 t:1 l:40 rid:172.16.10.3
      aid:0.0.0.0 chk:E1D2 inst:0 from Serial0/1/0
Corp#
*May 24 19:38:15.103: OSPFv3: rcv. v:3 t:1 l:40 rid:172.16.10.4
      aid:0.0.0.0 chk:7EBB inst:0 from Serial0/2/0
Corp#
*May 24 19:38:18.875: OSPFv3: rcv. v:3 t:1 l:40 rid:172.16.10.2
      aid:0.0.0.0 chk:192D inst:0 from Serial0/0/1
Corp#
*May 24 19:38:22.283: OSPFv3: rcv. v:3 t:1 l:40 rid:172.16.10.3
      aid:0.0.0.0 chk:E1D2 inst:0 from Serial0/1/0
Corp#un all
All possible debugging has been turned off
Corp#debug ipv6 ospf hello
  OSPFv3 hello events debugging is on
Corp#
*May 24 19:38:32.283: OSPFv3: Rcv hello from 172.16.10.3 area 0
from
   Serial0/1/0 FE80::213:60FF:FE20:4E4C interface ID 16
*May 24 19:38:32.283: OSPFv3: End of hello processing
Corp#
*May 24 19:38:35.103: OSPFv3: Rcv hello from 172.16.10.4 area 0
from
   Serial0/2/0 FE80::21A:6DFF:FE37:A44E interface ID 6
*May 24 19:38:35.103: OSPFv3: End of hello processing
Corp#
*May 24 19:38:38.875: OSPFv3: Rcv hello from 172.16.10.2 area 0
from
   Serial0/0/1 FE80::21A:6DFF:FE64:9B2 interface ID 6
```

```
*May 24 19:38:38.875: OSPFv3: End of hello processing
Corp#un all
Tüm geçerli debugging'ler kapatıldı.
Corp#
```

Gerçek dünyadaki gibi bozuk bir interface'e sahip olsak bile, bu benim eğlenceli kabul ettiğim bir modüldür. Umarım siz de, bu modülü benim gibi değerli ve ilginç bulmuşsunuzdur. IPv6'yı öğrenmek için yapabileceğiniz en iyi şey, bazı router'ları almak ve birileri onu kurcaladıktan sonra sizin kurcalamanızdır

# Özet

Bu bölümde, IPv6'yı çok temel anlamda ve IPv6'nın bir Cisco ağ topluluğunda nasıl çalışır hale getirildiğini ele aldım. Bu bölümü okuyarak öğrendiğiniz gibi, temellerini tartıştığınız ve yapılandırdığınızda bile, anlayacak çok şey vardır. CCNA hedeflerine ulaşmanız gerekenden daha fazlasını bildiğinize emin olun.

Neden IPv6'ya ihtiyacımız olduğunu ve onunla ilgili avantajlardan bahsederek başladım. Hem IPv6 ile adresleme hem de ifadelerin nasıl kısaltıldığını anlatarak devam ettim. IPv6 ile adreslemeden bahsettiğim esnada, farklı adres türlerini ve IPv6'da rezerve edilmiş özel adresleri gösterdim.

IPv6, çoğunlukla otomatik olarak dağıtılacaktır. Yani host'lar oto-yapılandırma kullanır, bu nedenle IPv6'nın autoconfiguration'ı nasıl kullandığından ve bir Cisco router yapılandırıldığında devreye nasıl girdiğinden bahsettim. Bundan sonra, router'a bir DHCP sunucusu nasıl eklendiğini gösterdim. Böylece router, host'lara, DNS sunucu adresi gibi seçenekler sağlayabilir.

ICMP, IPv6 için çok önemlidir. ICMP'nin, IPv6 ile nasıl çalıştığını gösterdim ve RIP, EIGRP ve OSPF'in, IPv6 ile nasıl yapılandırıldığıyla devam ettim.

IPv6'ya geçiş küçük bir olay değildir ve bunu yapmanın lehte ve aleyhte görüşlerini detaylı olarak inceledim. Ve size üç geçiş stratejisinden bahsettim: dual stacking, hem IPv4 hem de IPv6 kullanarak yapılan tunelling ve sadece son çare olarak kullanılan NAT-PT.

Son olarak, bu kitap boyunca kullandığım ağ topluluğunda, IPv6'nın nasıl yapılandırıldığını ve sonra IPv6 ile uygun çeşitli show komutları ile yapılandırmanın nasıl doğrulandığını gösterdim.

# Sınav Gereklilikleri

**IPv6'ya neden ihtiyacımız olduğunu anlamak:** IPv6'sız, IP adresleri tükenmiş bir dünya olacaktır.

**Link-local'i anlamak:** Link-local, IPv4'deki özel adreslere benzer, fakat route edilemezler, hatta kendi kurumunuzda bile.

**Uniqe local'i anlamak:** Bu link-local gibi, IPv4'deki özel adreslere benzer, fakat internete route edilemez. Bununla birlikte, link-local ile uniqe local arasındaki farklılık, uniqe local'in, kurum veya şirketinizde route edilebilir olmasıdır.

**IPv6 adreslemeyi hatırlamak:** IPv6 adresleme, IPv4 adreslemeye benzemez. IPv6, sadece 32 bit olan ve ondalık olarak gösterilen IPv4'ün tersine, daha fazla adres uzayına sahiptir, 128 bit uzunluğundadır ve hexadecimal olarak belirtilir.

# Yazılı Lab 13

Bu bölümde, aşağıdaki IPv6 sorularını cevaplarını yazın:

1. Hangi paket tipi, sadece bir interface'e adreslenir ve teslim edilir?
2. Hangi adres tipi, IPv4'deki düzenli, genel yönlendirilebilir adrestir?
3. Hangi adres tipinin route edilmesi düşünülmez?
4. Hangi adres tipi, global olarak hala benzersiz olduğu halde, İnternet'e yönlendirilmesi düşünülmez?
5. Hangi adres tipinin, çoklu interface'lere teslim edilmesi düşünülmektedir?
6. Hangi adres tipi çok sayıda interface'i tespit eder, fakat paketleri, ilk bulduğu adrese teslim eder?
7. Hangi routing protokol, FF02::5 multicast adresini kullanır?
8. IPv4, 127.0.0.1 loopback adresine sahipti. IPv6 loopback adresi nedir?
9. Bir link-local adresi ne ile başlar?
10. Bir uniqe local unicast aralığı ne ile başlar?

(Yazılı Lab 13'ün cevapları, bu bölüm için gözden geçirme sorularına cevaptan sonra bulunabilir.)

# Gözden Geçirme Soruları

*Aşağıdaki sorular, bu bölümün materyallerini anladığınızı test etmek için tasarlanmıştır. İlave sorulara nasıl sahip olacağınızla ilgili daha fazla bilgi için,bu kitabın Giriş bölümüne bakın.*

NOT

1. Bir global unicast adresini tanımladığınızda aşağıdakilerden hangisi doğrudur?
   A. Bir unicast adrese yönlendirilen paketler, tek bir interface'e teslim edilmektedir.
   B. Bunlar, IPv4'deki düzenli yönlendirilebilir bir adres gibi, tipik olarak genel yönlendirilebilir adreslerdir.
   C. Bunlar, route edilmesi düşünülmeyen, IPv4'deki özel adreslere benzemektedir.
   D. Bu adresler, non-routing amaçlar için düşünülmektedir, fakat nerdeyse global olarak benzersizdirler, bu yüzden bir adres çakışması olasılık dışıdır.
2. Bir unicast adresini tanımladığınızda aşağıdakilerden hangisi doğrudur?
   A. Bir unicast adrese yönlendirilen paketler, tek bir interface'e teslim edilmektedir.
   B. Bunlar, IPv4'deki düzenli yönlendirilebilir bir adres gibi tipik olarak genel yönlendirilebilir adreslerdir.
   C. Bunlar, route edilmesi düşünülmeyen, IPv4'deki özel adreslere benzemektedir.
   D. Bu adresler, nonrouting amaçlar için düşünülmektedir, fakat nerdeyse global olarak benzersizdirler, bu yüzden bir adres çakışması olasılık dışıdır.
3. Bir link-local adresi tanımladığınızda aşağıdakilerden hangisi doğrudur?
   A. Bir unicast adrese yönlendirilen paketler, tek bir interface'e teslim edilmektedir.
   B. Bunlar, IPv4'deki düzenli yönlendirilebilir bir adres gibi, tipik olarak genel yönlendirilebilir adreslerdir.
   C. Bunlar, route edilmesi düşünülmeyen, IPv4'deki özel adreslere benzemektedir.
   D. Bu adresler, nonrouting amaçlar için düşünülmektedir, fakat nerdeyse global olarak benzersizdirler, bu yüzden bir adres çakışması olasılık dışıdır.
4. Bir uniqe local adresi tanımladığınızda, aşağıdakilerden hangisi doğrudur?
   A. Bir unicast adrese yönlendirilen paketler, tek bir interface'e teslim edilmektedir.
   B. Bunlar, IPv4'deki düzenli yönlendirilebilir bir adres gibi, tipik olarak genel yönlendirilebilir adreslerdir.
   C. Bunlar route edilmesi düşünülmeyen, IPv4'deki özel adreslere benzemektedir.
   D. Bu adresler nonrouting amaçlar için düşünülmektedir, fakat nerdeyse global olarak benzersizdirler, bu yüzden bir adres çakışması olasılık dışıdır.
5. Bir multicast adresini tanımladığınızda aşağıdakilerden hangisi doğrudur?
   A. Bir unicast adrese yönlendirilen paketler, tek bir interface'e teslim edilmektedir.
   B. Paketler, adres tarafından belirlenen tüm interface'lere teslim edilir. Bu ayrıca, one-to-many adres olarak belirtilir.
   C. Çok sayıda interface'i belirler ve sadece bir adrese teslim edilir. Bu adres, one-to-one-of-many olarak belirtilir.
   D. Bu adresler nonrouting amaçlar için düşünülmektedir, fakat nerdeyse global olarak benzersizdirler, bu yüzden bir adres çakışması olasılık dışıdır.

6. Bir anycast adresini tanımladığınızda aşağıdakilerden hangisi doğrudur?
   A. Bir anycast adrese yönlendirilen paketler, tek bir interface'e teslim edilmektedir.
   B. Paketler, adres tarafından belirlenen tüm interface'lere teslim edilir. Bu ayrıca, one-to-many adres olarak belirtilir.
   C. Bu adres çok sayıda interface'i belirler ve anycast paketi, sadece bir adrese teslim edilir. Bu adres ayrıca, one-to-one-of-many olarak belirtilir.
   D. Bu adresler nonrouting amaçlar için düşünülmektedir, fakat nerdeyse global olarak benzersizdirler, bu yüzden bir adres çakışması olasılık dışıdır.
7. Lokal host'unuzun loopback adresine ping atmak istiyorsunuz. Ne yazmalısınız?
   A. `ping 127.0.0.1`
   B. `ping 0.0.0.0`
   C. `ping ::1`
   D. `trace 0.0.::1`
8. OSPFv3'ün kullandığı iki multicast adresi nedir?
   A. FF02::A
   B. FF02::9
   C. FF02::5
   D. FF02::6
9. RIPng'nin kullandığı multicast adresleri nelerdir?
   A. FF02::A
   B. FF02::9
   C. FF02::5
   D. FF02::6
10. EIGRPv6'nın kullandığı multicast adresleri nelerdir?
    A. FF02::A
    B. FF02::9
    C. FF02::5
    D. FF02::6
11. RIPng'yi aktif hale getirmek için aşağıdakilerden hangisini kullanırsınız?
    A. `Router1(config-if)# ipv6 ospf 10 area 0.0.0.0`
    B. `Router1(config-if)#ipv6 router rip 1`
    C. `Router1(config)# ipv6 router eigrp 10`
    D. `Router1(config-rtr)#no shutdown`
    E. `Router1(config-if)#ipv6 eigrp 10`
12. EIGRPv6'yı aktif hale getirmek için aşağıdakilerden hangisini kullanırsınız?
    A. `Router1(config-if)# ipv6 ospf 10 area 0.0.0.0`
    B. `Router1(config-if)#ipv6 router rip 1`
    C. `Router1(config)# ipv6 router eigrp 10`
    D. `Router1(config-rtr)#no shutdown`
    E. `Router1(config-if)#ipv6 eigrp 10`

13. OSPFv3'ü aktif hale getirmek için aşağıdakilerden hangisini kullanırsınız?
    A. Router1(config-if)# ipv6 ospf 10 area 0.0.0.0
    B. Router1(config-if)#ipv6 router rip 1
    C. Router1(config)# ipv6 router eigrp 10
    D. Router1(config-rtr)#no shutdown
    E. Router1(config-if)#ospf ipv6 10 area 0
14. IPv6 adresleriyle ilgili aşağıdaki ifadelerin hangisi doğrudur? (İki şık seçin.)
    A. Sol tarafta sıfırların olması gerekmektedir.
    B. Sıfırın ardışık hexadecimal alanlarını belirtmek için, (::) kullanılır.
    C. Alanları ayırmak için, (::) kullanılır.
    D. Tek bir interface, çok sayıda, farklı tip IPv6 adresine sahip olacaktır.
15. IPv4 ve IPv6 adresleri hakkında hangi iki ifade doğrudur?
    A. Bir IPv6 adresi, hexadecimal'de belirtilen 32 bit uzunluğundadır.
    B. Bir IPv6 adresi, ondalık olarak belirtilen 128 bit uzunluğundadır.
    C. Bir IPv6 adresi, ondalık olarak belirtilen 32 bit uzunluğundadır.
    D. Bir IPv6 adresi, hexadecimal'de belirtilen 128 bit uzunluğundadır.

# Gözden Geçirme Sorularının Cevapları

1. B Unicast adreslerin tersine, global unicast adresler route edilir.
2. A Açıklama: Bir unicast adrese yönlendirilen paketler, tek bir interface'e teslim edilmektedir. Yük dengelemesi için birçok interface,aynı adresi kullanabilir.
3. C Link-local adresleri, geçici bir LAN'ın, karşılaşma için birlikte gönderilmesi ya da route edilmeyecek olan, fakat dosya ve servislere lokal erişimi ve paylaşımı gerektiren küçük LAN için kullanılır.
4. D bu adresler, link-local gibi nonrouting amaçlar için düşünülmezler, fakat nerdeyse global olarak benzersizdirler, bu yüzden bir adres çakışması olasılık dışıdır. Uniqe local adresler, site-local adreslerin yerine konulması amacıyla tasarlanmışlardır.
5. B Bir multicast adrese yönlendirilen paketler, IPv4'deki gibi adres tarafından belirlenen tüm interface'lere teslim edilir. Bu ayrıca, one-to-many adres olarak belirtilir. Multicast adresler daima FF ile başladıklarından, IPv6'daki bir multicast adresini daima tespit edebilirsiniz.
6. C Anycast adresleri, multicast adreslerindeki gibi birçok interface'i tanımlar. Bununla birlikte büyük farklılık, anycast adresinin, tek bir adrese (routing uzaklığıyla belirtilen, bulduğu ilk adrese) teslim edilmesidir. Bu adres ayrıca, one-to-one-of-many olabilir.
7. C IPv4 ile loopback adresi, 127.0.0.1'dir. IPv6 ile bu adres, ::1'dir.
8. C,D Adjacency ve next-hop attribute'lar şimdi link-local adresleri kullanır. OSPFv3, güncelleme ve acknowledgment'ları göndermek için, OSPF router'lar için FF02::5 ve OSPF designated router'lar için, FF02::6 adresleri ile multicast trafiği kullanırlar. Bunlar sırayla, 224.0.0.5 ve 224.0.0.6'nın yerine geçmişlerdir.
9. B RIPng, FF02::9 IPv6 multicas adresi kullanır. IPv4 için multicast adresini hatırlarsanız, her IPv6 adresin sonundaki numarayla aynıdır.
10. A EIGRPv6'nın multicast adresi nerdeyse aynı kalmıştır. IPv4'de, o 224.0.0.10'du, şimdi FF02::A'dır. (A, hexadecimal gösterimde,10'a eşittir)
11. B IPv6 için RIPng'yi aktif hale getirmek oldukça basittir. RIP'ın çalışmasını istediğiniz interface'de, `ipv6 router rip` ***`number`*** komutuyla onu yapılandırabilirsiniz.
12. C,D,E RIPng ve OSPFv3'ün tersine, EIGRP'yi, hem global configuration hem de interface moddan yapılandırmanız gerekir ve komutu, `no shutdown` ile etkinleştirmeniz gerekir.
13. A OSPFv3'ü aktif hale getirmek için protokolü, RIPng'deki gibi etkinleştirin. Komut dizini, `ipv6 ospf process-id area` ***`area-id`***'dir.
14. Bir IPv6 adresin yazılı uzunluğunu kısaltmak için, ardışık sıfır alanları yerine, (::) kullanılabilir. Adresi biraz daha kısaltmayı denersek, sol taraftaki sıfırlar kaldırılabilir. IPv4'de olduğu gibi, tek bir cihazın interface'i, birden fazla adrese sahip olabilir. IPv6 ile daha fazla çeşit adres vardır ve aynı kural uygulanır. Tamamı aynı adrese atanmış, link-local, global unicast ve multicast adresleri olabilir.
15. C,D IPv4 adresleri, 32 bit uzunluğundadır ve ondalık formatta gösterilirler. IPv6 adresleri, 128 bit uzunluğunda ve hexadecimal olarak gösterilirler.

# Yazılı Lab 13'ün Cevapları

1. Unicast
2. Global unicast
3. Link-local
4. Unique local (site-local olarak belirtilir)
5. Multicast
6. Anycast
7. OSPF
8. ::1
9. FE80::
10. FC00::

# 14 Wide Area Network

# 14 Wide Area Network

- Wide Area Metwork'lere Giriş
- Kablo ve DSL
- Serial Wide Area Network'leri (WAN) Kablolamak
- High-Level Data-Link Control (HDLC) Protocol
- Point-to-Point Protocol (PPP)
- Frame Relay
- WAN Bağlantıları İçin SDM Kullanmak
- Virtual Private Network
- Özet
- Sınav Gereklilikleri
- Yazılı Lab 14
- Pratik Lab'lar
- Pratik Lab 14.1: PPP Enkapsülasyon ve Kimlik Doğrulama Yapılandırmak
- Pratik Lab 14.2: HDLC'yi Yapılandırmak ve Görüntülemek
- Pratik Lab 14.3: Frame Relay ve Subinterface'leri Yapılandırmak
- Gözden Geçirme Soruları
- Gözden Geçirme Sorularının Cevapları
- Yazılı Lab 11'in Cevapları

# Wide Area Network

Cisco IOS, uzaktaki diğer LAN'lara yerel LAN'larınızı genişletmenize yardımcı olacak çok sayıda farklı wide area network (WAN) protokolünü destekler. Ayrı bölgeler arasında bu günlerde ne kadar çok önemli bilginin değiş tokuş edildiğini söylememe gerek olduğunu zannetmiyorum. Fakat öyle olsa da, kendi yapısal kablolamanızı yapmanız ve şirketinizin tüm uzak lokasyonlarına kendinizin bağlanması çok uygun maliyetli veya randımanlı olmayabilir. Servis sağlayıcıların zaten sahip oldukları mevcut altyapıyı kiralamak, çok daha iyi bir çözümdür ve büyük zaman kazandırır.

Bu bölümde, WAN'lara uygun olarak kullanılan, farklı tip bağlantı, teknoloji ve cihazlardan bahsederek devam edeceğim. Ayrıca High-Level Data-Link Control (HDLC), Point-to-Point Protocol (PPP), Point-to-Point Protocol over Ethernet (PPPoE), kablo, DSL ve Frame Relay'in nasıl çalıştığına ve yapılandırılacağına da değineceğiz.

*Bu bölümde, WAN çözümü olarak wireless ağlardan bahsedebilecek olsam da, onun yerine Cisco'nun wireless çözümleri ve şirket içi ağları ve dış metropolitan area'larda onların nasıl kullanıldığıyla ilgili dolu dolu detaylı bilgileri bulabileceğiniz bölüm 12'yi size referans olarak gösterebilirim.*

**NOT**

Anlayacağınız gibi, burada Cisco'nun desteklediği tüm WAN türlerini işlemeyeceğim. Bu kitabın hedefi, CCNA konuları ile ilgili tam anlamıyla başarılı bir şekilde donatılmış olmanızı sağlamaktır. Bundan dolayı kablo, DSL, HDLC, PPP, PPPoE ve Frame Relay'e odaklanacağım. Sonra, VPN'e güçlü bir giriş ile bu bölümü tamamlayacağım.

İlk olarak, WAN temellerinin incelenmesiyle başlayalım.

*Bu bölüm ile ilgili son güncellemeler için www.lammle.com ve/veya www.sybex.com adreslerine bakınız.*

**NOT**

## Wide Area Metwork'lere Giriş

Local area network'lerin (LAN) yerine wide area network'lerin (WAN) kullanılmasını gerektiren nedir? Açıkçası, mesafeyle ilgili konulardır, fakat günümüzde wireless LAN'lar oldukça geniş alanları kapsamaktadır. Peki ya bant genişlikleri? Burada da gerçekten bazı büyük kanallar, büyük maliyetlere sebep olabilecektir. Yani, bu da çözüm olmayacaktır. Peki, o zaman ne yapacağız?

WAN'ı LAN'dan ayıran ana farklılıklardan birisi, genellikle bir LAN altyapısına kendiniz sahipken, WAN'ı daha çok bir servis sağlayıcısından kiralamanızdır. Dürüst olmak gerekirse, modern teknolojiler bile bu tanımlamayı bulanıklaştırmaktadır, fakat o hala Cisco'nun sınav hedeflerine uygundur.

Genel olarak edindiğiniz (Ethernet) veri hattından zaten bahsetmiştim, fakat şimdi, daha çok bir servis sağlayıcıdan tedarik edilen türlerini öğreneceğiz.

WAN teknolojilerini anlamanın anahtarı, ağlarınızı birbirleriyle birleştirmek için yaygın olarak servis sağlayıcı tarafından kullanılan, farklı WAN terimlerine ve bağlantı türlerine aşina olmaktır.

### WAN Terimlerini Açıklamak

Gidip, bir servis sağlayıcıdan WAN servisi almadan önce, tipik olarak bir servis sağlayıcının kullandığı aşağıdaki terimleri anlamak, gerçekten iyi bir fikir olacaktır:

**Customer premises equipment (CPE):** Customer premises equipment (CPE), abone tarafından sağlanan ve abonenin binasında yer alan ekipmandır.

**Demarcation point:** Demarcation point, servis sağlayıcının sorumluluğunun bittiği ve CPE'nin başladığı kesin noktadır. Genel olarak, telekom firması tarafından sağlanan ve kurulan telekom kabinindeki bir cihazdır. Bu kutudan CPE'ye kablo bağlantısı, sizin sorumluluğunuzdadır. Genellikle bir CSU/DSU veya ISDN interface'e bağlantıdır.

**Local loop:** Local loop, demarc'ı, central office denilen en yakın switching santraline bağlar.

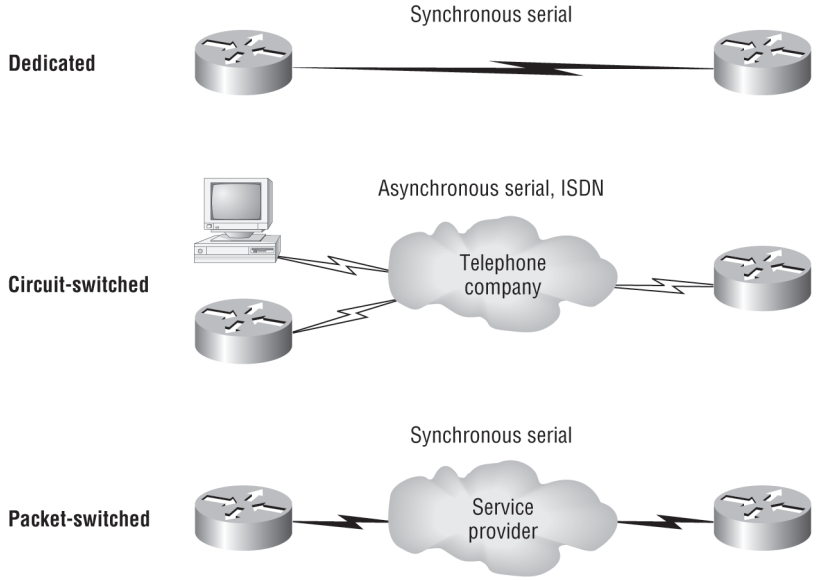
**Central office (CO):** Bu nokta, müşteri ağını, sağlayıcının switching ağına bağlar. Central Office'in (CO), bazen point of presence (POP*)* olarak belirtildiğini bilmek iyidir.

**Toll network:** Toll network, bir WAN sağlayıcı ağının içindeki trunk hattıdır. Bu ağ, ISP tarafından sağlanan switch ve cihazların koleksiyonudur.

WAN teknolojilerini anlamak için önemli olduklarından, bu terimlere kesinlikle aşina olmalısınız.

## WAN Bağlantı Türleri

Muhtemelen haberdar olduğunuz gibi, bir WAN, çok sayıda bağlantı çeşidi kullanabilir ve bugün piyasada kullanılan farklı WAN türlerinin hepsini size sunacağım. Şekil 14.1, bir DCE ağı üzerinden LAN'larınızı (DTE) birbirine bağlamak için kullanılabilecek farklı WAN türlerini göstermektedir.




***Şekil 14.1:*** *WAN bağlantı türleri.*

Aşağıdaki liste, farklı WAN bağlantı tiplerini açıklamaktadır:

**Leased line:** Bunlar genel olarak, point-to-point veya atanmış bağlantılar olarak belirtilir. Bir leased line, CPE'den DCE switch'e, sonra da uzak noktanın CPE'sine giden, önceden kurulu bir WAN iletişim yoludur. CPE, DTE ağlarına, veriyi aktarmadan önce bir çözüm bulmak için kullanışsız kurulum prosedürleri olmadan, istendiğinde haberleşme olanağı sağlar. Çok paranız olduğunda, 45 Mbps'a kadar senkron seri hatlar kullandıklarından, bu gerçekten sizin için çok iyi olur. HDLC ve PPP encapsulation, leased line'larda oldukça sık kullanılır. Onların detayına daha sonra gireceğim.

**Circuit switching:** Circuit switching terimini duyduğunuzda, aklınıza telefon konuşması gelsin. En büyük avantajı maliyettir. Sadece kullandığınız kadar para ödersiniz. Uçtan-uca bağlantı kurulmadan önce, veri transferi olmaz. Circuit switching, dial-up modem veya ISDN kullanır ve veri transferlerinde, düşük-bant genişliği kullanılır. Modemleri düşündüğünüzü biliyorum. Modem mi dedin? Bunlar şimdi müzede değiller mi? Wireless teknolojilerinin mümkün olduğu bu günlerde, modemi kim kullanır? Bu soruları sorduğunuzu biliyorum. Bazı insanlar ISDN'e sahip ve o hala geçerlidir (ve ben, modemi bazı insanların şimdi kullandıklarını ve daha sonra da kullanacaklarını düşünüyorum). Circuit switching, bazı yeni WAN teknolojilerinde de kullanılabilir.

**Packet switching:** Bu, tasarruf sağlamak için bant genişliğini diğer şirketlerle paylaşmanıza izin veren bir WAN switching yöntemidir. Packet switching, leased line gibi görünmek üzere tasarlanmış bir ağ gibi düşünülebilir, fakat ücretleri daha çok bir circuit switching gibidir. Fakat düşük maliyet her zaman iyi değildir. Kesinlikle olumsuz bir tarafı vardır: Veriyi sürekli olarak transfer etmeniz gerekirse, bu seçeneği unutun. Yerine, kendinize bir leased line edinin. Paket switching sadece,

veri transferinin devamlı olmadığı durumlarda kullanışlıdır. Frame Relay ve X.25, 56Kbps'dan T3'e (45Mbps) kadar hız aralıklarındaki, packet-switching teknolojileridir.

*MultiProtocol Label Switching (MPLS), circuit switching ve packet switching'in kombinasyonunu kullanır. Fakat bu kitabın alanı dışındadır. Öyle bile olsa, CCNA sınavınızı geçtikten sonra MPLS'e bakmak daha iyi olacaktır. MPLS'den birazdan bahsedeceğim.*

NOT

## WAN Desteği

Cisco, seri interface'lerinde HDLC, PPP ve Frame Relay'i destekler ve siz bunu, bir seri interface'de `encapsulation ?` yazarak görebilirsiniz (çıktınız, kullandığınız IOS versiyonuna bağlı olarak değişebilir):

```
Corp#config t
Corp(config)#int s0/0/0
Corp(config-if)#encapsulation ?
  atm-dxi      ATM-DXI encapsulation
  frame-relay  Frame Relay networks
  hdlc         Serial HDLC synchronous
  lapb         LAPB (X.25 Level 2)
  ppp          Point-to-Point protocol
  smds         Switched Megabit Data Service (SMDS)
  x25          X.25
```

Router'ımda, diğer interface türlerine sahip olsaydım, ISDN veya ADSL gibi diğer enkapsülasyon seçeneklerine de sahip olacağımı bilin. Ayrıca seri bir interface'de, Ethernet veya Token Ring enkapsülasyon yapılandıramayacağınızı hatırlayın.

Günümüzde kullanılan ve yoğun olarak bilinen WAN protokollerini açıklayacağım: Frame Relay, ISDN, LAPB, LAPD, HDLC, PPP, PPPoE, Kablo, DSL, MPLS ve ATM. Bildiğiniz gibi bir seri interface'de genelde kullanıldığını göreceğiniz protokoller sadece HDLC, PPP ve Frame Relay'dir. Fakat wide area bağlantıları için sadece seri interface'ler kullanmak zorunda olduğumuzu kim söyledi?

*Bölümün kalanı, kablo, DSL ve temel WAN protokollerinin nasıl çalıştığı ve onların Cisco Router'larla nasıl yapılandırılacağının detaylı açıklamalarına ayrılmıştır. Fakat en son CCNA sınav konularının ötesinde önemli olduklarından, ISDN, LAPB, LAPD, MPLS ve ATM konularından özel olarak bahsedeceğim. Onlardan birisi, sınav konularında karşınıza çıkarsa endişelenmeyin, bilgi konusunda güncellemeyi en kısa sürede www.lammle.com adresinde bulacağınıza söz veriyorum.*

NOT

**Frame Relay:** 1990'ların başında ortaya çıkan bir packet-switched teknolojisidir. Frame Relay, yüksek-performanslı bir Data Link ve Physical katman düzenlemesidir. Fiziksel hataları (gürültülü hatları) dengelemek için kullanılan X.25'deki teknolojilerin çoğunun çıkartılması haricinde, X.25'in varisidir. Frame Relay'in iyi tarafı, point-to-point linklerden daha uygun maliyetlidir ve tipik olarak 64Kbps'den 45Mbps'e (T3) kadar çalışır. Diğer bir Frame Relay avantajı, dinamik bant genişliği ve tıkanıklık denetimi için özellikler sağlamasıdır.

**ISDN:** Integrated Services Digital Network (ISDN), mevcut telefon hatları üzerindeki ses ve veriyi aktaran dijital servislerin bir bütünüdür. ISDN, analog dial-up linklerin onlara verebileceğinden daha hızlı bir bağlantıya ihtiyacı olan uzak kullanıcılar için uygun maliyetli bir çözüm önerir. Ayrıca Frame Relay veya T1 bağlantıları gibi diğer link türlerine, yedek link olarak iyi bir seçenektir.

**LAPB:** Link Access Procedure, Balanced (LAPB), Data Link katmanında connection-oriented protokol olarak, X.25 ile kullanılmak için oluşturulmuştur. Ayrıca basit bir veri link aktarımı olarak da kullanılabilir. LAPB'nin çok hoş olmayan özelliklerinden biri, onun kesin time-out ve windowing tekniklerinden dolayı, büyük miktarda ek yük oluşturma eğiliminde olmasıdır.

**LAPD:** Link Access Procedure, D-Channel (LAPD), D (sinyal) kanalı için bir protokol olarak, Data Link katmanında (katman2), ISDN ile kullanılır. LAPD, Link Access Procedure, Balanced (LAPB) protokolünden türemiştir ve öncelikle ISDN temel erişimin sinyalleşme gerekliliklerini sağlamak için tasarlanmıştır.

**HDLC:** High-Level Data-Link Control (HDLC), bir Data Link bağlantı protokolü olarak, IBM tarafından geliştirilen Synchronous Data Link Control'den (SDLC) türemiştir. HDLC, Data Link katmanında çalışır ve LAPB ile kıyaslandığında, oldukça düşük ek yük oluşturur.

Aynı link boyunca, çoklu Network katman protokollerini enkapsüle etme eğiliminde değildir. HDLC başlığı, HDLC enkapsülasyon içinde taşınan protokol türleri konusunda bir tanım içermez. Bundan dolayı, HDLC kullanan her üretici, Network katman protokolünü belirlemek için kendi yöntemini kullanır. Yani her üreticinin HDLC'si, onun belirli ekipmanına özeldir.

**PPP:** Point-to-Point Protocol (PPP), yaygın olarak bilinen endüstri-standardı bir protokoldür. HDLC'nin tüm çoklu protokol versiyonlarının tescilli olmasından dolayı, PPP, farklı üreticilerin ekipmanları arasında point-to-point linkleri oluşturmak için kullanılabilir. Network katman protokolünü tespit etmek ve senkron ile asenkron linklerde çalışması için multilink bağlantılar ve authentication sağlaması amacıyla, Data Link başlığındaki Network Control Protocol alanını kullanır.

**PPPoE:** Point-to-Point Protocol over Ethernet, Ethernet frame'lerdeki, PPP frame'lerini enkapsüle eder ve genelde ADSL servisleri ile birlikte kullanılır. Size, authentication, encryption ve compression gibi birçok bilindik PPP özelliği sağlar, fakat olumsuz özellikleri de vardır. Standart Ethernet'ten daha düşük maximum transmission unit'e (MTU) sahiptir ve firewall iyi bir şekilde yapılandırılmamışsa, bu küçük özellik sizi üzebilir.

Ethernet'in ana özelliğindeki PPPoE, DSL desteği de sağlarken, Ethernet interface'lerine direkt bir bağlantı ekler. An az bir bridging modem boyunca farklı hedeflere PPP oturumları açmak için paylaşımlı bir Ethernet interface'deki birçok host tarafından sıkça kullanılmaktadır.

**Kablo:** Modern bir HFC ağında, tipik olarak 500 ile 2,000 aktif abone, gönderilen ve alınan bant genişliğinin tamamının paylaşıldığı belli bir kablo ağ segmentine bağlanır. (HFC,Hybrid fibre-coaxial, geniş bant bir ağ oluşturmak için, fiber optik ve koaksiyel kablonun beraber kullanıldığı bir ağ için, telekomünikasyon endüstri terimidir). Bir cable TV (CATV) boyunca İnternet servisi için gerçek bant genişliği müşteriye download için 27Mbps, upload için 2.5Mbps kadardır. Kullanıcılar genellikle 256Kbps ile 6Mbps arasında erişim hızına sahip olurlar.

**DSL:** Digital subscriber line, sarmal-çift telefon kabloları üzerinden gelişmiş servisler (yüksek hızda veri ve bazen video) taşımak için geleneksel telefon firmaları tarafından kullanılan bir teknolojidir. Genelde, HFC ağlarından daha düşük bir veri taşıma kapasitesine sahiptirler ve veri hızları, hattın uzunluğu ve kalitesiyle sınırlı aralıkta olabilir. DSL, komple uçtan-uca bir çözüm değildir, fakat dial-up, kablo ve wireless gibi bir Physical katman aktarım teknolojisinden daha iyidir. DSL bağlantıları, yerel bir telefon ağının sonuna yerleştirilir. Bağlantı, CPE ve Digital Subscriber Line Access Multiplexer (DSLAM) arasında bir bakır kablonun her iki ucundaki modem çifti arasında kurulur. Bir DSLAM, servis sağlayıcının Central Office'inde (CO) yerleştirilir ve çoklu DSL abonelerinin bağlantılarına yoğunlaşır.

**MPLS:** MultiProtocol Label Switching (MPLS), bir packet-switched ağ boyunca circuit-switched ağın bazı özelliklerini taklit eden bir veri-taşıma mekanizmasıdır. MPLS, paketlere etiketler (numaralar) düzenleyen ve sonra paketleri göndermek için bu etiketleri kullanan bir switching mekanizmasıdır. Etiketler, ağın MPLS kenarına atanır ve MPLS ağına yönlendirme, yalnızca etiketler bazında olur. Etiketler genellikle katman3 hedef adreslere eşlenir. (IP hedef tabanlı routing'e eşittir.) MPLS TCP/IP dışında protokollerin aktarılmasını desteklemek için tasarlanmıştır. Bundan dolayı ağdaki label switching, katman3 protokole bakılmaksızın aynı şekilde çalışır. Büyük ağlarda, MPLS etiketleme sonucunda, sadece border switch'ler, bir routing lookup çalıştırır. Tüm core router'lar, etiketler bazında paketleri gönderir. Bu paketlerin, servis sağlayıcı ağına daha hızlı gönderilmesini sağlar. (Birçok firma artık Frame Relay ağlarını, MPLS ile değiştirmektedir.)

**ATM:** Asynchronous Transfer Mode (ATM), zaman-duyarlı trafiğin oluşturulması için geliştirilmiştir. Ses, veri ve videonun eş zamanlı aktarılmasını sağlar. ATM, paket yerine, 53 byte sabit hücreler (cell) kullanır. Ayrıca verinin daha hızlı aktarılmasına yardımcı olmak için isochronous clocking (harici zaman denetimi) kullanabilir. Günümüzde Frame Relay çalışıyorsanız, ATM üzerinden Frame Relay çalışırsınız.

# Kablo ve DSL

Cisco router'larda kullanılan seri enkapsülasyon bağlantıları (HDLC, PPP ve Frame Relay) konusundan bahsetmeden önce, DSL ve kablo modem ağ kurulumu arasındaki pratik farklılıkları anlamanıza gerçekten yardımcı olacağını düşündüğüm birkaç bilgiyi paylaşacağım.

DSL ve kablo İnternet servisleri, genel olarak aynı şeyleri yapar. Fakat hala, anlamanız gereken bazı önemli temel farklılıklara sahiptirler:

**Hız:** Çoğunuz kablonun DSL internetten daha hızlı olduğunu söyleyebilir, fakat gerçek dünyada kablo her zaman yarışı kazanmaz.

**Güvenlik:** DSL ve kablo, farklı ağ güvenlik modelleri temelinde kurulmuştur ve yakın zamana kadar kablo, bu yarışta kaybeden taraftı olmuştur. Fakat şimdi neredeyse eşittirler ve her ikisi de, çoğu kullanıcının ihtiyaçlarına uygun güvenlik sunmaktadırlar. Uygun dediğimde, her iki alternatifle ilgili bazı gerçek güvenlik sorunları olduğunu kastediyorum. ISP'nizin kim olduğunun önemi yoktur.

**Popülarite:** Kablo internet Amerika'da kesinlikle daha popülerdir, fakat DSL, yakalamaya başlamaktadır.

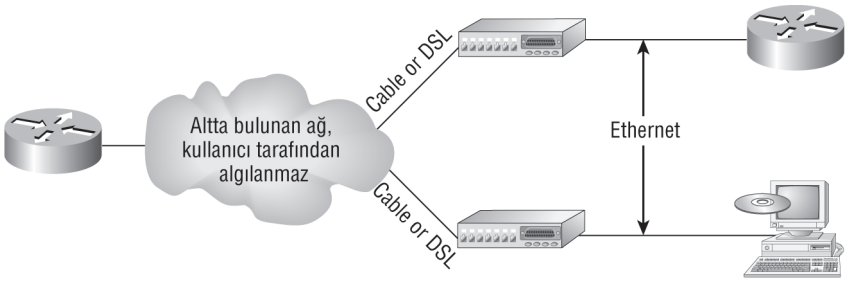
**Müşteri memnuniyeti:** Amerika'da tersi doğrudur. Yani DSL daha iyidir. Fakat ISP'sinden tamamıyla memnun olan birini tanıyor musunuz?

Şekil 14.2, bir bağlantının, modemlerden, direkt olarak bir PC'ye veya router'a nasıl sonlandırılabileceğini göstermektedir. Router'ınız bu interface'de hem DHCP hem de PPPoE çalışabilir. DSL ve kablo yüksek-hızlı İnternet servisleri, dünyadaki milyonlarca ev ve iş kullanıcıları için kullanışlıdır. Fakat bazı yerleşimlerde sadece biri mümkündür.

DSL ve kablo modemler arasındaki farklılıkların bazılarının asıl teknolojilerle hiçbir ilgisi yoktur. Bunlar ISP ile ilgili olanlardır. Kurulum ve onarım sorunları için maliyet, güvenirlik ve müşteri destek kalitesi, bir servis sağlayıcıdan diğerine değişiklik gösterir.

## Kablo

Kablo, bir SOHO (small office/home office) için oldukça düşük maliyetli bir bağlantıdır. Hatta daha büyük firmalar için dahi, kablo (veya DSL), bir yedek link olarak çok iyi olabilir.




***Şekil 14.2:*** *Kablo ve DSL kullanarak broadband erişimi.*

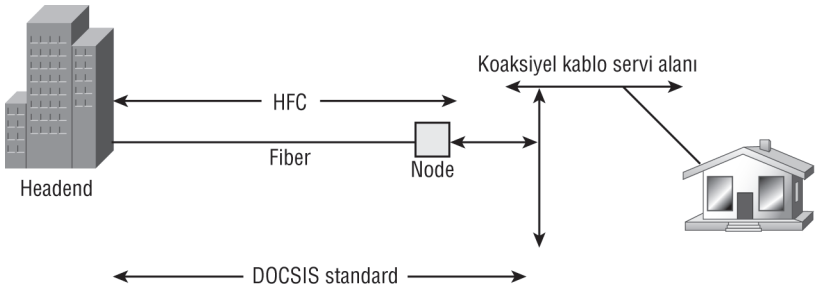
Aşağıda, birkaç kablo network terimi bulabilirsiniz:

**Headend:** Headend; kablo sinyallerinin alındığı, işlemden geçtiği ve formatlandığı yerdir. Sinyaller daha sonra headend'den, distribution network'e aktarılır.

**Distribution network:** Burası, 100'den 2000 müşteriye kadar aralıktaki nispeten küçük servis alanlarıdır. Tipik olarak distribution ağın trunk bölümünün fiber optik ile değiştirildiği, fiber-koaksiyel veya HFC mimarisinin karışımıdırlar. Fiber, headend ve bir optik düğümden bağlantıyı şekillendirir. Işığı radyo frekansına (RF) dönüştürür ve sonra servis verilen belirli bir alan boyunca, koaksiyel bir kablo yardımıyla dağıtılır.

**DOCSIS (data over cable service interface specification):** Tüm kablo modemler ve benzer cihazlar, bu standarda uymak zorundadır.

Şekil 14.3, farklı ağ türlerini nerede bulacağınızı ve listelediğim terimlerin, bir ağda nasıl kullanılacağını göstermektedir.




***Şekil 14.3:*** *Kablo ağı ve terimleri.*

Problem şudur; ISP'ler genelde, kablo operatörünün ana headend'inden yayılan bir fiber ağı kullanır, hatta bazen bölgesel headend'ler bile olabilir. Bir semtin hubsite'ına, sonra da fiber optik düğüme ulaşır ki, 25'den 2,000'e veya daha fazla eve servis sağlayabilir. (Beni yanlış anlamayın, bütün linklerin problemleri vardır, kabloya takmış değilim.)

Şu da başka bir problemdir: Kablo'nuz varsa, PC'nizin komut istemcisini açın ve `ipconfig` yazın, ardından subnet mask'ınızı kontrol edin. O muhtemelen /20 veya /21 klas B adresidir. Bunun, kablo network bağlantısı başına 4,906 veya 2,048 host anlamına geldiğini biliyorsunuz. Bu da çok iyi değildir.

Kablo dediğimizde transmisyon için koaksiyel kablo kullandığımızı kastediyoruz. Şimdi, abonelere düşük maliyetli yayın önerme anlamında, CATV veya ortak antenli televizyon kullanılmaktadır. Kablo, çok yüksek para ödemeden ses, veri ve analog ile dijital video sağlayabilmektedir.

Ortalama kablo bağlantınız, size maksimum 2Mbps yükleme hızı sağlar. Ve bunu diğer abonelerle paylaşmak zorunda olduğunuzu hatırlayın. Bu yeterli değilse, aşırı yüklü web sunucuları ve eski NET tıkanıklıkları gibi faktörler de vardır. Email kontrol eden komşularınız, bunu çok fazla değiştirmez. Öyleyse, kim veya nedir sıkıntı? Eğer çevrimiçi bir oyuncuysanız, en yüksek periyotlarda bir miktar gecikme fark edeceksiniz. Komşularınızdan birisi korsan Star Wars filmlerinin tüm koleksiyonu gibi, büyük boyutta bir veri gönderiyorsa bu tüm bağlantının en yüksek sınırına ulaşabilir ve herkesin browser'ı durma noktasına gelebilir.

Kablo modem erişimi, DSL kurulumundan daha hızlı veya kolay olabilir de olmayabilir de. Ve erişiminiz, nerede yaşadığınıza ve diğer farklı faktörlere bağlı olarak değişiklik gösterebilir. Genelde daha kullanışlı ve biraz daha ucuz olması, onun burun farkıyla kazanmasını sağlayacaktır. Fakat üzülmeyin, şayet kablo erişimi, semtinizde yoksa DSL de iyidir (her şey dial-up'tan daha iyidir).

## Digital Subscriber Line (DSL)

Size yüksek hızda veri aktarımı sağlayan, bakır telefon kablosu kullanan bir teknolojidir. DSL, bir telefon hattı, bir DSL modem (genelde servis içeren), bir Ethernet kartı veya Ethernet bağlantısı olan bir router ve bulunduğunuz yere servis sağlayabilecek bir ISP gerektirir.

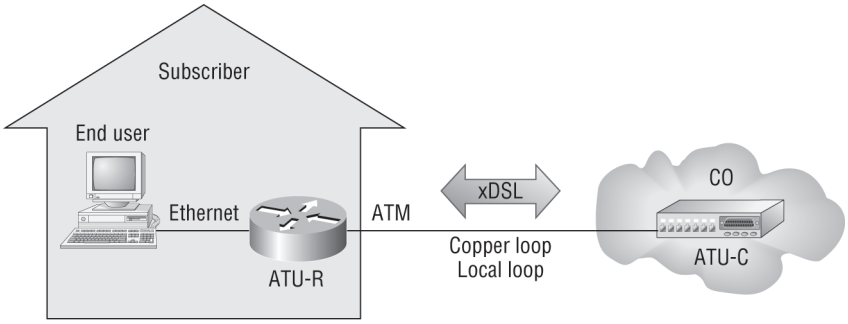
DSL'in orijinal olarak anlamı digital subscriber loop'tur. Fakat anlamı şimdi digital subscriber line olarak değişmiştir. DSL grup tipleri, upstream ve downstream hız bağlantıları temelinde iki kategoriye ayrılmıştır:

**Simetrik DSL:** Downstream ve upstream bağlantıların hızı eşit ve simetriktir.

**Asimetrik DSL:** Bir ağın iki ucu arasında farklı aktarım hızları olur. Downstream hızı genellikle daha yüksektir.

Şekil 14.4, verinin bakır teller üzerinde dolaştığı aktarım teknolojisi olan xDSL kullanan bir ev kullanıcısının ortalamasını göstermektedir.

xDSL terimi, ADSL, high-bit-rate DSL (HDSL), Rate Adaptive DSL (RADSL), Synchronous DSL (SDSL), ISDN DSL (IDSL) ve very-high-data-rate DSL (VDSL) gibi bir DSL değişim numarası içerir.




Tüm DSL tipleri, katman 1 teknolojisidir.
ATU-R = ADSL Transmission Unit - Remote
ATU-C = ADSL Transmission Unit - Central)

***Şekil 14.4:*** *Ev kullanıcısından, Central Office'e (CO) xDSL bağlantısı.*

ADSL ve VDSL gibi, ses frekans bandını kullanmayan DSL çeşitleri, DSL hatlarının hem veri hem de ses sinyallerinin eş zamanlı taşınmasına izin verirler. SDSL ve IDSL gibi, tüm frekans aralığını kapsayan diğerleri, sadece veri taşırlar. DSL bağlantısının size sağladığı veri servisi daima açıktır.

DSL servisinin hızı, CO'dan ne kadar uzakta olduğunuza bağlı olarak önerilebilir. (ne kadar yakın olursa o kadar iyidir). Gerçekten, fiziksel olarak yeterince yakınsanız, 6.1Mbps civarında bir hıza ulaşabilirsiniz.

## ADSL

ADSL, aynı anda hem veri hem sesi destekler. Video, film ve müzik indirme, online oyun oynama, sörf yapma ve bazı büyük eklentiler içeren email'ler alma amacıyla daha fazla indirme bant genişliğine ihtiyaç duyan yerleşik aboneler için oldukça iyi olduğundan, downstream için, upstream'dan daha fazla bant genişliği ayrılacak şekilde tasarlanmıştır. ADSL, 256Kbps ile 8Mbps arasında bir downstream hızı sağlayacaktır. Fakat upstream için, 1Mbps civarında bir hıza ulaşabilirsiniz.

POTS, analog ses aktarımı için bir kanal sağlar ve aynı sarmal-çift telefon hattı üzerinden ADSL ile bir sorun olmadan aktarabilir. Aslında, ADSL'in türüne bağlı olarak, sadece iki değil, yaygın olarak üç bilgi kanalı, aynı anda, aynı kabloyu kullanır. Bu sebeple insanlar, bir telefon hattını ve bir ADSL bağlantısını aynı anda kullanabilirler ve her iki servis de birbirinden etkilenmeyecektir.

ATM, DSLAM olarak bilinen, DSL interface kartları ( ATU-C'ler) içeren bir ATM switch'te sonlandırılan CPE'den, DSL katman1 bağlantısı boyunca kullanılan Data Link katmanı protokolüdür. ADSL bağlantıları, DSLAM'daki uçlarıyla buluştuktan sonra, veriyi bir ATM ağı üzerinden, aggregation router denilen, abonenin IP bağlantısının sona erdiği katman 3 cihaza anahtarlar.

Şimdi enkapsülasyonun ne kadar önemli olduğunu biliyorsunuz. Tahmin edebileceğiniz gibi, bir ATM boyunca IP paketlerinin ve DSL bağlantısının buna sahip olması gerekmektedir. Bu, interface tipiniz ve servis sağlayıcının switch'ine bağlı olarak, şu üç yoldan biriyle gerçekleşir:

**PPPoE:** Bu, sonraki bölümde detaylı şekilde incelenecektir.

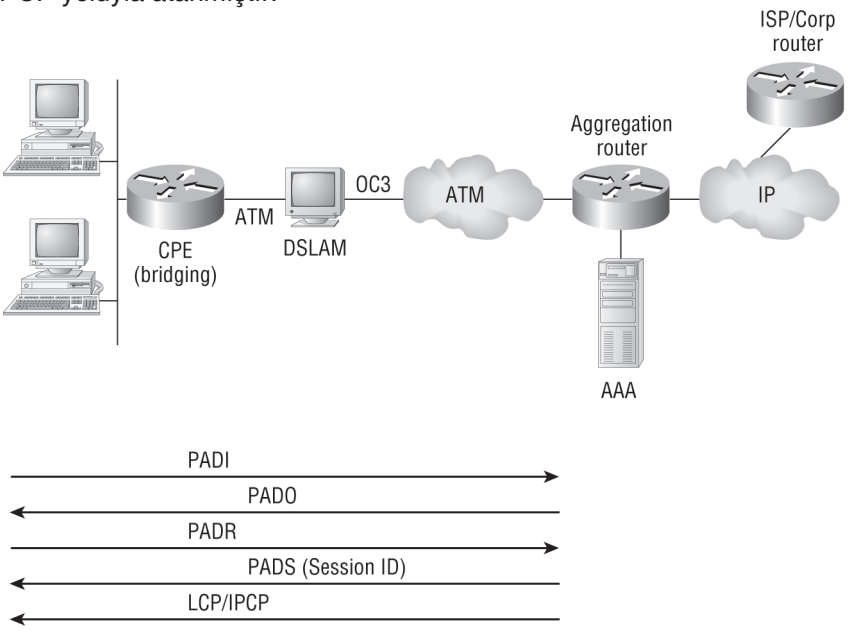
**RFC1483 Routing:** RFC1483, bir ATM ağı boyunca connectionless network trafiğini taşımak için iki farklı yöntem açıklar: routed protokoller ve bridged protokoller.

**PPPoA:** Point-to-Point Protocol (PPP) over ATM, PPP frame'lerinin, ATM AAL5'de (ATM Adaptation 5) enkapsüle edilmesi için kullanılır. Tipik olarak kablo modemler, DSL ve ADSL servisleri ile kullanılır ve authentication, encryption ve compression gibi genel PPP özellikleri sunar. Aslında, PPPoE ile karşılaştırılınca, daha az ek yüke sahiptir.

**PPPoE:**

ADSL servisleri ile birlikte kullanılan PPPoE (Point-to-Point Protocol over Ethernet), Ethernet frame'lerindeki PPP frame'lerini enkapsüle eder ve authentication, encryption ve compression gibi genel PPP özelliklerini kullanır. Fakat daha önce söylediğim gibi kötü yapılandırılmış bir firewall'unuz varsa, o tam bir sorundur. Bu, diğer Ethernet cihazlarıyla bağlantı kurmak için kullanabilecekleri ve IP paketlerini taşımak için bir noktadan-noktaya bağlantı başlatabilecekleri şekilde, bir PPP linkinin özellikleriyle, PPP boyunca çalışan diğer protokolleri ve IP'yi katmanlarına ayıran bir tünelleme protokolüdür.

Şekil 14.5, PPPoE'nin, ADSL boyunca tipik kullanımını göstermektedir. Görebileceğiniz gibi, bir PPP oturumu, son kullanıcının PC'sinden, router'a bağlanmıştır ve abone PC'nin IP adresi, router tarafından, IPCP yoluyla atanmıştır.




***Şekil 14.5:*** *ADSL ile PPPoE.*

PPPoE, özel PPP tabanlı yazılımları seri hat kullanılmayan bir bağlantıyla başa çıkabilme yeteneğiyle donatmak için kullanılır. Ayrıca evde, paket-tabanlı Ethernet gibi network ortamlarında ve internet bağlantı hesabı oluşturmak için login ve password ile özel bir bağlantı sağlamak için kullanılır. Diğer bir faktör de, linkin diğer tarafındaki IP adresinin verilmesi ve PPPoE bağlantısının açık olduğu belirli zaman diliminde erişilebilir olmasıdır. Böylece, IP adresinin yeniden kullanılmasına izin verilir.

PPPoE, bir keşif aşamasına ve PPP oturum aşamasına (RFC2516'ya bakın) sahiptir. Şu şekilde çalışır: İlk olarak istemci makinelerin isteklerinin ihtiyaçlarının karşılaması için, en uygun sunucuyu belirleyebilmesi için bir keşif prosesi çalıştırmak zorundadır. Bu esnada host, bir PPPoE

oturumu başlatır. Bundan sonra, karşı cihazın Ethernet MAC adresini belirlemek ve bir PPPoE oturum ID'si oluşturmak zorundadır. PPP, peer-to-peer ilişkileri sınırlasa da, keşif bölümü aslında bir istemci/sunucu ilişkisidir.

Seri bağlantılara geçmeden önce, işlemek istediğim son bir şey daha var: Cisco LRE.

### Cisco Long Range Ethernet (LRE)

Cisco Long Range Ethernet çözümü, Ethernet servis kapasitesini önemli şekilde artırmak için VDSL (Very High Data Rate Digital Subscriber Line) teknolojisi denilen bir takım prosesler çalıştırır. Ve LRE şu etkileyici sonuca ulaşabilir: Mevcut sarmal-çift kablo boyunca yaklaşık 1.5km. uzaklığa, 5'den, 15Mbps'a kadar hız (full duplex).

Aslında Cisco LRE teknolojisi POTS, dijital telefon ve ISDN trafik hatlarında broadband servisi sağlar ve ayrıca ADSL teknolojileriyle uygun modlarda çalışabilir. Bu esneklik önemlidir çünkü onu broadband servislerinin zaten olduğu, fakat geliştirilmesi gereken binalarda ve/veya yapılarda servis sağlayıcılar için LRE'yi uygun hale getirir.

## Serial Wide Area Network'leri (WAN) Kablolamak

Hayal edebileceğiniz gibi, her şeyin iyi gittiğine emin olmak için WAN'ınızı bağlamadan önce bilmeniz gereken birkaç şey var. İlk olarak, hem Cisco'nun sağladığı WAN Physical katman kurulum türünü anlamanız hem de çeşitli WAN serial konnektör türlerine aşina olmanız gerekir.

Cisco serial bağlantıları neredeyse tüm WAN tiplerini destekler. Tipik WAN bağlantınız, 45Mbps (T3) hız ile HDLC ve PPP ve Frame Relay kullanan dedicated leased line'dır.

HDLC, PPP ve Frame Relay, aynı Physical katman düzenlemesini kullanabilir. Çeşitli bağlantı tiplerini inceleyeceğim ve sonra CCNA konularında belirtilmiş WAN protokollerinden bahsedeceğim.

### Seri Aktarım

WAN seri konnektörleri, tek bir kanaldan, her seferinde 1 bit'in yer aldığı seri aktarım kullanır.

*Paralel aktarım, her seferinde en az 8 bit geçirebilir. Fakat tüm WAN'lar seri aktarım kullanır.*

NOT

Cisco router'lar, Cisco'dan veya Cisco ekipman sağlayıcıdan almanız gereken 60 pinli özel bir seri konnektör kullanır. Cisco ayrıca, 60 pinli basit seri kablonun onda biri (1/10) boyutta, smart-serial denilen yeni bir özel bağlantıya sahiptir. Bu kablo konnektörünü kullanmadan önce, router'ınızda doğru interface tipine sahip olduğunuza emin olmalısınız.

Kablonun diğer ucunda sahip olduğunuz konnektör tipi, servis sağlayıcınıza ve onların kenar-cihaz ihtiyaçlarına bağlıdır. Karşılaşacağınız farklı sonlandırma türleri vardır:

- EIA/TIA-232
- EIA/TIA-449
- V.35 (bir CSU/DSU'ya bağlanmak için kullanılır)
- EIA-350

Şunları netleştirdiğinize emin olun: Seri linkler, frekans (hertz) olarak belirtilir. Bu frekanslarda taşınabilen veri miktarı, bant genişliği olarak tanımlanır. Bant genişliği, seri kanalın taşıyabileceği saniyedeki veri miktarının bit olarak değeridir.

### Data Terminal Equipment (DTE) ve Data Communication Equipment (DCE)

Varsayılan olarak router interface'leri, Data Terminal Equipment (DTE)'dir ve onlar, channel service unit/data service unit (CSU/DSU) gibi Data Communication Equipment'lara (DCE) bağlanırlar. CSU/DSU, sonra bir demarcation lokasyonuna (demarc) bağlanır ve servis sağlayıcının son

sorumluluk noktasıdır. Çoğu zaman demarc, telekom dolabında yer alan, RJ-45 (8-pin modüler) dişi konektöre sahip bir fiştir.

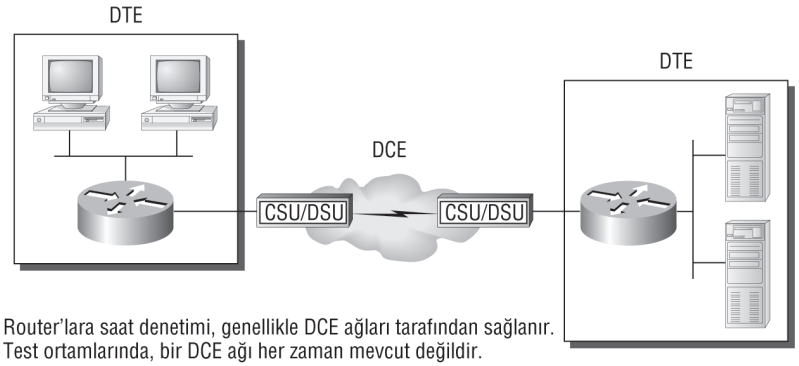
Aslında demarc'ları zaten duymuş olabilirsiniz. Şayet bir problem için servis sağlayıcınızı ararsanız, demarc'a kadar tüm testlerin normal olduğunu söyleyeceklerdir. Böylece problemin CPE'de olması gerekir. Yani, problem onlarla değil sizinle ilgilidir.

Şekil 14.6 tipik bir DTE-DCE–DTE bağlantısını ve ağda kullanılan cihazları göstermektedir.

Bir WAN'ın arkasındaki düşünce iki DTE ağının, bir DCE ağı üzerinden bağlanmasıdır. DCE ağları, diğer uçtaki CSU/DSU'ya kadar servis sağlayıcının kablolaması ve switch'leri boyunca CSU/DSU'yu içerir. Ağın DCE cihazı (CSU/DSU), DTE-bağlı interface'lere (router'ın seri interface'lerine) saat denetimi sağlar.

Belirtildiği gibi DCE ağları router'a saat denetimi sağlar, bu CSU/DSU'dur. Şayet lab ortamındaysanız, WAN çapraz kablosu kullanıyorsanız ve CSU/DSU'ya sahip değilseniz, kablonun sonundaki, DCE'de, size bölüm 4'te gösterdiğim, `clock rate` komutunu kullanarak saat denetimi sağlayabilirsiniz.

**NOT** *EIA/TIA-232, V.35, X.21 ve HSSI (High-Speed Serial Interface) gibi terimler, DTE (router) ve DCE cihazı (CSU/DSU) arasındaki physical katmanı açıklar.*




***Şekil 14.6:*** *DTE-DCE-DTE WAN bağlantısı.*

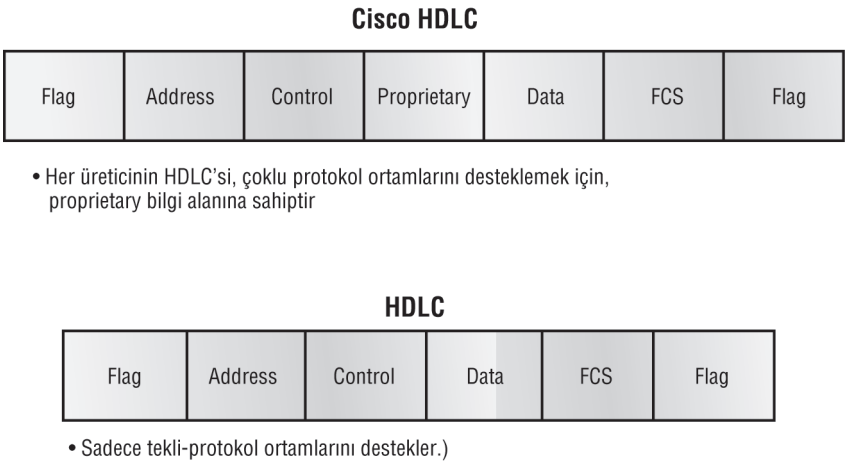
# High-Level Data-Link Control (HDLC) Protocol

High-Level Data-Link Control (HDLC) protokolü; popüler ISO standardı, bit-oriented, Data Link katmanı protokolüdür. Frame özellikleri ve checksum'lar kullanan senkron seri veri linklerindeki veri için bir enkapsülasyon yöntemi belirtir. HDLC, leased line'larda (kiralık hatlarda) kullanılan bir noktadan-noktaya protokoldür. HDLC ile authentication kullanılmayabilir.

Byte-temelli protokollerde kontrol bilgisi tamamı byte kullanılarak şifrelenir. Diğer taraftan bit-temelli protokoller, kontrol bilgisini göstermek için tek bit kullanır. Bazı yaygın bit-oriented protokoller SDLC, LLC, HDLC ve IP'dir.

HDLC, Cisco router'lar tarafından senkron seri linklerde kullanılan varsayılan enkapsülasyondur. Cisco HDLC, tescillidir ve diğer üreticilerin HDLC kurulumlarıyla haberleşemezler. Fakat bu Cisco için çok kötü değildir. Çünkü herkesin HDLC uygulaması, kendine özgüdür. Şekil 14.7 Cisco HDLC formatını göstermektedir.

Şekilde görüldüğü gibi her üreticinin kendine has HDLC enkapsülasyon yöntemi olmasının sebebi, her üreticinin çoklu Network katman protokollerini enkapsüle etmek amacıyla HDLC protokolü için farklı bir yol kullanmasıdır. Üretici, farklı katman3 protokolleriyle haberleşmek amacıyla HDLC için bir yola sahip değilse, o zaman HDLC sadece bir protokol taşıyabilecektir. Bu özel başlık, HDLC enkapsülasyonun veri alanında yer alır.



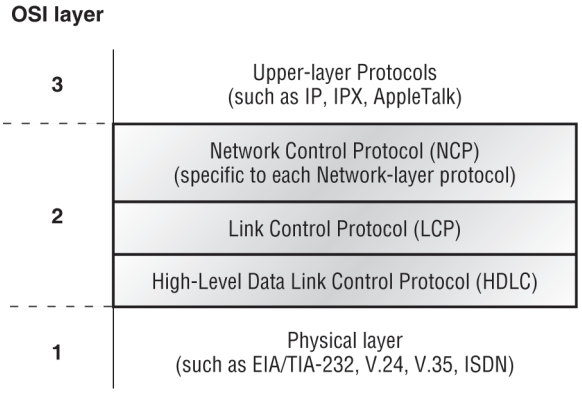

***Şekil 14.7:*** *Cisco HDLC frame formatı.*

Sadece Cisco router'a sahip olduğunuzu ve Cisco olmayan bir router'a bağlamanız gerektiğini varsayalım. Ne yaparsınız? Çalışmayacağı için varsayılan HDLC seri enkapsülasyonu kullanamazsınız. Yerine, PPP gibi üst katman protokolleri belirlemenin ISO standardını kullanabilirsiniz. PPP'nin kaynakları ve standartlarıyla ilgili daha fazla bilgi için RFC 1661'i inceleyebilirsiniz. Gelin detaylı olarak PPP'yi ve PPP enkapsülasyon kullanarak router'ların nasıl bağlanacağını görelim.

# Point-to-Point Protocol (PPP)

Point-to-Point Protocol'e (PPP) biraz zaman harcayalım. Onun, asenkron serial (dial-up) veya senkron serial (ISDN) ortam aracı boyunca kullanılabilen bir Data Link katman protokolü olduğunu hatırlayın. Data-link bağlantılarını oluşturmak ve sürdürmek için, Link Control Protocol (LCP) kullanır. Network Control Protocol (NCP), bir noktadan-noktaya bağlantıda kullanılması için, çoklu Network katman protokollerine (routed protokolleri) izin vermek için kullanılmaktadır.

HDLC, Cisco seri linklerinde varsayılan seri enkapsülasyon olduğu ve oldukça iyi çalıştığı halde, neden veya ne zaman PPP kullanımını seçeriz? PPP'nin esas amacı, katman3 paketlerini, bir Data Link katmanında noktadan-noktaya link boyunca taşımaktır ve marka bağımlı değildir. Bundan dolayı, tamamıyla Cisco router'lara sahip olmadığınız sürece, seri interface'lerinizde, PPP'ye ihtiyacınız vardır. HDLC enkapsülasyonun, Cisco tescilli olduğunu hatırladınız, değil mi? Artı, PPP çeşitli katman3 protokollerini enkapsüle ettiğinden ve authentication, dinamik adresleme ve callback, sağladığından, PPP, sizin için, HDLC yerine en iyi enkapsülasyon çözümü olabilir.




***Şekil 14.8:*** *Noktadan-Noktaya Protokol yığını.*

Şekil 14.8, OSI referans modeli ile karşılaştırılan, protokol yığınını göstermektedir.

PPP, dört ana bileşene sahiptir:

**EIA/TIA-232-C, V.24, V.35 ve ISDN:** Seri haberleşme için bir Physical katman uluslararası standardı.

**HDLC:** Seri linkler boyunca datagram'ları enkapsüle etmenin bir yöntemi.

**LCP:** Noktadan-noktaya bağlantıyı kurma, yapılandırma, sürdürme ve sonlandırma yöntemi.

**NCP:** Farklı Network katman protokollerini kurma ve yapılandırma yöntemi. NCP, çoklu Network katman protokollerinin eş zamanlı kullanımına izin vermek için tasarlanmıştır. Buradaki bazı protokol örnekleri, IPCP (Internet Protocol Control Protocol) ve IPXCP'dir (Internetwork Packet Exchange Control Protocol).

PPP protokol yığınının, sadece Physical ve Data Link katmanında belirtildiğini kafanıza kazıyın. NCP, bir PPP veri linki boyunca protokollerin enkapsüle edilerek, çoklu Network katman protokollerin haberleşmesine izin vermesi için kullanılmaktadır.

NOT *Cisco ve Cisco olmayan bir router, seri bir bağlantıya sahip olursa, HDLC, varsayılan olarak çalışmayacağından, PPP veya Frame Relay gibi başka enkapsülasyon yöntemi yapılandırmanız gerektiğini hatırlayın.*

## Link Control Protocol (LCP) Yapılandırma Seçenekleri

Link Control Protocol (LCP), aşağıdakileri içeren farklı enkapsülasyon seçenekleri sunar:

**Authentication:** Bu seçenek, linkin arayan tarafına kullanıcıyı tanımlayan bilgiyi göndermesini söyler. İki yöntem, PAP ve CHAP'tır.

**Compression:** Bu, aktarımdan önce yükü veya veriyi sıkıştırarak PPP bağlantılarının throughput'unu artırmak için kullanılır. PPP, veri frame'ini alınan uçta tekrar eski haline getirir.

**Error detection:** PPP güvenli, döngü olmayan veri linki sağlamak için Quality and Magic Number seçeneği kullanır.

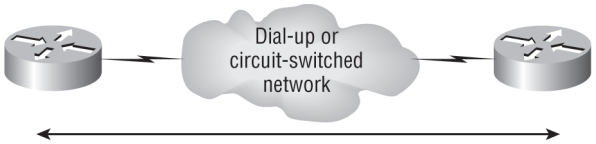
**Multilink:** IOS versiyonu 11.1'den başlayarak multilink, Cisco router'larla PPP'de desteklenmektedir. Bu seçenek, ayrı bazı fiziksel yolların, katman3'te tek mantıksal yol olarak görünmesini sağlar. Örneğin, PPP multilink çalışan iki T1, bir katman3 routing protokolüne, tek bir 3Mbps yol olarak görünecektir.

**PPP callback:** PPP, başarılı authentication sonrası tekrar arama yapması için yapılandırılabilir. PPP callback, erişim ücretleri bazında kullanımı izlemek, hesap kayıtları için ve diğer birçok sebepten dolayı, sizin için iyi bir özellik olabilir. Callback'i etkin kılarak, aranan bir router (istemci), uzak bir router (sunucu) ile bağlantı kuracaktır ve daha önce açıkladığım gibi kimlik doğrulaması yapacaktır. (Bunun çalışması için her iki router'ın da callback özelliği ile yapılandırılması gerektiğini bilin). Authentication tamamlanınca uzak router, bağlantıyı sonlandıracaktır ve sonra uzak router'dan arayan router'a yeniden bir bağlantı başlatacaktır.

NOT *PPP callback'inizde Microsoft cihazlar varsa Microsoft, IOS 11.3(2)T ve sonrasının desteklediği, Microsoft Callback Control Protocol (CBCP) olarak bilinen tescilli bir callback kullanır.*

## PPP Oturum Kurulumu

PPP bağlantıları başladığında, Şekil 14.9'da gösterildiği gibi linkler üç fazlı oturum kurulumundan geçer.




PPP Oturum Kurulumu
1. Link kurulum fazı.
2. Authentication fazı (seçmeli)
3. Network katmanı protokol fazı.

***Şekil 14.9:*** *PPP oturum kurulumu.*

**Link kurulum fazı:** Linki yapılandırmak ve test etmek için her PPP cihazı tarafından, LCP paketleri gönderilir. Bu paketler, her cihazın compression, authentication ve verinin boyutunu görmesine izin veren Configuration Option denilen bir alan içerir. Eğer Configuration Option alanı yoksa varsayılan yapılandırma kullanılacaktır.

**Authentication fazı:** Şayet gerekirse bir linkte kimlik doğrulaması için CHAP ve PAP kullanılabilir. Authentication, Network katman protokolü bilgisi okunmadan önce yer alır. Ve link-kalite belirlemesiyle, eş zamanlı olması muhtemeldir.

**Network katmanı protokol fazı:** PPP, Network katman protokollerinin, enkapsüle edilmesine ve bir PPP veri linki boyunca gönderilmesini sağlamak için Network Control Protocol (NCP) kullanır. Her Network katman protokolü (IP, IPX, AppleTalk vs, routed protokolleri), NCP ile bir servis kurar.

## PPP Authentication Yöntemleri

PPP linkleriyle kullanabileceğiniz iki kimlik doğrulama yöntemi vardır:

**Password Authentication Protocol (PAP):** Password Authentication Protocol (PAP), iki yöntemden daha az güvenli olanıdır. Şifreler, clear text gönderilir ve PAP sadece ilk link kurulumunda çalışır. PPP linki ilk olarak kurulduğunda kimlik doğrulaması onaylanana kadar, uzak düğüm ilk gönderen router'a kullanıcı adı ve şifresi gönderir.

**Challenge Handshake Authentication Protocol (CHAP):** Challenge Handshake Authentication Protocol (CHAP), bir linkin ilk başlamasında ve router'ın, aynı host'la hala haberleştiğinden emin olmak için linkte periyodik kontrollerde kullanılır.

PPP, ilk link kurulum fazını tamamladıktan sonra lokal router, uzak cihaza bir istek gönderir. Uzak cihaz, MD5 denilen tek-yönlü hash fonksiyonu kullanarak hesaplanan bir değer gönderir. Lokal router, eşleştiğinden emin olmak için bu hash değerini kontrol eder. Şayet değer eşleşmezse link hemen sonlandırılır.

## Cisco Router'larda PPP'yi Yapılandırmak

Bir interface'de PPP enkapsülasyon yapılandırmak, oldukça basittir. Onu CLI'dan yapılandırmak için şu basit router komutlarını kullanın:

```
Router#config t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#int s0
Router(config-if)#encapsulation ppp
Router(config-if)#^Z
Router#
```

Çalışması için seri bir hatta bağlı her iki interface'de, PPP enkapsülasyonun etkinleştirilmesi gerekir ve `help` komutu yardımıyla, bazı ilave yapılandırma seçenekleri vardır.

## PPP Authentication Yapılandırmak

Seri interface'inizi, PPP enkapsülasyon desteklemesi için yapılandırdıktan sonra PPP kullanarak router'lar arasında kimlik doğrulaması yapılandırabilirsiniz. İlk olarak, router'ın hostname'ini ayarlamanız gerekir. Sonra, router'ınıza bağlanacak uzak router için kullanıcı adı ve şifre ayarlayın:

İşte bir örnek:

```
Router#config t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#hostname RouterA
RouterA(config)#username RouterB password cisco
```

`Hostname` komutunu kullandığınızda kullanıcı adının, router'ınıza bağlanan uzak router'ın hostname'i olduğunu hatırlayın. O ayrıca büyük/küçük harf duyarlıdır. Her iki router'daki şifre de aynı olmalıdır. Şifre `show run` komutu ile görebileceğiniz şekilde clear text'dir. Password'ü, `service password-encryption` komutu kullanarak şifreleyebilirsiniz. Bağlanmayı planladığınız her uzak sistem için yapılandırılmış kullanıcı ismi ve şifresine sahip olmanız gerekir.

Hostname, kullanıcı adı ve şifreleri ayarladıktan sonra CHAP veya PAP'ı kimlik doğrulama yöntemi olarak seçin:

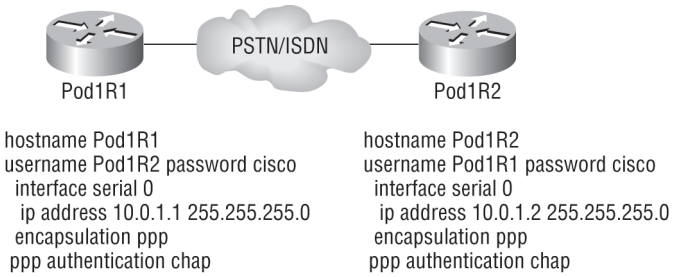
```
RouterA#config t
Enter configuration commands, one per line. End with CNTL/Z.
RouterA(config)#int s0
RouterA(config-if)#ppp authentication chap pap
RouterA(config-if)#^Z
RouterA#
```

NOT *Bu bölümde daha sonra SDM kullanarak PPP'de authentication ayarlayacağım. CLI en kolay yöntemdir.*

Şayet aynı hatta her iki yöntemi de yapılandırırsanız, burada görüldüğü gibi, link görüşmesi esnasında, sadece ilk yöntem kullanılacaktır. İkinci yöntem, ilk yöntemin başarısız olması durumunda, bir yedek olarak davranır.

## PPP Enkapsülasyonu Doğrulamak

Şimdi PPP enkapsülasyon aktiftir. Onun çalışır olduğunun nasıl doğrulanacağını göstermeme izin verin. İlk olarak, bir ağ örneğinin şekline bakalım. Şekil 14.10 noktadan-noktaya seri veya ISDN bağlantısıyla bağlı iki router'ı göstermektedir.




***Şekil 14.10:*** *PPP kimlik doğrulama örneği.*

Yapılandırmayı, `show interface` komutu kullanarak doğrulamaya başlayabilirsiniz:

```
Pod1R1#sh int s0/0
Serial0/0 is up, line protocol is up
  Hardware is PowerQUICC Serial
  Internet address is 10.0.1.1/24
  MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec,
     reliability 239/255, txload 1/255, rxload 1/255
  Encapsulation PPP
  loopback not set
  Keepalive set (10 sec)
  LCP Open
  Open: IPCP, CDPCP
[output cut]
```

Altıncı satırın, enkapsülasyonu PPP olarak listelediğine ve sekizinci satırın, LCP'nin açık olduğunu gösterdiğine dikkat edin. Yani, oturum kurmayı müzakere etmektedir ve her şey iyi gitmektedir. Dokuzuncu satır bize NCP'nin, IP ve CDP protokolleri için dinlemede olduğunu söylemektedir.

Her şey mükemmel olmasaydı, ne görecektim? Şekil 14.11'de gösterilen yapılandırmayı yazacağım ve anlayacağım.


```
hostname Pod1R1
username Pod1R2 password Cisco
 interface serial 0
  ip address 10.0.1.1 255.255.255.0
 encapsulation ppp
ppp authentication chap
```

```
hostname Pod1R2
username Pod1R1 password cisco
 interface serial 0
  ip address 10.0.1.2 255.255.255.0
 encapsulation ppp
ppp authentication chap
```

***Şekil 14.11:*** *Başarısız PPP kimlik doğrulaması.*

Buradaki problem nedir? Kullanıcı adı ve şifrelere bir bakalım. Router PodR1'in yapılandırmasında bulunan Pod1R2 username komutundaki ***C***, büyük harfle yazılmıştır. Kullanıcı adı ve şifrelerin, büyük/küçük harf duyarlı olmasından dolayı bu bir yanlıştır. Gelin `show interface` komutuna bakalım ve neler olduğunu anlayalım:

```
Pod1R1#sh int s0/0
Serial0/0 is up, line protocol is down
  Hardware is PowerQUICC Serial
  Internet address is 10.0.1.1/24
  MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec,
     reliability 243/255, txload 1/255, rxload 1/255
  Encapsulation PPP, loopback not set
  Keepalive set (10 sec)
  LCP Closed
  Closed: IPCP, CDPCP
```

İlk olarak, çıktının ilk satırındaki `Serial0/0 is up, line protocol is` down'a dikkat edin. Uzak router'dan keepalives gelmemektedir. Sonra, kimlik doğrulaması başarısız olduğundan, LCP'nin kapalı olduğuna dikkat edin.

## PPP Kimlik Doğrulamasını Debug etme

Ağdaki iki router arasında olan CHAP authentication'ı görüntülemek için `debug PPP authentication` komutunu kullanın.

PPP enkapsülasyon ve kimlik doğrulama, her iki router'da da doğru yapılandırıldıysa ve kullanıcı adı ve şifreleriniz de doğruysa `debug PPP authentication` komutu şöyle bir çıktı görüntüleyecektir:

```
d16h: Se0/0 PPP: Using default call direction
1d16h: Se0/0 PPP: Treating connection as a dedicated line
1d16h: Se0/0 CHAP: O CHALLENGE id 219 len 27 from "Pod1R1"
1d16h: Se0/0 CHAP: I CHALLENGE id 208 len 27 from "Pod1R2"
1d16h: Se0/0 CHAP: O RESPONSE id 208 len 27 from "Pod1R1"
1d16h: Se0/0 CHAP: I RESPONSE id 219 len 27 from "Pod1R2"
1d16h: Se0/0 CHAP: O SUCCESS id 219 len 4
1d16h: Se0/0 CHAP: I SUCCESS id 208 len 4
```

Daha önce Şekil 14.11'deki gibi PPP kimlik doğrulamasının başarısız olduğu örnekteki gibi, yanlış bir kullanıcı adına sahipseniz çıktı şu şekilde görünecektir:

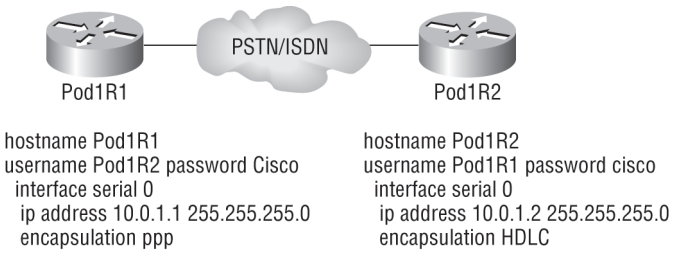
```
1d16h: Se0/0 PPP: Using default call direction
1d16h: Se0/0 PPP: Treating connection as a dedicated line
1d16h: %SYS-5-CONFIG_I: Configured from console by console
1d16h: Se0/0 CHAP: O CHALLENGE id 220 len 27 from "Pod1R1"
```

```
1d16h: Se0/0 CHAP: I CHALLENGE id 209 len 27 from "Pod1R2"
1d16h: Se0/0 CHAP: O RESPONSE id 209 len 27 from "Pod1R1"
1d16h: Se0/0 CHAP: I RESPONSE id 220 len 27 from "Pod1R2"
1d16h: Se0/0 CHAP: O FAILURE id 220 len 25 msg is "MD/DES compare
failed"
```

CHAP kimlik doğrulamasıyla PPP, üç-yönlü bir kimlik doğrulamadır ve kullanıcı adı ile şifreler olması gereken şekilde tamamıyla aynı yapılandırılmadıysa, kimlik doğrulaması başarısız olacaktır ve link çalışmayacaktır.

## Eşleşmeyen WAN Enkapsülasyonları

Noktadan-noktaya bir linkiniz varsa ve enkapsülasyonlar aynı değilse, link asla çalışmayacaktır. Şekil 14.12, PPP ile bir link ve HDLC ile başka bir linki göstermektedir.




***Şekil 14.12:*** *Eşleşmeyen WAN enkapsülasyonları.*

Bu çıktıdaki, router Pod1R1'e bakın:

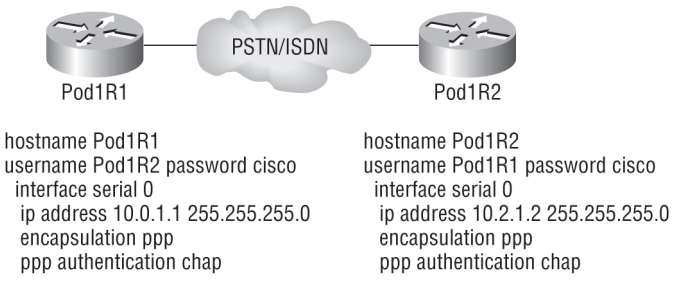
```
Pod1R1#sh int s0/0
Serial0/0 is up, line protocol is down
  Hardware is PowerQUICC Serial
  Internet address is 10.0.1.1/24
  MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec,
     reliability 254/255, txload 1/255, rxload 1/255
  Encapsulation PPP, loopback not set
  Keepalive set (10 sec)
  LCP REQsent
Closed: IPCP, CDPCP
```

Seri interface down'dır ve LCP istek göndermektedir, fakat herhangi bir cevap almayacaktır, çünkü Pod1R1 router HDLC enkapsülasyon kullanmaktadır. Bu sorunu çözmek için Pod1R2 router'ına gitmek ve seri interface'inde PPP enkapsülasyon yapılandırmanız gerekir. Bir şey daha var; kullanıcı adları yapılandırıldığı ve yanlış olduğu halde, seri interface altında `ppp authentication chap` komutu kullanılmadığından, bu sorun olmaz. Username komutunun bu örnekte bir önemi yoktur.

**NOT** *Bir tarafta PPP, diğer tarafta HDLC'ye sahip olamayacağınızı daima hatırlayın. Onlar anlaşamayacaklardır!*

## Eşleşmeyen IP adresleri

Karmaşık problemlerden birisi de, seri interface'lerinizde yapılandırılmış HDLC veya PPP olması, fakat IP adreslerinizin yanlış olmasıdır. Interface'ler'in up olmasından dolayı her şey normal görünüyor. Şekil 14.13'e bakalım ve ne demek istediğimi anlayalım? Farklı subnetlerdeki iki router bağlanmaktadır. Router Pod1R1, 10.0.1.1/24 ve router Pod1R2, 10.2.1.2/24 ile.




***Şekil 14.13:*** *Eşleşmeyen IP adresleri.*

Bu asla çalışmayacaktır. Çıktıya bir bakalım:

```
Pod1R1#sh int s0/0
Serial0/0 is up, line protocol is up
  Hardware is PowerQUICC Serial
  Internet address is 10.0.1.1/24
  MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation PPP, loopback not set
  Keepalive set (10 sec)
  LCP Open
  Open: IPCP, CDPCP
```

Fark ettiniz mi? Router'lar arasındaki IP adresleri yanlıştır, fakat linkler çalışıyor görünmektedir. HDLC ve Frame Relay gibi PPP'de bir katman2 WAN enkapsülasyondur ve IP adresleriyle ilgilenmezler. Link up'tır, fakat yanlış yapılandırıldığından bu link boyunca IP'yi kullanamazsınız.

Bu sorunu bulmak ve çözmek için her router'da `show running-config` veya `show interfaces` komutlarını kullanabilirsiniz veya bölüm 5'de öğrendiğiniz `show cdp neighbors` detail komutunu kullanabilirsiniz:

```
Pod1R1#sh cdp neighbors detail
- - - - - - - - - - - - - - - -
Device ID: Pod1R2
Entry address(es):
  IP address: 10.2.1.2
```

Doğrudan bağlı komşuların IP adresine göz atıp doğrulayabilir ve probleminizi çözebilirsiniz.

Frame Relay'e geçmeden önce PPPoE'e bir göz atalım.

## PPPoE Yapılandırması

PPPoE'i destekleyen interface'e sahip bir router'ınız varsa ve interface, bir DSL modeme bağlıysa, router'ı, PPPoE istemcisi olarak yapılandırabilirsiniz. Bu arada bu hizmet için ISP'nizden, destek aldığınızı düşünüyoruz.

Gelin bir router'daki PPPoE istemci yapılandırmasına bir bakalım. Fiziksel interface altında şunlar görünmektedir:

```
R1(config)#int f0/0
R1(config-if)#p?
pppoe  pppoe-client  priority-group
R1(config-if)#pppoe ?
  enable        Enable pppoe
  max-sessions  Maximum PPPOE sessions
R1(config-if)#pppoe enable ?
  group  attach a BBA group
  <cr>
R1(config-if)#pppoe enable group ?
```

```
  WORD    BBA Group name
  global  Attach global PPPoE group
R1(config-if)#pppoe enable group global
R1(config-if)#pppoe-client dial-pool-number ?
  <1-255>  Dialer pool number
R1(config-if)#pppoe-client dial-pool-number 1
!
interface FastEthernet4
 description $ETH-WAN$
 no ip address
 duplex auto
 speed auto
 pppoe enable group global
 pppoe-client dial-pool-number 1
!
```

Bütün bunlardan sonra, fiziksel interface altında sadece iki komut gerekir: `pppoe enable` ve `pppoe-client` komutu. Her ikisi de, henüz bizim oluşturmadığımız mantıksal interface'lerdir.

Router'ınıza PPPoE bağlantısı eklemek için ayrıca bir dialer innterface oluşturmanız gerekir. Bu bir mantıksal interface'dir ve onun altında `ip address negotiated` komutunu ekleyeceğim, böylece bir DHCP adresi alınabilir ve interface'de yapılandırılabilir. DSL modem ile router arasında özel IP adresi kullanıyorsanız, bu interface'e kolayca statik IP adresi de ekleyebilirsiniz.

Şuna bakalım:

```
!
interface Dialer0
 ip address negotiated
 ip mtu 1452
 encapsulation ppp
 dialer pool 1
 dialer-group 1
 ppp authentication chap callin
 ppp chap hostname Todd
 ppp chap password 0 lammle
!
```

Mantıksal interface'in kendisini `dial pool 1` ve `dialer-group 1` komutlarıyla, fiziksel interface ile nasıl ilişkilendirdiğine özellikle dikkat edin.

Son olarak, `ppp authentication` ve `ppp chap` komutları kullanılarak, dialer interface altında, PPP authentication ayarlanır. CLI aracılığıyla bu komutları, global configuration modda kullandım. Fakat bu kurulumda komutları direkt olarak mantıksal interface altında yapılandıracağım.

Bu oldukça kolay bir yapılandırma olmasına rağmen çok güzel çalışır. Birazdan SDM kullanarak, PPPoE'in nasıl yapılandırılacağını göstereceğim.

# Frame Relay

Frame Relay, son 10 yıldır yayılan WAN servislerinin hala en popüler olanlarından bir tanesidir ve bunun içinde iyi bir sebebi vardır: Maliyeti. Bu önemli maliyet faktörünü önemsememe ayrıcalığına sahip çok az network tasarımı veya tasarımcısı vardır.

Varsayılan olarak Frame Relay, non-broadcast multi access (NBMA) ağı olarak sınıflandırılır. Yani, ağ boyunca RIP güncellemeleri gibi broadcast göndermez. Merak etmeyin, bunu en kısa süre içinde inceleyeceğiz.

Frame Relay'in kökeninde, X.25 denilen bir teknoloji vardır ve daha fazla ihtiyaç duyulmayan hata-düzeltme bileşenleri göz ardı edilirken, günümüzün güvenli ve nispeten temiz telekom ağlarıyla ilgili olan X.25 bileşenlerini içerir. Frame Relay, HDLC ve PPP protokollerinden bahsettiğimde öğrendiğiniz basit leased-line ağlardan daha karmaşıktır. Leased-line ağlarını hayal etmek kolaydır, fakat Frame Relay için öyle değildir. Önemli derecede karmaşık ve çok yönlü olabilir. Bundan dolayı, ağ çizimlerinde sık sık, bir bulut olarak gösterilirler. Buna birazdan değineceğim, şimdi Frame Relay kavramına giriş yapacağım ve onun daha basit leased line teknolojilerinden nasıl ayrıldığını göstereceğim.

Bu teknolojiye giriş yapmanızla beraber, Frame Relay'in temellerini güçlü şekilde kavramanız için gereken, tüm yeni terminolojinin sanal bir sözlüğüne sahip olacaksınız.

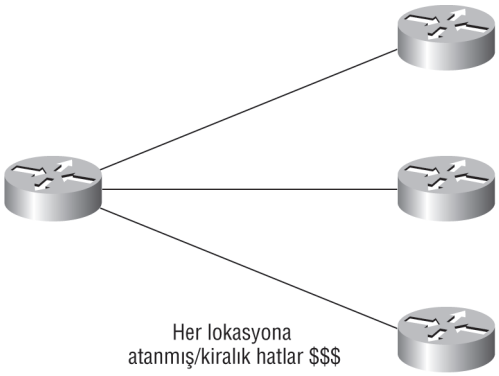
## Frame Relay Teknolojisine Giriş

Bir CCNA olarak, Frame Relay teknolojisinin temellerini anlamanız ve onu basit senaryolarda yapılandırabilmeniz gerekmektedir. İlk olarak, Frame Relay'in bir packet-switched teknoloji olduğunu anlayın. Şimdiye kadar öğrendiklerinize ilave olarak onunla ilgili bazı noktaları bildiğinizden emin olmalısınız:

- Onu yapılandırmak için `encapsulation hdlc` veya `encapsulation ppp` komutlarını kullanmayacaksınız.
- Frame Relay, noktadan-noktaya leased-line gibi çalışmaz (onun gibi görünüp onun gibi davrandığı halde).
- Frame Relay, genellikle, leased-line'lardan daha ucuzdur. Fakat bu kazançları sağlamak için bazı fedakarlıklar gerekir.

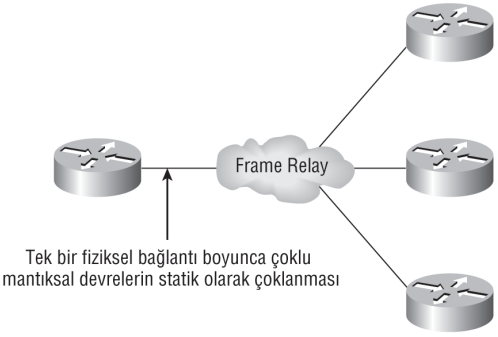
Öyleyse, hala neden Frame Relay kullanmayı düşünürsünüz? Frame Relay'den önce bir ağın neye benzediği ile ilgili bir fikre sahip olmak için, Şekil 14.14'e bir bakın. Şimdi de Şekil 14.15'i inceleyin. Corporate router ile Frame Relay switch arasında sadece bir bağlantı olduğunu görebilirsiniz. Bu oldukça kazançlıdır!

Örneğin, şirket ofisine, yedi uzak şube eklemeniz gerekseydi ve router'ınızda sadece bir seri interface olsaydı, Frame Relay, hayatınızı kurtaracaktı. Şimdi, tek arıza noktasına sahipsiniz ve bunun iyi bir şey olmadığını belirtmek zorundayım. Fakat Frame Relay, para tasarrufu yapmak için kullanılmaktadır, ağı daha güvenli yapmak için değil.




***Şekil 14.14:*** *Frame Relay'den önce.*

Frame Relay, düşük-maliyeli bir mesh network oluşturur.




***Şekil 14.15:*** *Frame Relay'den sonra.*

Şimdi, CCNA hedeflerine hazırlandığınızda bilmeniz gereken Frame Relay terimlerini işleyeceğim.

## Committed Information Rate (CIR)

Frame Relay, aynı anda birçok farklı müşteriye bir packet-switched ağı sağlamaktadır. Switch maliyetlerini, birçok müşteri arasında dağıttığından, bu gerçekten güzel bir şeydir. Fakat Frame Relay'in, tüm müşterilerin, sürekli ve aynı anda veri aktarmasına gerek duymadığı varsayımına dayanır.

Frame Relay, her kullanıcıya atanmış bir bant genişliği sağlayarak çalışır ve kullanıcıya, telekom ağındaki kaynaklar uygun olduğunda, onların garanti bant genişliklerini arttırmalarını da sağlar. Aslında, Frame Relay sağlayıcılar, müşterilerinin gerçekte ihtiyaçları olduğundan daha az miktarda bant genişliği satın almalarına izin verir. Frame Relay ile iki ayrı bant genişliği düzenlemesi vardır:

**Access rate:** Frame Relay interface'in aktarabileceği maksimum hız.

**CIR:** Taşınması garanti verinin maksimum bant genişliği. Aslında, servis sağlayıcının aktarmanıza izin vereceği, ortalama miktardır.

Şayet bu iki değer aynı olursa, Frame Relay bağlantısı bir leased-line'a benzer. Fakat onlar farklı değerlere ayarlanabilir. İşte bir örnek: 256Kbps bir CIR ve T1(1.544Mbps) erişim hızı satın aldığınızı düşünelim. Bunu yaparak, ihtiyacınız olan trafiğin ilk 256Kbps kısmının taşınması garanti olacaktır. Bunun dışındaki her şey, burst olarak belirtilir. Bu sizin garanti 256Kbps hızı aşan bir aktarımdır ve (şayet bu sizin anlaşmanızdaki değerse) T1 erişim hızına kadar herhangi bir değer olabilir. Şayet CIR ve MBR (maximum burst rate) olarak bilinen aşırı burst boyutları, erişim hızını aşarsa, ilave trafiğinizi unutmanız gerekir. Bu aslında belirli bir servis sağlayıcının abone ücretine bağlı olmakla beraber, muhtemelen iptal edilecektir.

Mükemmel bir dünyada, bu daima güzel bir şekilde çalışır. Fakat garanti kelimesini hatırladınız mı? Mecbur olması gereken, 256Kbps garanti hızındaki gibi? Yani, gönderdiğiniz verinin sizin garanti 256Kbps hızınızı aşan kısmı, bazen en iyi çaba gösterilerek ulaştırılacaktır. Telekom firmanızın sizin aktardığınız anda verinizi ulaştırma kapasitesi yoksa, frame'leriniz atılacaktır ve DTE bilgilendirilecektir. Zamanlama her şeydir. Sadece telekom firmanızın ekipmanlarının o an uygun olması durumunda, garanti 256 Kbps hızınızın altı katı bir hızla (T1) verinizi gönderebilirsiniz.

NOT *CIR, Frame Relay switch'in, veri transferi için mutabık olduğu, saniyedeki bit miktarını tanımlayan değerdir.*

## Frame Relay Enkapsülasyon Tipleri

Cisco router'larda Frame Relay'i yapılandırdığımızda, seri interface'lerde bir enkapsülasyon olarak Frame Relay'i belirtmeniz gerekir. Daha önce söylediğim gibi, Frame Relay ile PPP veya HDLC kullanamazsınız. Frame Relay yapılandırdığınızda, bir Frame Relay enkapsülasyon belirtirsiniz.(aşağıdaki çıktıdaki gibi). HDLC ve PPP'nin tersine, Frame Relay ile iki enkapsülasyon tipi vardır: Cisco ve IETF (Internet Engineering Task Force). Aşağıdaki router çıktısı, Cisco router'ınızda Frame Relay seçtiğinizde, bu iki farklı enkapsülasyon yöntemini göstermektedir:

```
RouterA(config)#int s0
RouterA(config-if)#encapsulation frame-relay ?
  ietf  Use RFC1490 encapsulation
  <cr>
```

Manuel olarak IETF yazmadıkça, Cisco varsayılan enkapsülasyondur.. İki Cisco cihaz bağlandığında, Cisco kullanılacak olan yöntemdir. Şayet Cisco bir cihazı, Cisco olmayan bir cihazla, Frame Relay ile bağlamanız gerekirse, IETF tip enkapsülasyonu seçmelisiniz. Hangisini seçerseniz seçin, iki uçta da Frame Relay enkapsülasyonun aynı olduğundan emin olun.

## Sanal Devreler (Virtual Circuit)

Frame Relay, leased line'ların kullandığı gerçek devrelerin tersine, sanal devreler kullanarak çalışır. Bu sanal devreler, servis sağlayıcının bulutuna bağlı binlerce cihazla beraber oluşan linklerdir. Frame Relay, iki DTE cihazınız arasında sanal bir devre sağlar. Onları bir devre üzerinden bağlı gibi gösterir. Frame'ler, geniş, paylaşılan bir alt yapıya gönderilir. Siz sadece bir sanal devreye sahip olduğunuzdan, bulutta olan karmaşıklığı asla görmezsiniz.

Tüm bunların üzerinde, iki sanal devre çeşidi vardır: Permanent ve switched. Permanent virtual circuit'ler (PVC), günümüzde kat kat daha fazla kullanılan, en yaygın çeşittir. Buradaki permanent'ın anlamı şudur; telekom firması eşleşmeler oluşturur ve faturanızı ödediğiniz sürece, onlar yerinde kalacaktır.

Switched virtual circuit'ler (SVC), bir telefon aramasına benzer. Verinin aktarılması gerektiğinde sanal devreler kurulur. Daha sonra veri transferi tamamlandığında, kaldırılır.

*Kuzey Amerika'daki bir telekom firması tarafından önerilen, SVC kullanan bir Frame Relay servisi hiç görmedim. Genelde özel Frame Relay ağlarında kullanılmaktadır.*

NOT

## Data Link Connection Identifier (DLCI)

Frame Relay PVC'leri, DTE uç cihazlarını, Data Link Connection Identifier (DLCI)'larla tespit eder. Bir Frame Relay servis sağlayıcısı tipik olarak, farklı sanal devreleri ayırt etmek için Frame Relay interface'lerinde kullanılan, DLCI değerleri atar. Birçok sanal devre, bir multipoint Frame Relay interface'inde sonlandırılabildiğinden, çok sayıda DLCI ona bağlıdır.

Açıklamama izin verin. Üç şube ofis ile bir merkez ofise sahip olduğunuzu farz edelim. Her şubeyi merkez ofise T1 kullanarak bağlasaydınız, merkez ofisteki router'ınızda her bir T1 için bir tane olmak üzere, üç seri interfece'e ihtiyacınız olacaktı. Basit, değil mi? Onun yerine, Frame Relay PVC'leri kullandığınızı düşünün. Her şubede, servis sağlayıcıya bağlı bir T1 ve merkez ofise sadece bir T1'e sahip olabilirsiniz. Merkez ofisteki tek T1'de, her biri bir şubeye giden üç PVC olacaktır. Sadece bir interface ve tek CSU/DSU olmasına rağmen üç PVC, üç ayrı devre gibi çalışır. Para kazancı konusunda ne söylediğimi hatırlıyor musunuz? İki ilave T1 interface'i ve CSU/DSU çifti ne kadardır? Cevap: Çok fazla. Öyleyse, neden gidip bonusunuzdaki artış yüzdenizi sormuyorsunuz?

Devam etmeden önce, Inverse ARP'ı (IARP) açıklamak ve bir Frame Relay ağında DLCI'larla nasıl kullanıldığını tartışmak istiyorum. Biraz ARP'a benzer ama o, bir DLCI'ı, IP adresine eşleştirir (ARP'ın, MAC adresini, IP adresine eşleştirmesi gibi). IARP'ı yapılandırmayacaksanız, onu pasif hale getirebilirsiniz. O, bir Frame Relay router'da çalışır ve Frame Relay switch'e nasıl ulaşacağını bilmesi için DLCI'ı bir IP adresine eşleştirir. `show frame-relay map` komutu ile IP'den DLCI eşleşmesini görebilirsiniz.

Şayet ağınızda, Cisco olmayan bir router varsa ve router, IARP'ı desteklemiyorsa, `frame-relay map` komutu ile statik olarak IP'den DLCI eşleşmesi sağlamanız gerekir.

*Inverse ARP (IARP), bilinen bir DLCI'ı bir IP adresine eşleştirmek için kullanılır.*

NOT

DLCI'lardan biraz daha bahsedelim. Onlar yerel olarak önemlidirler. Global anlamlılık, tüm ağın global anlam öneren LMI uzantıları kullanmasını gerektirir. Bundan dolayı global DLCI'ları sadece özel ağlarda bulursunuz.

Fakat DLCI'ların ağ boyunca bir frame'e sahip olmada işlevsel olmak için global olarak önemli olması gerekmez. Açıklamama izin verin: RouterA, RouterB'ye bir frame göndermek istediğinde, IARP'a veya ulaşmaya çalıştığı IP adresine manuel bir DLCI eşleşmesi olup olmadığına bakar. Frame'i, Frame Relay başlığındaki DLCI alanında bulunan DLCI değeri ile gönderir. Servis sağlayıcının switch'i bu frame'i alır ve gözlemlediği DLCI/fiziksel-port kombinasyonuna bakar. Bu kombinasyonla ilgili olarak, başlıkta kullanmak için, yeni lokal olarak önemli (yani, kendisiyle next-hop switch arasında) bir DLCI bulur ve tablosunda, aynı kayıtta, bir fiziksel çıkış portu yer almaktadır. Bu aynı şekilde RouterB'de de olur. Her cihaz çifti arasındaki DLCI tamamıyla farklı olabileceği

halde, RouterA'dan RouterB'ye tüm sanal devrenin tespit edebileceğini söyleyebilirsiniz. Buradaki önemli nokta, RouterA'nın bu farklılıklardan habersiz olmasıdır. DLCI'ı lokal olarak anlamlı yapan budur. DLCI'ların, PVC'nizin diğer ucunu bulmak için gerçekte telekom firması tarafından kullanıldığını aklınızda tutun.

DLCI'ların neden lokal olarak önemli olduğunu bulmak için Şekil 14.16'ya bakın. Şekilde DLCI 100'ün, RouterA'ya lokal olarak anlamlı olduğu düşünülür ve RouterA ile onun eriştiği Frame Relay switch arasındaki devreyi tanımlar. DLCI 200, RouterB ile onun eriştiği Frame Relay switch arasındaki devreyi belirtecektir.


***Şekil 14.16:*** *DLCI'lar, router'ınıza mevzilenmişlerdir.*

NOT *DLCI'lar, bir lokal router ile bir Frame Relay switch arasındaki mantıksal devreyi tanımlarlar.*

Bir PVC'yi tanımlamak için kullanılan DLCI numaraları, tipik olarak servis sağlayıcı tarafından atanır ve 16'dan başlar.

Bir DLCI numarasını, bir interface'e atamak için yapılandırabilirsiniz:

```
RouterA(config-if)#frame-relay interface-dlci ?
  <16-1007> Define a DLCI as part of the current
            subinterface
RouterA(config-if)#frame-relay interface-dlci 16
```

## Local Management Interface (LMI)

Local Management Interface (LMI), router'ınız ve onun bağlı olduğu ilk Frame Relay switch arasında kullanılan, sinyalleşme standardıdır. Çalışmayla ve servis sağlayıcının ve DTE (sizin router'ınız) arasındaki sanal devrenin durumu ile ilgili bilginin geçmesine izin verir. Aşağıdaki bilgileri iletir:

**Keepalive:** Bunlar, verinin aktığını doğrular.

**Multicasting:** Bu, örneğin routing bilgisinin verimli dağıtılması ve bir Frame Relay ağı boyunca ARP isteklerine izin veren LMI düzenlemesinin isteğe bağlı bir uzantısıdır. Multicasting, 1019'dan 1022'ye kadar rezerve edilmiş DLCI'leri kullanır.

**Global adresleme:** Bu, Frame Relay bulutunun tamamıyla bir LAN gibi çalışmasına izin vererek DLCI'lara global anlam kazandırır.

**Sanal devrelerin durumları:** Bu, DLCI durumunu sağlar. Durum sorguları ve mesajları, gönderilecek düzenli LMI trafiği olmadığında, keepalive'lar gibi kullanılır.

LMI'ın sizin router'larınız arasındaki bir iletişim olmadığını hatırlayın. O, router'ınız ve en yakın Frame Relay arasındaki bir iletişimdir. Böylece, diğer ucundaki almazken, PVC'nin diğer ucundaki router'ın, LMI'ı aktif olarak alması mümkündür. Ve tabi ki, PVC'ler, uçlardan birisi arızalı olduğunda çalışmazlar. (ben buna, LMI iletişimin lokal doğasını aydınlatması diyorum)

Üç farklı LMI mesaj formatı vardır: Cisco, ANSI ve Q.933A. Kullanımdaki farklı çeşitler, telekom firmasının anahtarlama donanımının tipine ve yapılandırmasına bağlıdır. Yani, router'ınızı, telekom firması tarafından sağlanması gereken doğru formatla yapılandırmanız zorunluluktur.

NOT *IOS 11.2 versiyonu ile başlayarak, LMI tipi, otomatik algılanmaktadır. Bu interface'in, switch tarafından desteklenen LMI tipini belirlemesini mümkün kılar. Şayet autosense özelliğini kullanmazsanız, hangi tipi kullanacağınızı anlamak için Frame Relay servis sağlayıcınızla irtibata geçmeniz gerekecektir.*

Cisco ekipmanında varsayılan ayar, Cisco'dur. Fakat servis sağlayıcınızın belirttiğine bağlı olarak ANSI veya Q.933A olarak değiştirmek zorunda kalabilirsiniz. Üç farklı LMI tipi, aşağıdaki router çıktısında görünmektedir:

```
RouterA(config-if)#frame-relay lmi-type ?
  cisco
  ansi
  q933a
```

Çıktıda göründüğü gibi, üç standart LMI sinyalleşme formatı da desteklenmektedir. Her birinin açıklaması şöyledir:

**Cisco:** LMI, varsayılan olarak Gang of Four tarafından tanımlanmıştır. Local Management Interface (LMI), Cisco Systems, StrataCom, Northern Telecom ve Digital Equipment Corporation tarafından geliştirildi ve Gang-of-Four LMI veya Cisco LMI olarak ünlü oldu.

**ANSI:** ANSI T1.617 standardı içeren AnnexD.

**ITU-T (Q.933A):** AnnexA, ITU-T standardına dahildir ve Q.933a komut anahtar sözcüğü kullanılarak tanımlanır.

Router'lar, bir frame-encapsulated interface'de, servis sağlayıcının Frame Relay switch'inden LMI bilgisi alır ve sanal devrelerin durumunu, aşağıdaki üç farklı durumdan birine günceller:

**Active state:** Her şey çalışır durumdadır ve router'lar, bilgiyi değiş tokuş edebilir.

**Inactive state:** Router'ın interface'i aktiftir ve switching ofisine bir bağlantıyla çalışıyordur. Fakat uzak router, çalışmıyordur.

**Deleted state:** Interface'de, switch'ten gelen herhangi bir LMI bilgisi yoktur. Bu, eşleşme hatası veya hat arızasından olabilir.

## Frame Relay Tıkanıklık Kontrolü

CIR hakkında konuştuklarımızı hatırlıyor musunuz? CIR'ı ne kadar düşük ayarlarsanız, veriniz için o kadar büyük bir risk olacağı çok net olmalı. Bu, CIR miktarının üzerindeki verinin (burst oluştuğunda) ne zaman aktarılacağı ve ne zaman aktarılmayacağı anahtar bilgisine sahip olarak, kolayca önlenebilir. Telekom'un paylaştığımız altyapısının boş ve müsait olduğunu ve onun, dolu ve tıkalı olduğunu anlamanın herhangi bir yolu var mıdır? Ayrıca, eğer bir yolu varsa, onu nasıl yaparız? Bu tamda benim şimdi, Frame Relay switch'in, DTE tıkanıklık problemlerini nasıl bildirdiği konusunu açıklayacağım konudur ve bu çok önemli soruların cevabıdır.

Aşağıda üç tıkanıklık terimi ve anlamlarını bulabilirsiniz:

**Discard Eligibility (DE):** Bildiğiniz gibi paketleri bir PVC'nin CIR miktarı üzerinde aktardığınızda (burst ettiğinizde), sağlayıcının ağının o an tıkalı olması durumunda, CIR'ı aşan her paketin atılması mümkündür. Bundan dolayı, aşırı bit miktarı, Frame Relay başlığında, bir Discard Eligibility (DE) ile işaretlenir. Ve servis sağlayıcının ağı tıkalı olursa, Frame Relay switch, DE bit setli paketleri düşürecektir. Şayet bant genişliğiniz, sıfır CIR ile yapılandırılırsa, DE daima aktif olacaktır.

**Forward Explicit Congestion Notification (FECN):** Frame Relay ağ, bulutta tıkanıklık belirlerse, switch, bir Frame Relay paket başlığındaki Forward Explicit Congestion Notification (FECN) bit'ini 1'e ayarlayacaktır. Bu, hedef DTE'ye, frame'in dolaştığı yolun tıkalı olduğunu belirtecektir.

**Backward Explicit Congestion Notification (BECN):** Switch, Frame Relay ağında tıkanıklık algılarsa o, kaynak router için hedeflenen bir Frame Relay frame'inde, Backward Explicit Congestion Notification (BECN*)* bit'i ayarlayacaktır. Bu router'ı, ilerde tıkanıklık olduğu konusunda uyarır.

NOT *Bu konuda daha fazla bilgi almak için, Cisco'nun web sitesinde, "Frame Relay Traffic Shaping" yazarak arama yapın.*

Fakat Cisco router'lar, siz onlara söylemedikçe bu tıkanıklık için bir eylemde bulunmayacaklardır.

### Frame Relay Tıkanıklı Kontrolü Kullanarak Hata Tespiti Yapmak

Şimdi, kullanıcılarınızın tamamının, şirket merkezine Frame Relay bağlantılarının çok yavaş olması konusunda sızlandıklarını düşünelim. Linkin, aşırı yüklü olduğundan kuşkulandığınızdan, `show frame-relay pvc` komutu ile Frame Relay tıkanıklık kontrol bilgisini doğrularsınız ve şu çıktıyı alırsınız:

```
RouterA#sh frame-relay pvc
PVC Statistics for interface Serial0/0 (Frame Relay DTE)

              Active     Inactive      Deleted       Static
  Local          1            0            0            0
  Switched       0            0            0            0
  Unused         0            0            0            0

DLCI = 100, DLCI USAGE=LOCAL, PVC STATUS=ACTIVE, INTERFACE=Serial0/0
 input pkts 1300          output pkts 1270       in bytes 21212000
 out bytes 21802000       dropped pkts 4         in pkts dropped 147
 out pkts dropped 0       out bytes dropped 0      in FECN pkts 147
 in BECN pkts 192        out FECN pkts 147
 out BECN pkts 259        in DE pkts 0             out DE pkts 214
 out bcast pkts 0         out bcast bytes 0
 pvc create time 00:00:06, last time pvc status changed 00:00:06
Pod1R1#
```

Görmek istediğiniz, `in BECN pkts 192` çıktısıdır. Çünkü bu, lokal router'a, şirket merkezine gönderilen trafiğin, tıkanıklık yaşadığını söyler. BECN'in anlamı, size dönüşe başlayan frame'in yolunun tıkanık olmasıdır.

## Frame Relay Kurulumu ve İzleme

Söylediğim gibi, tonlarca Frame Relay komutu ve yapılandırma seçeneği vardır. Fakat ben, CCNA sınav konularına çalıştığınızda, gerçekten bilmeniz gereken bir tanesine odaklanacağım. İki router ve onları bağlayan bir PVC'den oluşan, en basit yapılandırma seçeneklerinden biriyle başlayacağım. Sonra, subinterface'ler kullanarak daha zor bir yapılandırma ve yapılandırmayı doğrulamak için kullanılan bazı komutları göstereceğim.

### Tek Interface

Basit bir örneğe bakarak başlayalım. İki router'ı, tek bir PVC ile bağlamak istediğimizi farz edelim. Konfigürasyon şu şekilde görünecektir:

```
RouterA#config t
Enter configuration commands, one per line.  End with CNTL/Z.
RouterA(config)#int s0/0
RouterA(config-if)#encapsulation frame-relay
RouterA(config-if)#ip address 172.16.20.1 255.255.255.0
RouterA(config-if)#frame-relay lmi-type ansi
RouterA(config-if)#frame-relay interface-dlci 101
RouterA(config-if)#^Z
RouterA#
```

İlk adım, enkapsülasyonu Frame Relay olarak belirtmektir. Cisco veya IETF olarak, belirli bir enkapsülasyon tipi belirlemediğimden, varsayılan Cisco'nun kullanıldığına dikkat edin. Diğer router, Cisco olmayan bir router olsaydı, IETF belirtmem gerekecekti. Sonra, interface'e bir IP adresi tanımladım, daha sonra LMI tipi olarak, Telekom firması tarafından sağlanan bilgiye bağlı olarak, ANSI'yi belirttim.(Cisco, varsayılandır). Son olarak, kullanmak istediğimiz, (bana, ISP'im tarafından verilen) PVC'yi belirten ve bu fiziki interface'de sadece bir PVC olduğunu gösteren, DLCI 101'i ekledim.

Her iki tarafta doğru bir şekilde yapılandırıldıysa, devre aktif olacaktır.

*Bu yapılandırma tipinin komple bir örneği için, bir router'dan, kendi Frame Relay switch'inizi yapılandırmayı açıklayan Pratik Lab 14.3'e bakın.*

NOT

## Subinterface'ler

Şimdiye kadar muhtemelen, tek bir seri interface'de çoklu sanal devrelere sahip olabileceğinizi ve hepsinin, ayrı bir interface gibi davranacağını biliyorsunuzdur. Bunu, subinterface'ler kullanarak yapabiliriz. Bir subinterface'i, IOS tarafından tanımlanan mantıksal bir interface olarak düşünün. Birkaç subinterface, tek bir donanımsal interface'i paylaşacaktır. Yapılandırma amaçlarından dolayı, subinterface'ler ayrı fiziksel interface'ler gibi çalışacaktır.

Routing güncellemelerini kabul etmeyerek split horizon sorunlarından kaçınacak şekilde, Frame Relay ağındaki bir router'ı yapılandırmak için, subinterface'e atanan subnet ve benzersiz bir DLCI ile her PVC'yi, ayrı bir subinterface olarak yapılandırın.

Subinterface'leri, `int s0.`***`subinterface number`*** gibi bir komut kullanarak tanımlayın. İlk olarak fiziksel seri interface'de enkapsülasyon ayarlamanız gerekir. Daha sonra, genellikle PVC başına bir subinterface olarak, subinterface'leri tanımlayabilirsiniz. İşte bir örnek:

```
RouterA(config)#int s0
RouterA(config-if)#encapsulation frame-relay
RouterA(config-if)#int s0.?
  <0-4294967295>  Serial interface number
RouterA(config-if)#int s0.16 ?
  multipoint      Treat as a multipoint link
  point-to-point  Treat as a point-to-point link
RouterA(config-if)#int s0.16 point-to-point
```

Bir fiziksel interface üzerinde, oldukça fazla sayıda subinterface tanımlayabilirsiniz. Fakat sadece 1000 civarında DLCI olduğunu unutmayın. Yukarıdaki örnekte, bunun, bu PVC'ye taşıyıcı tarafından atanan DLCI numarasını belirtmesinden dolayı, subinterface 16'yı kullandım. İki subinterface çeşidi vardır:

*Subinterface'ler oluşturduğunuzda, fiziksel interface altında, bir IP adresine sahip olmadığınızdan emin olun.*

NOT

**Point-to-point:** Tek bir sanal devre, bir router'ı, diğerine bağladığında kullanılır. Her point-to-point subinterface, kendi subnetinde bulunmalıdır.

*Bir point-to-point subinterface, DLCI başına tek bir IP subnetini eşleştirir ve NBMA split horizon sorunlarını çözer.*

NOT

**Multipoint:** Multipoint, frame relay switch'e bağlı router'ların tüm interface'leri için tek bir subnet kullanıldığı yıldız şeklinde dizilmiş sanal devrelerin merkezinde bulunan router'da kullanılır. Bunu genellikle, bu modda hub router'a ve fiziksel interface'de (daima point-to-point) veya point-to-point subinterface modda spoke router'lara uygulanmış bulursunuz.

Sonra, çoklu subinterface'ler çalışan bir gerçek router örneği göstereceğim. Aşağıdaki çıktıda, subinterface numaralarının, DLCI numaralarıyla eşleştiğine dikkat edin. Zorunlu değildir ama interface'leri yönetmenizde size yardımcı olur:

```
interface Serial0
 no ip address (notice there is no IP address on the physical
interface!)
 no ip directed-broadcast
 encapsulation frame-relay
!
interface Serial0.102 point-to-point
 ip address 10.1.12.1 255.255.255.0
 no ip directed-broadcast
frame-relay interface-dlci 102
!
interface Serial0.103 point-to-point
 ip address 10.1.13.1 255.255.255.0
 no ip directed-broadcast
frame-relay interface-dlci 103
!
interface Serial0.104 point-to-point
 ip address 10.1.14.1 255.255.255.0
 no ip directed-broadcast
frame-relay interface-dlci 104
!
interface Serial0.105 point-to-point
 ip address 10.1.15.1 255.255.255.0
 no ip directed-broadcast
frame-relay interface-dlci 105
!
```

Belirtilen bir LMI tipi olmadığına dikkat edin. Yani, ya router'lar varsayılan olarak Cisco çalışıyordur ya da (Cisco IOS 11.2 ve sonrası çalışıyorsa) autodetect çalışıyorlardır. Her interface'in, tek bir DLCI'a eşlendiğini ve ayrı bir subnet olarak belirtildiğine dikkatinizi çekmek istiyorum. Point-to-point subinterface'lerin, split horizon sorunlarını da çözdüğünü hatırlayın.

## Frame Relay'i İzlemek

Frame Relay enkapsülasyon kurulup çalışınca, interface ve PVC'lerinizin durumunu sık sık kontrol etmek için bazı komutlar kullanılmaktadır. Onları gözlemek için `show frame ?` komutunu kullanın:

```
RouterA>sho frame ?
end-to-end     Frame-relay end-to-end VC information
fragment       show frame relay fragmentation information
ip             show frame relay IP statistics
lapf           show frame relay lapf status/statistics
lmi            show frame relay lmi statistics
map            Frame-Relay map table
pvc            show frame relay pvc statistics
qos-autosense  show frame relay qos-autosense information
```

```
route          show frame relay route
svc            show frame relay SVC stuff
traffic        Frame-Relay protocol statistics
vofr           Show frame-relay VoFR statistics
```

`show frame-relay` komutuyla göreceğiniz en yaygın parametreler, lmi, pvc ve map'tir.

Şimdi, en sık kullanılan komutlara ve sağladığı bilgilere bir göz atalım.

*show frame-relay* **LMI Komutu**

`show frame-relay lmi` komutu size, lokal router ve Frame Relay switch arasında değiş tokuş edilen LMI trafik istatistiklerini verecektir. İşte bir örnek:

```
Router#sh frame lmi

LMI Statistics for interface Serial0 (Frame Relay DTE)
LMI TYPE = CISCO
  Invalid Unnumbered info 0     Invalid Prot Disc 0
  Invalid dummy Call Ref 0      Invalid Msg Type 0
  Invalid Status Message 0      Invalid Lock Shift 0
  Invalid Information ID 0      Invalid Report IE Len 0
  Invalid Report Request 0      Invalid Keep IE Len 0
  Num Status Enq. Sent 0        Num Status msgs Rcvd 0
  Num Update Status Rcvd 0      Num Status Timeouts 0
Router#
```

`show frame-relay lmi` komutundan router çıktısı, LMI hatalarını ve LMI tipini gösterir.

*show frame pvc* **Komutu**

`show frame pvc` komutu size, tüm yapılandırılmış PVC'leri ve DLCI numaralarının bir listesini sunacaktır. Her PVC bağlantısının durumunu ve istatistikleri sağlar. Ayrıca size PVC başına, router'dan alınan BECN ve FECN paketlerinin sayısını da verir.

İşte bir örnek:

```
RouterA#sho frame pvc
PVC Statistics for interface Serial0 (Frame Relay DTE)
DLCI = 16,DLCI USAGE = LOCAL,PVC STATUS =ACTIVE,
INTERFACE = Serial0.1
 input pkts 50977876    output pkts 41822892
  in bytes 3137403144
 out bytes 3408047602   dropped pkts 5
  in FECN pkts 0
 in BECN pkts 0      out FECN pkts 0     out BECN pkts 0
 in DE pkts 9393     out DE pkts 0
 pvc create time 7w3d, last time pvc status changed 7w3d
DLCI = 18,DLCI USAGE =LOCAL,PVC STATUS =ACTIVE,
INTERFACE = Serial0.3
 input pkts 30572401   output pkts 31139837
  in bytes 1797291100
```

```
out bytes 3227181474   dropped pkts 5
 in FECN pkts 0
in BECN pkts 0      out FECN pkts 0     out BECN pkts 0
in DE pkts 28      out DE pkts 0
pvc create time 7w3d, last time pvc status changed 7w3d
```

Şayet sadece PVC 16 hakkında bilgiyi görmek istiyorsanız `show frame-relay pvc 16` komutunu yazabilirsiniz.

*show interface* **Komutu**

LMI trafiğini kontrol etmek için, `show interface` komutunu kullanabilirsiniz. `show interface` komutu, hem enkapsülasyon hem de katman2 ve katman3 hakkındaki bilgileri gösterir. O ayrıca, line, protokol, DLCI ve LMI bilgisini de görüntüler. Şunu inceleyin:

```
RouterA#sho int s0
Serial0 is up, line protocol is up
 Hardware is HD64570
 MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec, rely
  255/255, load 2/255
 Encapsulation FRAME-RELAY, loopback not set, keepalive
  set (10 sec)
 LMI enq sent 451751,LMI stat recvd 451750,LMI upd recvd
  164,DTE LMI up
 LMI enq recvd 0, LMI stat sent 0, LMI upd sent 0
 LMI DLCI 1023 LMI type is CISCO frame relay DTE
 Broadcast queue 0/64, broadcasts sent/dropped 0/0,
  interface broadcasts 839294
```

Yukarıdaki LMI DLCI, kullanılan LMI tipini tanımlamak için kullanılmaktadır. Şayet o, 1023 olursa, varsayılan LMI tipi olan, Cisco'dur. LMI DLCI, sıfır ise, o zaman ANSI LMI tipidir (Q.933A'da sıfır kullanır). LMI DLCI, 0 ve 1023 dışında bir şey ise, servis sağlayıcınızı arayın, büyük bir probleminiz var demektir.

*show frame map* **Komutu**

`show frame map` komutu Network katmanı, DLCI eşleşmesini göstermektedir. Şöyle çalışmaktadır:

```
RouterB#show frame map
Serial0 (up): ipx 20.0007.7842.3575 dlci 16(0x10,0x400),
              dynamic, broadcast,, status defined, active
Serial0 (up): ip 172.16.20.1 dlci 16(0x10,0x400),
              dynamic, broadcast,, status defined, active
Serial1 (up): ipx 40.0007.7842.153a dlci 17(0x11,0x410),
              dynamic, broadcast,, status defined, active
Serial1 (up): ip 172.16.40.2 dlci 17(0x11,0x410),
              dynamic, broadcast,, status defined, active
```

Seri interface'in, birisi IP diğeri IPX için iki eşleşmeye sahip olduğuna dikkat edin. Ayrıca, network katman adreslerinin, dynamic protocol Inverse ARP (IARP) ile çözümlenmesi de önemlidir. DLCI numaralarının listelenmesinden sonra, parantez içinde bazı sayılar görebilirsiniz. İlki, serial0'da kullanılan, 16 DLCI numarasının, hexadecimal karşılığı olan, 0x10 ve serial1'de, DLCI 17 için kullanılan 0x11'dir. İkinci sayılar, Frame Relay frame'inde yapılandırılan DLCI numaraları olan, 0x400 ve 0x410'dur. Bit'lerin frame'de yayılmaları farklı olduğundan, onlar değişiktir.

### *debug frame lmi* Komutu

`debug frame lmi` komutu, varsayılan olarak, router konsollarındaki çıktıyı gösterecektir (her hangi bir debug komutu ile olduğu gibi). Bu komutun size vereceği bilgi, router ve switch'in doğru LMI bilgisini değiş tokuş ettiğini anlamanıza yardım ederek, Frame Relay bağlantısını doğrulama ve hata tespiti yapmanızı mümkün kılar. İşte bir örnek:

```
Router#debug frame-relay lmi
Serial3/1(in): Status, myseq 214
RT IE 1, length 1, type 0
KA IE 3, length 2, yourseq 214, myseq 214
PVC IE 0x7 , length 0x6 , dlci 130, status 0x2 , bw 0
Serial3/1(out): StEnq, myseq 215, yourseen 214, DTE up
datagramstart = 0x1959DF4, datagramsize = 13
FR encap = 0xFCF10309
00 75 01 01 01 03 02 D7 D6

Serial3/1(in): Status, myseq 215
RT IE 1, length 1, type 1
KA IE 3, length 2, yourseq 215, myseq 215
Serial3/1(out): StEnq, myseq 216, yourseen 215, DTE up
datagramstart = 0x1959DF4, datagramsize = 13
FR encap = 0xFCF10309
00 75 01 01 01 03 02 D8 D7
```

## Frame Relay Ağlarında Hata Tespiti

Neye baktığınızı bildiğiniz sürece, Frame Relay ağlarında hata tespiti, diğer ağ türlerinde hata tespitinden daha zor değildir. Frame Relay yapılandırmalarında yaygın olarak olan bazı temel problemleri ve onları nasıl çözeceğimize bakacağız.

Listenin başında, enkapsülasyon problemleri var. Yakın zamanda öğrendiğiniz gibi, iki Frame Relay enkapsülasyon vardır: Cisco ve IETF. Cisco varsayılan olandır ve Frame Relay ağının her iki ucunda Cisco router'a sahipsiniz demektir. Eğer Frame Relay ağınızın uzak ucunda, Cisco router'a sahip değilseniz, aşağıda gösterildiği gibi IETF çalıştırmak zorundasınız:

```
RouterA(config)#int s0
RouterA(config-if)#encapsulation frame-relay ?
  ietf  Use RFC1490 encapsulation
  <cr>
RouterA(config-if)#encapsulation frame-relay ietf
```

Doğru enkapsülasyon kullandığınızı doğrulayınca, Frame Relay eşleşmelerinizi kontrol etmeniz gerekmektedir. Örnek olarak, Şekil 14.17'ye bakın.


```
RouterA#show running-config
interface s0/0
ip address 172.16.100.2 255.255.0.0
encapsulation frame-relay
frame-relay map ip 172.16.100.1 200 broadcast
```

***Şekil 14.17:*** *Frame Relay eşleşmeleri.*

Peki, RouterA Frame Relay ağı boyunca, RouterB ile neden konuşamıyor? Bunu anlamak için `frame-relay map` ifadesine yakından bakın. Şimdi problemi anladınız mı? Frame Relay switch ile iletişim için uzak bir DLCI kullanamazsınız, kendi DLCI'ınızı kullanmalısınız. Eşleşme DLCI 200 yerine, DLCI 100'ü içermeliydi.

Doğru Frame Relay enkapsülasyon sahip olduğunuza nasıl emin olacağınızı ve DLCI'ların sadece lokal olarak önemli olduğunu bildiğinizden, Frame Relay ile ilgili tipik bazı routing protokol sorunlarına bakalım. Şekil 14.18'deki iki yapılandırmada herhangi bir problem olup olmadığına bakın.


```
RouterA#show running-config
interface s0/0
ip address 172.16.100.2 255.255.0.0
encapsulation frame-relay
frame-relay map ip 172.16.100.1 100
router rip
     network 172.16.0.0
```

```
RouterB#show running-config
interface s0/0
ip address 172.16.100.1 255.255.0.0
encapsulation frame-relay
frame-relay map ip 172.16.100.2 200
router rip
     network 172.16.0.0
```

***Şekil 14.18:*** *Frame Relay routing problemleri.*

Konfigürasyonlar oldukça iyi görünmektedir. Öyleyse problem nedir? Frame Relay'in, varsayılan olarak non-broadcast multi-access (NBMA) ağı olduğunu hatırlayın. Yani, PVC boyunca her hangi bir broadcast göndermeyecektir. Bu nedenle, eşleşme ifadeleri, hattın sonunda `broadcast` değişkenine sahip olmayacağından, RIP güncellemeleri gibi broadcsat'ler, PVC boyunca gönderilmeyecektir.

Şimdi, seri WAN bağlantılarımızı yapılandırmak için SDM'i kullanalım.

# WAN Bağlantıları İçin SDM Kullanmak

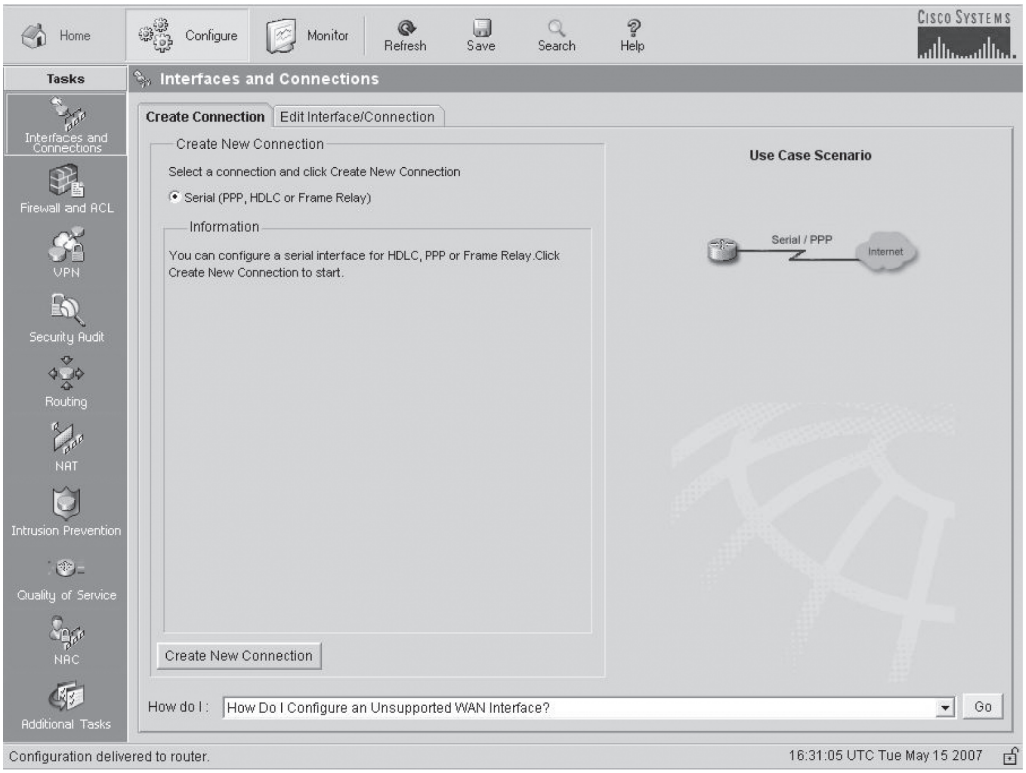
WAN bağlantılarının, SDM kullanılarak nasıl yapılandırılacağını size göstereceğim. Seçeneklerim sadece, HDLC (varsayılan), PPP ve Frame Relay. Router'larınız, onda kurulan interface'lere bağlı olarak farklı seçeneklere sahip olabilirsiniz.

HDLC zaten çalıştığı için ve HDLC ile çok fazla yapılandırma olmadığından, router'lar arasında PPP yapılandıracağım, kimlik doğrulama kullanacağım ve sonra SDM ile bir router'da Frame Relay'in nasıl yapılandırılacağını göstereceğim.

## SDM kullanarak, Kimlik Doğrulama ile PPP Yapılandırmak

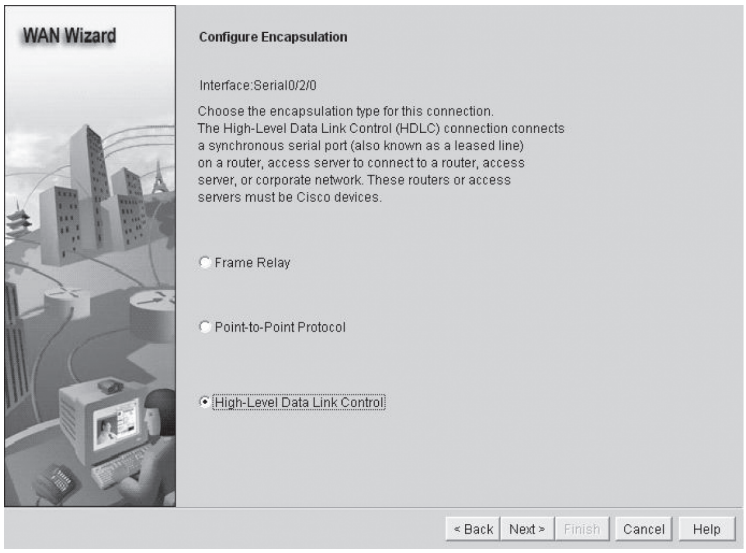
İlk olarak, kimlik doğrulaması ile PPP kullanarak, Corp router ile R3 router arasındaki seri WAN linkini yapılandıracağım. Yapmam gereken ilk şey, Interfaces and Connections Tasks'dan interface'i silmek, daha sonra Edit Interface Connection sekmesine ve sonrada Delete'e tıklamaktır. Şayet bunu yapmazsam, interface, SDM yardımıyla yapılandırmaya uygun olarak görünmeyecektir. Onu, CLI'dan kolayca yapabilirdim. Fakat onu yapacağımız yer burası değildir.

Interface yapılandırmasını silince, the Create Connection sekmesindeki Create New Connectio'a tıkladım.

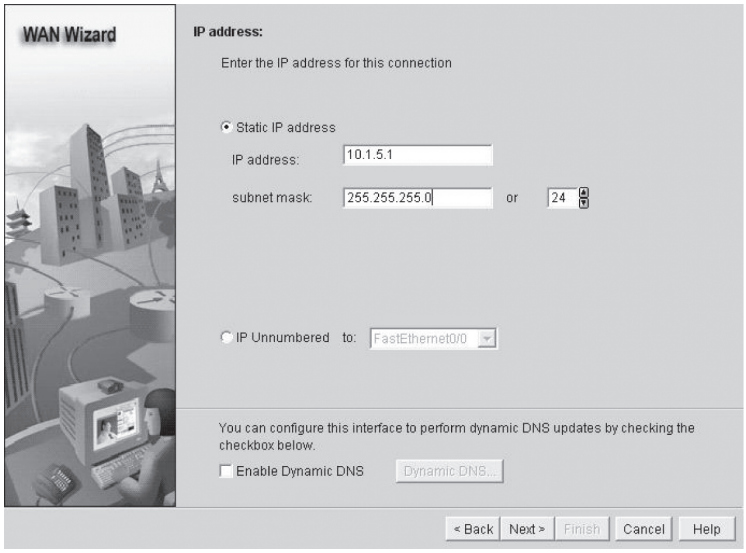



Create New Connection'a tıklayınca, serial WAN Configuration Wizard'ın ilk ekranı geldi.

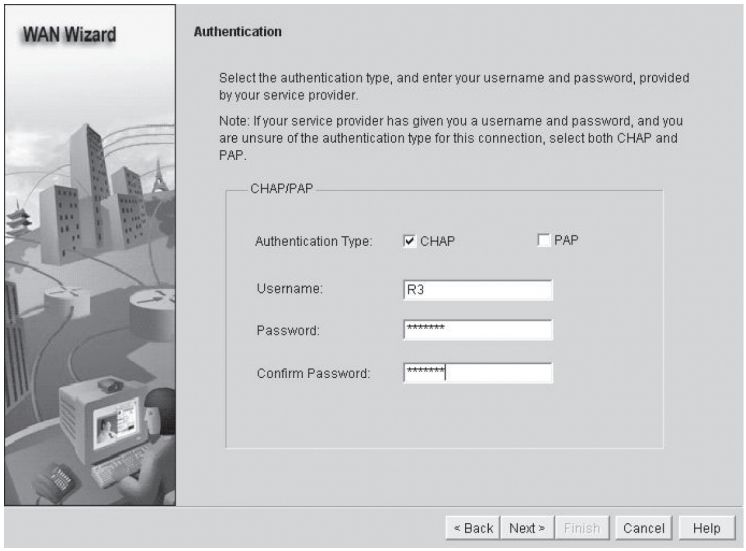

Sonra Next butonuna tıkladım ve HDLC'nin hazır olduğu ekranı geldi. Onun geçerli olması için tekrar Next butonuna tıklamam gerekti.



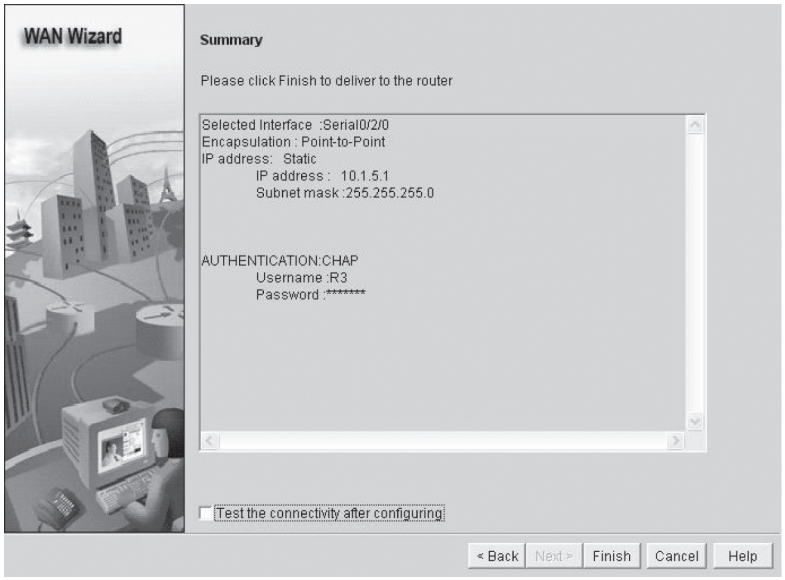

Bunun yerine Point-to-Point protokolüne ardından Next Butonuna tıkladım. Böylece IP adres ekranına ulaştım



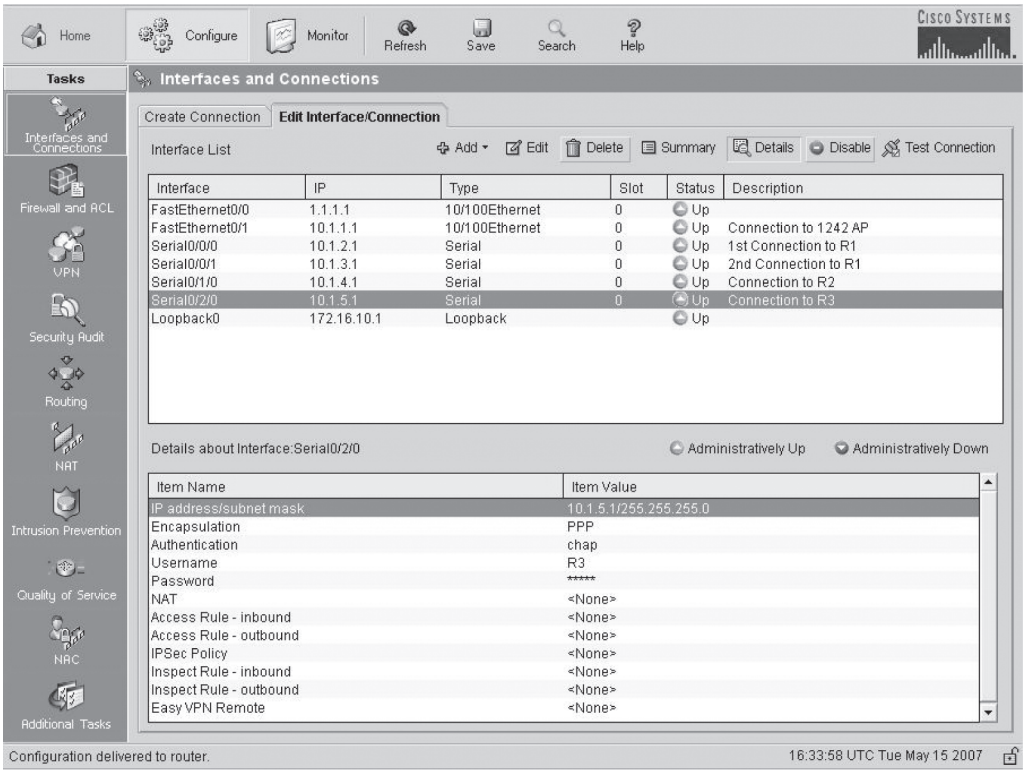

Vermek istediğim IP adresini ekledim, Next butonuna tıkladım ve Authentication ekranı geldi.



Buraya herhangi bir kimlik doğrulama bilgisi girmek zorunda değilsiniz. Ben yine de gireceğim. Username alanı, uzak router'ın (R3) ismi içindir (veya ISP'nizin size sağladığı şifre ile aynı). Sonra Next butonuna tıkladım. Konfigürasyonun özetini gösteren bir ekran göründü. Ve Finish butonuna tıkladım.




Interfaces and Connections'dan, Edit Interface/Connection sekmesinde, Serial0/2/0'ımın, Corp router'ında, PPP ve CHAP authentication ile ayarlandığını görebiliyoruz.




Şimdi R3 router'a gideceğim ve Corp router'da gösterdiğim aynı yapılandırmayı uygulayacağım. Kullanıcı adı olarak Corp ve aynı şifreyi kullanacağım (aynı PPP CLI yapılandırmasında daha önce gösterdiğim gibi).

Her iki router yapılandırıldıktan sonra, Corp router'dan CLI çıktısı aşağıdaki gibidir:

```
 !
interface Serial0/2/0
 description Connection to R3$FW_OUTSIDE$
 ip address 10.1.5.1 255.255.255.0
 ip verify unicast reverse-path
 ip virtual-reassembly
 encapsulation ppp
 clock rate 2000000
 ppp authentication chap callin
 ppp chap hostname R3
 ppp chap password 0 cisco
```

Bitirdiğimi düşündünüz, değil mi? Bütün bunlardan sonra, linkler yapılandırıldı ve çalışıyoruz, değil mi? Fakat henüz değil. Bir ISP'ye bağlı olsak, çalışabilirdi. Aslında T1 gibi bir point-to-point bağlantıya sahip değiliz. Bundan sonra ISP authentication komutlarını sağlamalıydı. Sorun, atanmış bir point-to-point serial bağlantımız olması ve authentication ile SDM PPP'nin, CLI'dan yardım almaksızın çalışmamasıdır. Şekle gidelim. (CLI ile daha kolay olduğunu söylemiştim!)

Router'ların çok kolay şekilde yapılandırılmasına rağmen, her şeyin neden çalışmadığını bilmemin sebebi şudur:

```
Corp#sh int s0/2/0
Serial0/2/0 is up, line protocol is down
  Hardware is GT96K Serial
  Description: Connection to R3$FW_OUTSIDE$
  Internet address is 10.1.5.1/24
  MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation PPP, LCP Listen
[output cut]
```

Listelenen ilk öğe, Physical katmanda, haberleşme sinyalinin algılandığını, fakat Data Link katmanında, "line protocol is down" göstermektedir. Yani, R3 router'dan keepalive'lar alamamaktayız. Fakat neden? Doğru yapılandırdığımıza ve her şeyin oldukça kolay olduğuna eminim. Gelin, çalışır durumdaki kimlik doğrulamaya bakalım ve ne anlayacağımızı görelim:

```
Corp#debug ppp auth
*May 15 18:46:12.039: Se0/2/0 PPP: Authorization required
*May 15 18:46:12.039: Se0/2/0 CHAP: O CHALLENGE id 33 len 23 from
"R3"
*May 15 18:46:12.039: Se0/2/0 CHAP: I CHALLENGE id 33 len 25 from
"Corp"
*May 15 18:46:12.043: Se0/2/0 CHAP: I RESPONSE id 33 len 25 from
"Corp"
*May 15 18:46:12.043: Se0/2/0 CHAP: Using hostname from interface
CHAP
*May 15 18:46:12.043: Se0/2/0 CHAP: Using password from interface
CHAP
```

```
*May 15 18:46:12.043: Se0/2/0 CHAP: O RESPONSE id 33 len 23 from
"R3"
*May 15 18:46:12.043: Se0/2/0 PPP: Sent CHAP LOGIN Request
*May 15 18:46:12.043: Se0/2/0 PPP: Received LOGIN Response FAIL
*May 15 18:46:12.043: Se0/2/0 CHAP: O FAILURE id 33 len 25 msg is
"Authentication failed"
Corp#un all
```

Aslında, kimlik doğrulama komutları çalışmayı deniyor gibi görünüyor, fakat sonunda bir şeyler başarısız oluyor. Burası, sizin kimlik doğrulamanızı yapılandıran bir servis sağlayıcıya bağlanamıyorsanız, CLI'ın devreye girdiği yerdir. Şimdi her router'a gideceğim ve `username` komutunu kullanacağım. Bu oldukça basittir, fakat bunun için SDM'in, en azından istemci sağlamaması beni şaşırtıyor. Komut aşağıdaki gibidir:

```
Corp(config)#username R3 password cisco
```

Ve şimdi R3 router için:

```
R3(config)#username Corp password cisco
```

Sonunda çalışıyor olmalısınız. Bunu kontrol edelim:

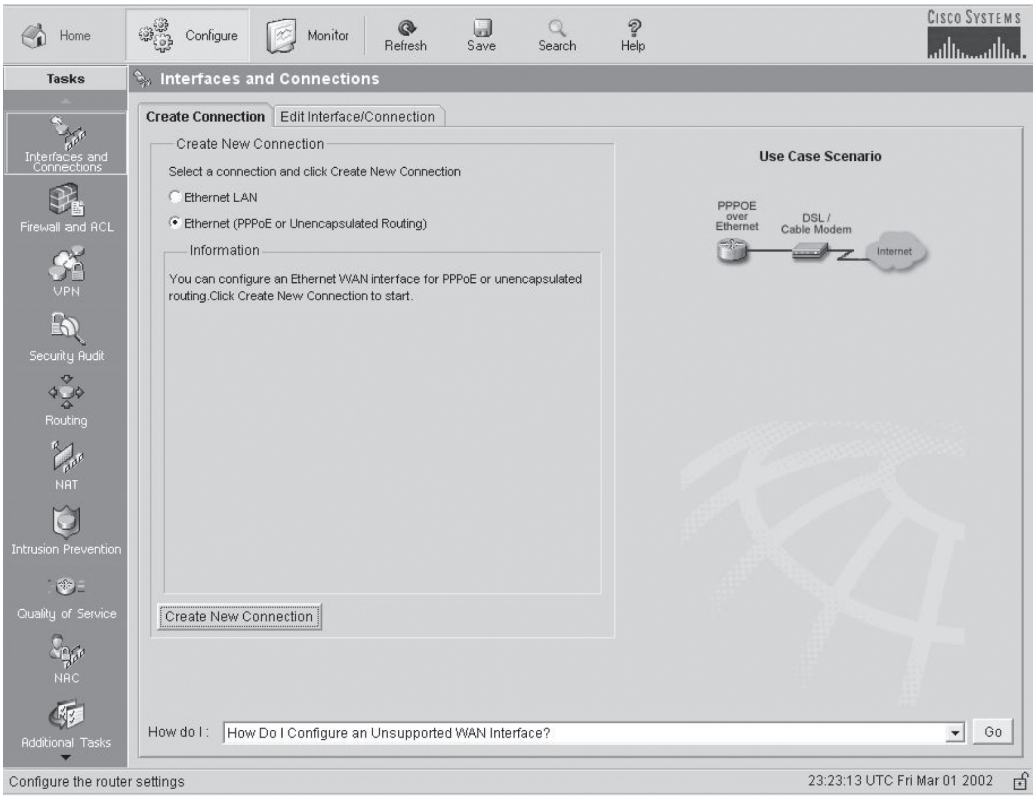
```
Corp#debug ppp auth
PPP authentication debugging is on
*May 15 16:53:34.479: Se0/2/0 PPP: Authorization required
*May 15 16:53:34.479: Se0/2/0 CHAP: O CHALLENGE id 1 len 25 from
"Corp"
*May 15 16:53:34.483: Se0/2/0 CHAP: I RESPONSE id 1 len 23 from
"R3"
*May 15 16:53:34.483: Se0/2/0 PPP: Sent CHAP LOGIN Request
*May 15 16:53:34.483: Se0/2/0 PPP: Received LOGIN Response PASS
*May 15 16:53:34.487: Se0/2/0 PPP: Sent LCP AUTHOR Request
*May 15 16:53:34.487: Se0/2/0 PPP: Sent IPCP AUTHOR Request
*May 15 16:53:34.487: Se0/2/0 LCP: Received AAA AUTHOR Response
PASS
*May 15 16:53:34.487: Se0/2/0 IPCP: Received AAA AUTHOR Response
PASS
*May 15 16:53:34.487: Se0/2/0 CHAP: O SUCCESS id 1 len 4
*May 15 16:53:34.487: Se0/2/0 PPP: Sent CDPCP AUTHOR Request
*May 15 16:53:34.491: Se0/2/0 PPP: Sent IPCP AUTHOR Request
*May 15 16:53:34.491: Se0/2/0 CDPCP: Received AAA AUTHOR Response
PASS
```

SDM'in, bir ISP'ye bağlandığınızı ve ISP'nin, kimlik doğrulaması için username ve password sağlayacağını varsaydığını anlayın. Komik, ama buna rağmen kesinlikle doğrudur!

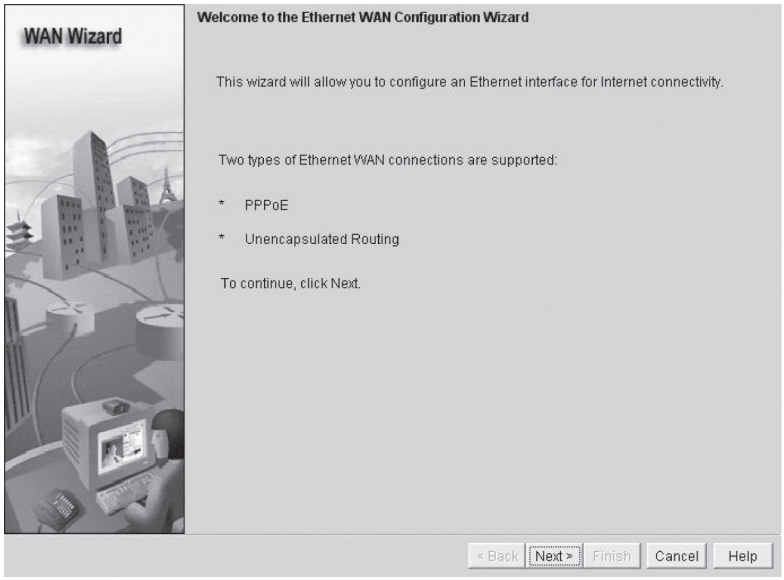
## SDM ile PPPoE Yapılandırmak

SDM ile PPPoE yapılandırmak için, ilk olarak bir DSL modeme bağlı router'a ve sonra daha önce yaptığımız gibi, interface'i bir istemci olarak yapılandırmaya ihtiyacımız vardır. Onu, 871W router'ımda SDM kullanarak, zahmetsiz yapacağım.

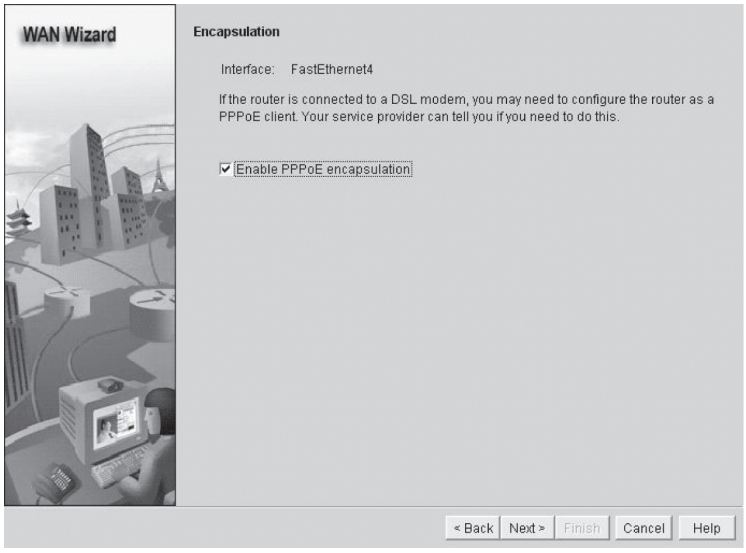
SDM ile router'a bağlandıktan sonra, SDM'den FastEthernet interface'ini sildim. Sonra, Tasks menüsündeki Interfaces and Connections altında Create Connections'ı seçtim.



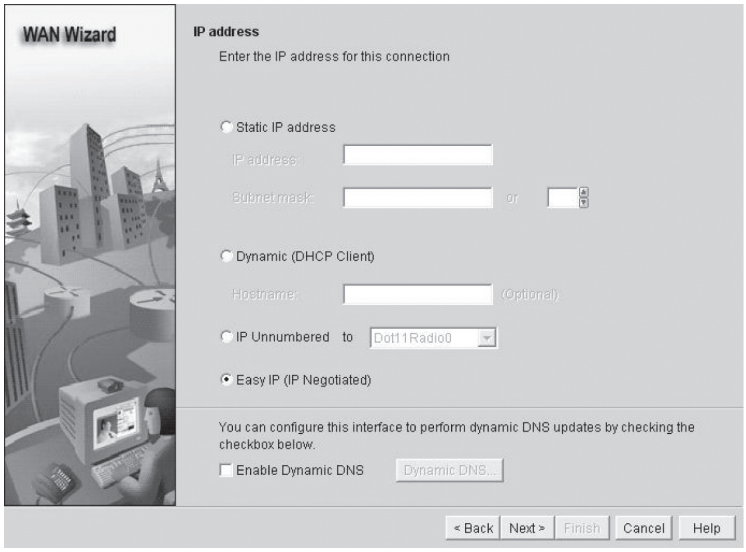

Buradan, Ethernet (PPPoE veya Unencapsulated Routing)'i seçtim, sonra Create New Connection'a tıkladım ve Ethernet WAN Configuration Wizard'ın giriş ekranına geldim.



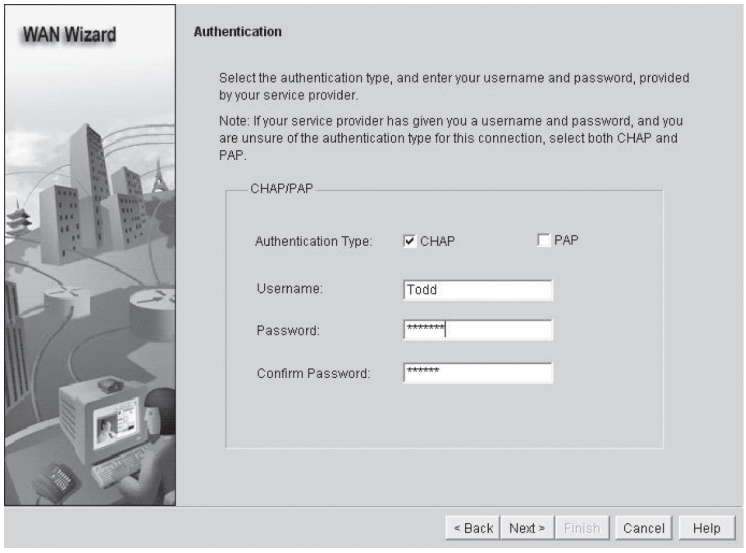
Buradan, Next butonuna tıkladım. Encapsulation ekranından, Enable PPPoE Encapsulation'ı kontrol ettim ve Next butonuna tıkladım.




Benden IP bilgim istendi. Easy IP (IP Negotiated, yani DHCP)'yi seçtim ve Next butonuna tıkladım.

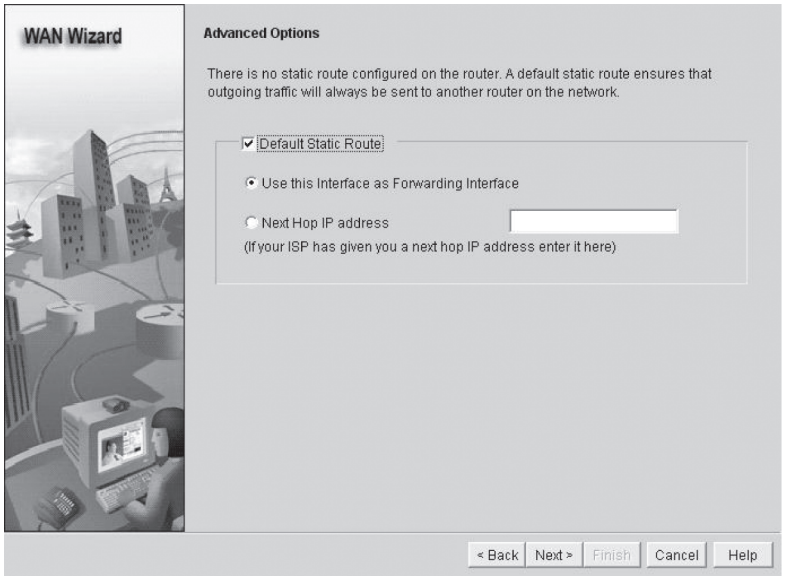



Şimdiki ekran, kimlik doğrulama bilgimi istedi.

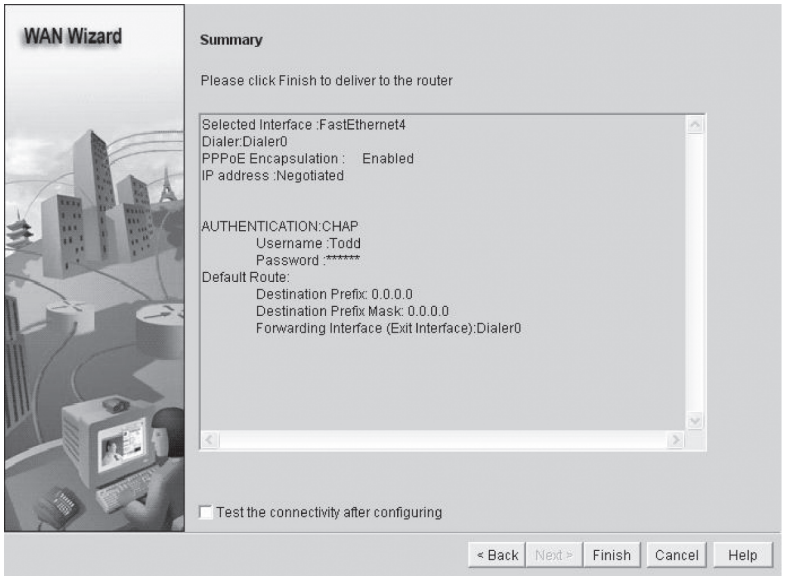


Normal olarak, bunun çalışır olması için bu bilginin, ISP'im tarafından sağlanmış olması gerekir. Burada, bazı basit bilgileri doldurdum ve Next butonuna tıkladım.

Sonraki ekran benden routing bilgisini istedi.




Bunu, Default Static Route'u kontrol ederek ve Use this Interface as Forwarding Interface'i seçerek, gateway of last resort'um yapmak için seçtim. Next-hop adresini ekleyebilirdim, DHCP kullandığımdan, onun değişmesi muhtemeldir. Next'i seçtikten sonra, konfigürasyonumun özetini aldım.




Konfigürasyonu router'ıma yüklemek için Next butonuna tıkladım.

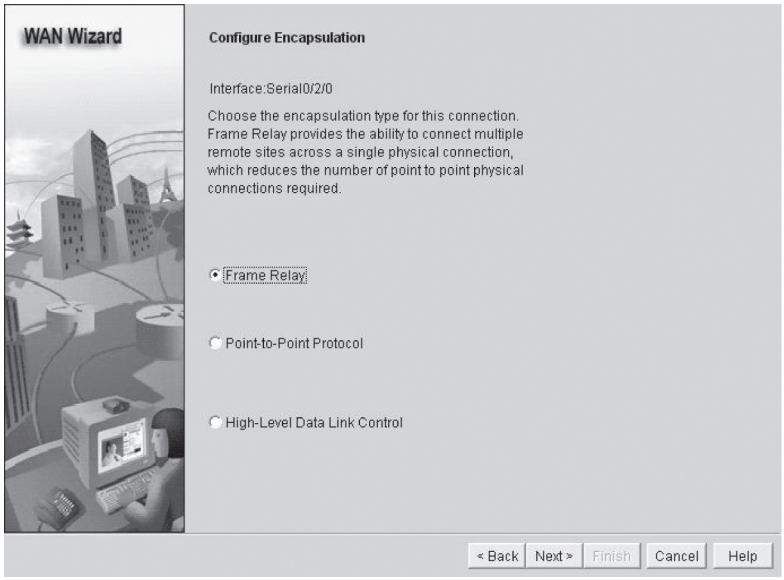
Şimdi, SDM ile Frame Relay yapılandırmasına geçelim.

**NOT** *SDM ile PPPoE yapılandırmasıyla ilgili daha fazla bilgi için, SDM demosunu kullanın. Bilgisayarınıza SDM'i kurabilir ve bölüm 4'de, Pratik Lab 4.6'da açıklandığı şekilde, tam demo programını çalıştırabilirsiniz.*

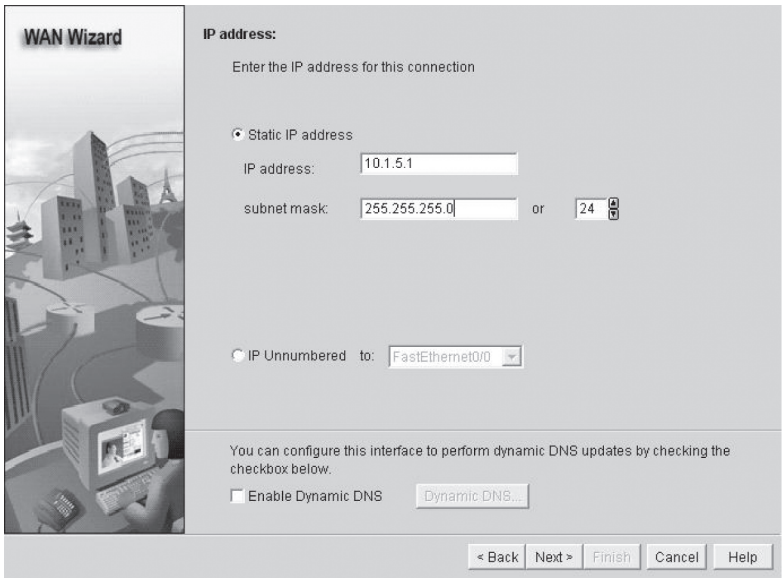
## SDM ile Frame Relay Yapılandırmak

Şimdi, Corp ve R3 router'ları arasındaki seri bağlantımız etkindir ve PPP çalışmaktadır. PPPoE'nun nasıl yapılandırılacağını biliyorsunuz, ben SDM kullanarak Frame Relay'i kurmanın ne kadar basit olduğunu göstermek istiyorum. SDM ile interface yapılandırmanızı sildikten sonra, yeni bir seri bağlantı oluşturacağım.

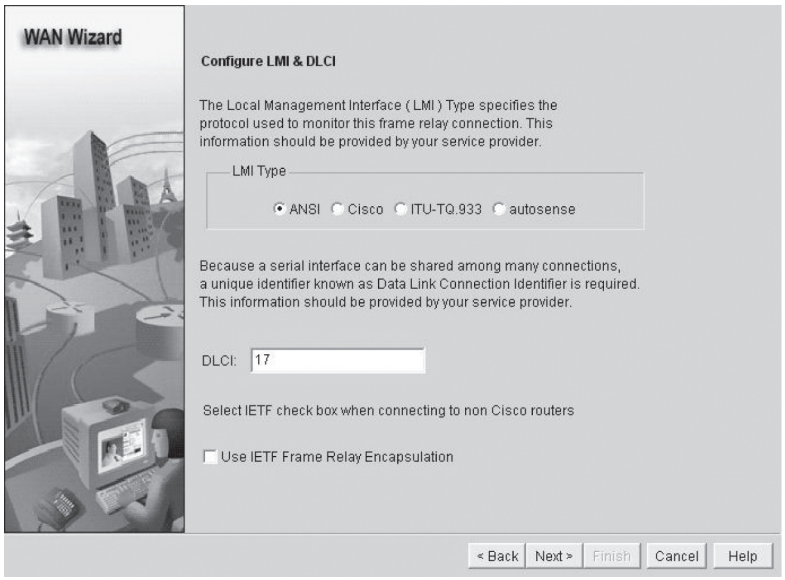
Interface Wizard'ı açacağım ve Corp router'ın Serial0/2/0 interface'i için Frame Relay'i seçeceğim.



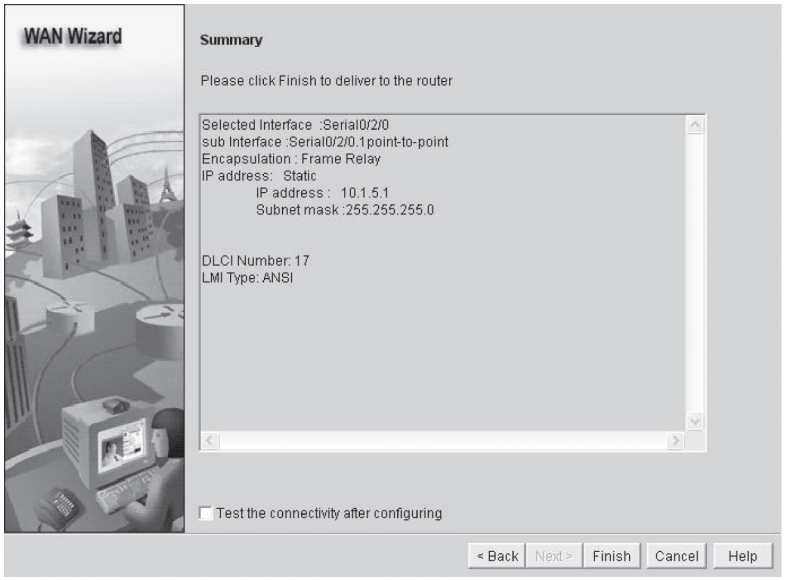

Sonra, PPP yapılandırdığımda aldığım ekran geldi. Statik IP adresi verdim ve Next'i seçtim.




Gerçek hayatta burası servis sağlayıcım tarafından bana verilen LMI ve DLCI bilgisini eklediğim yerdir.




Sol alt taraftaki check box'ın, IETF enkapsülasyon kullanmak isteyip istemediğimi sorduğuna dikkat edin. Bunun Frame Relay bulutunun diğer tarafında Cisco router olmadığı anlamına geldiğini hatırlayın. Next butonuna tıkladıktan sonra, Summary ekranına geldim.




Bu, konfigürasyonumun özetini gösterir. Onu router'ıma göndermek için Finish butonuna tıkladım. Tam olarak router'ıma ne indirildiğini anlamak için CLI'da bakalım. Aşağıdaki çıktıda, fiziksel interface'den IP adresinin, SDM'in oluşturduğu subinterface'e nasıl taşındığına dikkat edin.

```
!
interface Serial0/2/0
 description Connection to R3$FW_OUTSIDE$
 no ip address
 ip verify unicast reverse-path
 ip virtual-reassembly
 encapsulation frame-relay
 clock rate 2000000
 frame-relay lmi-type ansi
```

```
!
interface Serial0/2/0.1 point-to-point
 ip address 10.1.5.1 255.255.255.0
 frame-relay interface-dlci 17 CISCO
!
```

Şayet, bir ISP'ye bağlı router olsaydı ve ISP doğru şekilde yapılandırılsaydı, o zaman PVC'im aktif olmalıydı. LMI tipi olarak Cisco'yu seçtiğimi hatırlayın. Yani ISP'min bir Cisco Frame Relay switch'i olması gerekir. Bu nerdeyse olasılık dışıdır. Çoğu zaman ANSI LMI kullanacaksınız.

Şimdi, Corp router ve R3 router'ım arasında bir VPN (virtual private network) oluşturalım.

# Virtual Private Network

VPN terimini daha önce birden fazla kez duyduğunuza bahse girerim. Belki ne olduğunu bile biliyorsunuzdur. Virtual private network (VPN), TCP/IP olmayan protokollerin tünellenmesi ve gizliliğini mümkün kılarak, internet boyunca özel ağların oluşturulmasına izin verir. VPN'ler, daha pahalı ve kalıcı araçları kullanmak yerine, internet gibi genel bir ortam üzerinden bağlantısız ağların bağlanırlığını ve uzak kullanıcılar sağlamak için kullanılır.

VPN tipleri, bir görevde oynadıkları role bağlı olarak isimlendirilir. Üç farklı VPN kategorisi vardır:

**Remote access VPN'ler:** Remote access VPN, telecommuter gibi kullanıcıların ne zaman ve nerede olurlarsa olsun şirket ağına güvenli erişime izin verirler.

**Site-to-site VPN'ler:** Site-to-site VPN veya intranet VPN'leri, bir şirketin uzak yerleşimlerini, şirket omurgasına Frame Relay gibi daha pahalı WAN bağlantıları gerektirmek yerine, internet gibi genel bir ortam üzerinden güvenli bir şekilde bağlamaya izin verir.

**Extranet VPN'ler:** Extranet VPN'ler, bir kuruluşun tedarikçilerinin, partnerlerinin ve müşterilerinin, business-to-business (B2B) haberleşmeleri için sınırlı bir yolla, şirket ağına bağlanmalarına izin verir.

VPN'ler ucuz ve güvenli olduğundan, VPN'lerin nasıl oluşturulduğunu anlamayı gerçekten istiyorsunuz, değil mi? VPN oluşturmanın birden fazla yolu vardır. İlk yaklaşım, bir IP ağındaki uç noktalar arasında kimlik doğrulama ve şifreleme servisleri oluşturmak için IPSec kullanmaktır. İkinci yol, bir ağdaki uç noktalar arasında bir tünel oluşturmanıza izin veren, tünelleme protokolleri yardımıyla olur. Tünelin kendisinin, veri veya protokollerin, başka protokole enkapsüle edilmesi anlamına geldiğini anlayın.

İlk olarak IPSec'ten bahsedeceğim. Fakat önce kullanımdaki dört en yaygın tünelleme protokolünden bahsetmek istiyorum:

**Layer 2 Forwarding (L2F):** Layer 2 Forwarding (L2F), Cisco'ya özgü bir tünelleme protokolüdür ve virtual private dial-up network'ler (VPDN) için geliştirilen ilk tünelleme protokolüdür. VPDN, bir cihazın, bir şirket ağına, güvenli bir bağlantı kurması için, dial-up bağlantı kullanmasına izin verir. Daha sonra yerine L2F'le uyumlu olan L2TP geçmiştir.

**Point-to-Point Tunneling Protocol (PPTP):** Point-to-Point Tunneling Protocol (PPTP), verinin uzak ağlardan, şirket ağına güvenlik transferine izin vermek için, Microsoft tarafından geliştirilmiştir.

**Layer 2 Tunneling Protocol (L2TP):** Layer 2 Tunneling Protocol (L2TP), L2F ve PPTP yerine koymak için, Cisco ve Microsoft tarafından, oluşturulmuştur. L2TP, L2F ve PPTP'nin kapasitelerini, tek tünelleme protokolünde birleştirmiştir.

**Generic Routing Encapsulation (GRE):** Generic Routing Encapsulation (GRE), diğer bir Cisco tescilli tünelleme protokolüdür. IP tünellerinde değişik protokollerin enkapsüle edilmesine izin veren, sanal point-to-point linkleri oluşturur.

Şimdi VPN ve farklı VPN çeşitleri konusu kafanızda netleştiğinden, IPSec'e geçme zamanı geldi.

## Cisco IOS IPSec'e Giriş

IPSec, OSI modelinin 3.katmanında çalışan, IP tabanlı bir ağ boyunca güvenli veri aktarımına izin veren, protokol ve algoritmaların, endüstri standart ailesidir.

"IP-tabanlı ağ dediğimi fark ettiniz mi? Kendi başına IPSec'in, IP olmayan trafiği şifrelemek için kullanılamayacağından, bu gerçekten önemlidir. Yani, IP olmayan trafiği şifrelemek zorunda olduğunuz bir durumdaysanız, onun için GRE tüneli oluşturmanız ve sonra bu tüneli şifrelemek için IPSec kullanmanız gerekecektir.

## IPSec Dönüşümleri

IPSec transform, belirtilen güvenlik algoritması ile tek bir güvenlik protokolü tanımlar. Bu seçenekler olmadan, IPSec bu kadar kullanışlı olmayabilirdi. Bu teknolojilere aşina olmak çok önemlidir. Bu nedenle, güvenlik protokollerini açıklamama ve IPSec'in güvendiği, desteklenen şifreleme ve hashing algoritmalarını kısaca tanıtmama izin verin.

### Güvenlik Protokolleri

IPSec tarafından kullanılan iki ana güvenlik protokolü, Authentication Header (AH) ve Encapsulating Security Payloa'dur (ESP).

#### Authentication Header (AH)

AH protokolü, veri ve paket doğrulaması amacıyla tek-yönlü hash kullanan paketin IP başlığı için kimlik doğrulaması sağlar. Gönderici tek-yönlü hash üretir, sonra alıcı aynı tek-yönlü hash'ı üretir. Şayet paket herhangi bir şekilde değişirse, doğrulanmayacak ve atılacaktır. Böylece IPSec, güvenirliği garantilemek için AH'ye güvenir. AH, paketin tamamını kontrol eder, fakat herhangi bir şifreleme servisi önermez.

Bu, bir paketin verisinin sadece bütünlüğünün kontrolünü sağlayan ESP'den farklıdır.

#### Encapsulating Security Payload (ESP)

ESP size, borsanın ne zaman ve nasıl artacağını ve top gibi düşeceğini söylemeyecektir. Fakat ESP; gizlilik, veri kökenli kimlik doğrulama, bütünlük, anti-replay servisi ve trafik akışının analizini engelleyecek sınırlı trafik akış gizliliği sağlayacaktır.

**Gizlilik:** Gizlilik, DES veya 3DES gibi simetrik şifreleme algoritmaları kullanılmasıyla sağlanır. Gizlilik, diğer tüm servislerden ayrı olarak seçilebilir. Fakat seçilen gizlilik VPN'nin tüm uç noktalarında aynı olmalıdır.

**Veri kökenli kimlik doğrulama ve bütünlük:** Veri kökenli kimlik doğrulama ve bütünlük, opsiyonel gizlilik ile birlikte bir seçenek olarak önerilen ortak servislerdir.

**Anti-Replay servisi:** Anti-Replay servisi, sadece veri kökenli kimlik doğrulama seçildiğinde kullanabilirsiniz. Anti-Replay servisi alıcı tabanlıdır. Yani, servis sadece alıcı sıra numarasını kontrol ettiğinde etkilidir. Merak ediyorsanız, bir replay atağı, bir hacker'ın, onaylanmış bir paketi çaldığı ve sonra onu ilgili hedefe aktardığı zamanki ataktır. Onaylanmış, kopya ip paketi hedefe ulaştığında, servisleri ve diğer işlemleri aksatabilir. Sequence Number, alanı, bu tip atakları engellemek için tasarlanmıştır.

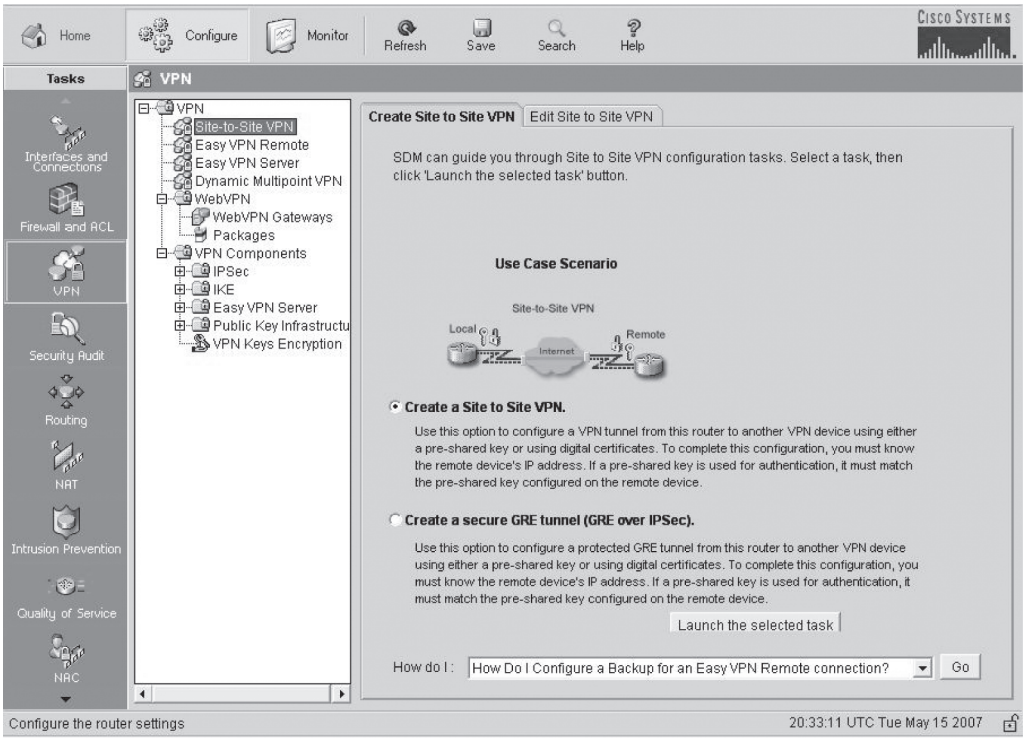
NOT *CCNA hedeflerinin kapsamından daha fazla bilgi edinmek için www.lammle.com adresine bakın.*

**Trafik akışı:** Trafik akışının gizliliğinin çalışması için tünel modu seçmiş olmanız gerekir. Çok sayıda trafiğin biriktiği güvenli bir gateway'de uygulandığında, en verimli şekilde çalışır.

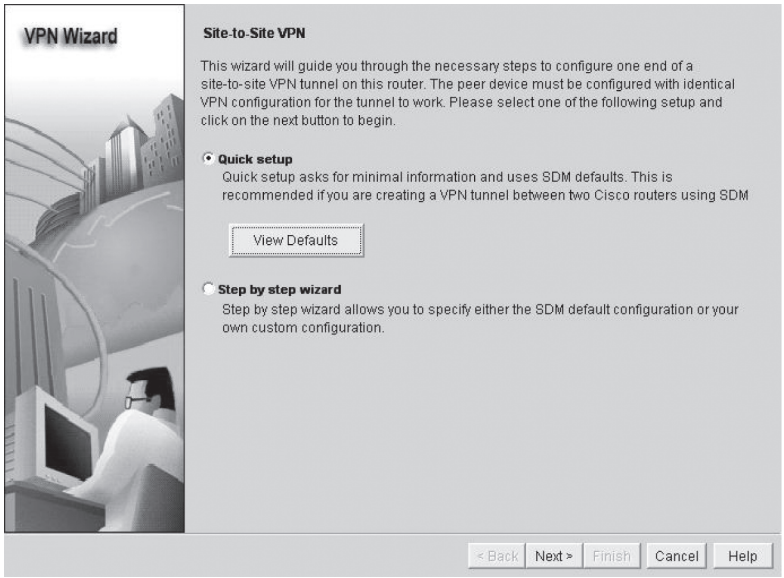
## SDM Kullanarak VPN/IPSec Yapılandırmak

Router'ınızdaki VPN'i yapılandırmanın birçok farklı yolu vardır. Burada, Corp ve R3 router'ları arasında düz bir VPN bağlantısı ekliyorum.

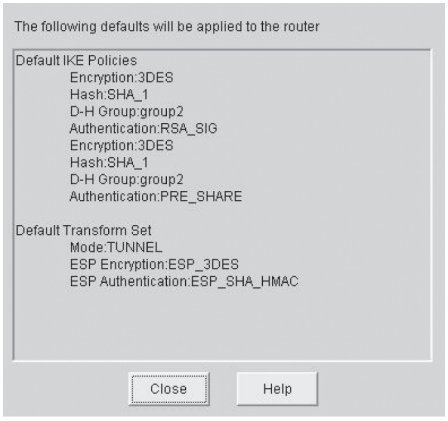
Tasks bar'daki VPN'e tıkladıktan sonra, Site-to-Site VPN'e tıkladım ve Create Site to Site VPN sekmesi geldi.



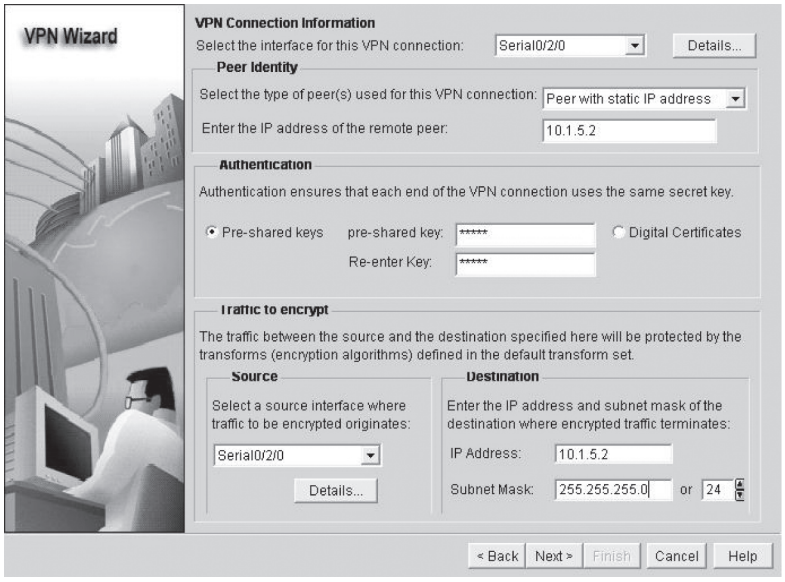

Create a Site to Site VPN'i seçtim ve sonra Site to Site VPN ekranı gelmesi için Launch the Selected Task'a tıkladım.



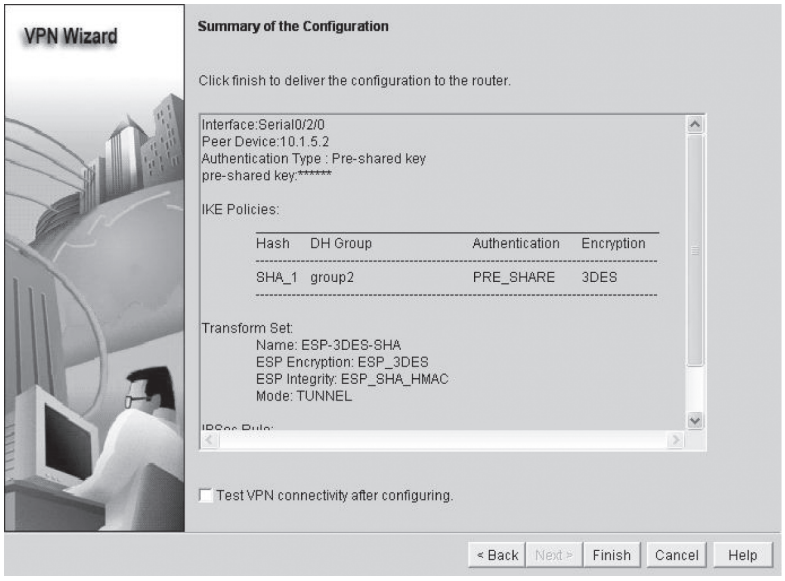

View Defaults'a tıkladım ve hangi router'ın yapılandırılacağına göz attım.




Close butonuna tıkladıktan sonra, VPN Connection Information ekranına erişmek için Next butonuna tıkladım.

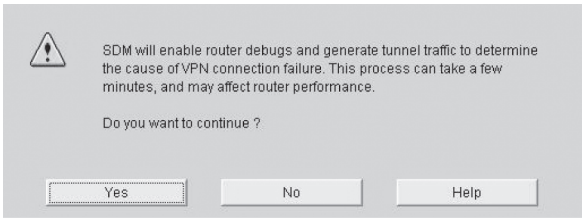



R3 router'a statik IP adresi verdim, bir pre-shared key ekledim, Cop router'ımın kaynak adresimi ve R3 router'ımınkiyle aynı adres olan hedef adresini seçtim. Sonra Next butonuna tıkladım.

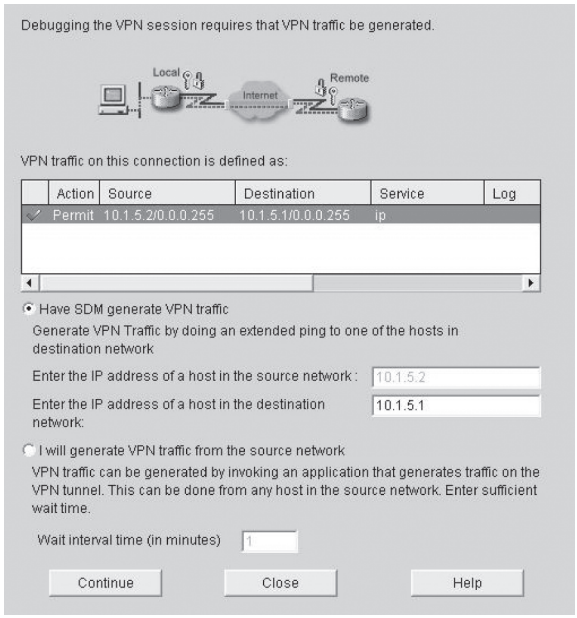


PSec çalışan VPN yapılandırmasının bir özetini aldım. SDM'den önce, varsayılan olarak VPN'leri, IPSec ile yapılandırmak zorundaydım. Bu çok daha kolaydır! Finish butonuna tıkladım.

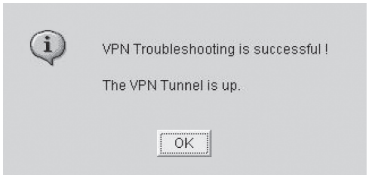
Şimdi en iyi bölüm geliyor. Bu son ekranı, SDM'den aldım.




VPN bağlantısını test etmek için OK butonuna tıklayıp tıklamayacağınız sormaktadır. Yes butonuna tıkladım tabi ki. SDM, kaynak ve hedef adresini soran başka bir ekran geldi ve trafik üretmem veya SDM'in üretmesini isteyip istemediğim soruldu. SDM'in yapmasını seçtim.




SDM linkte bir problem olduğunu ve SDM'in bunu çözebileceğini belirtti. Sonra, bu ekranı aldım.




Bir problem buldu ve benim için düzeltti. İleri seviyede teknik bir yardımcınız varmış gibi, çok güzel! SDM, sadece bu özelliği için dahi sahip olmaya değer. Corp router'ın `running-config`'ine bakalım ve router'ın yapılandırmasın için ne yüklediğini anlayalım:

```
!
crypto isakmp policy 1
 encr 3des
 authentication pre-share
 group 2
crypto isakmp key cisco address 10.1.5.2
!
!
crypto ipsec transform-set ESP-3DES-SHA esp-3des esp-sha-hmac
```

```
!
crypto map SDM_CMAP_1 1 ipsec-isakmp
 description Tunnel to10.1.5.2
 set peer 10.1.5.2
 set transform-set ESP-3DES-SHA
 match address 104
!
interface Serial0/2/0
[output cut]
 crypto map SDM_CMAP_1
!
access-list 104 remark SDM_ACL Category=4
access-list 104 remark IPSec Rule
access-list 104 permit ip 10.1.5.0 0.0.0.255 10.1.5.0 0.0.0.255
!
```

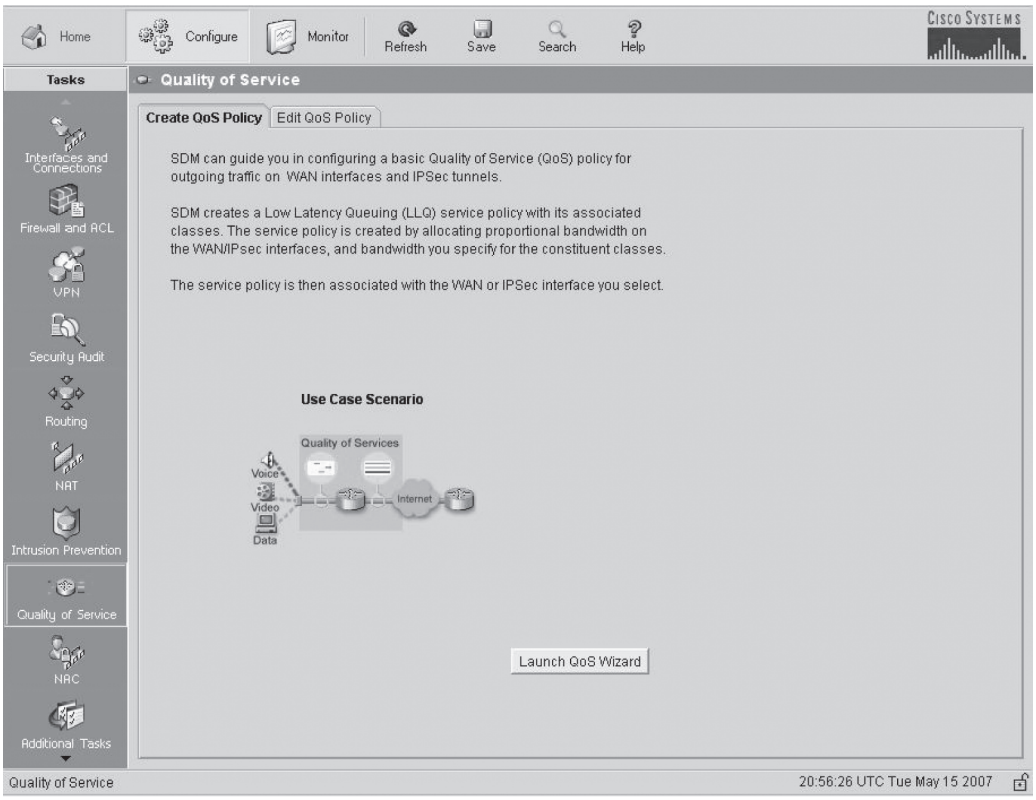
En zor yapılandırmaları SDM yardımıyla kolayca yapabildiğimize göre niye duralım? Şimdi bazı QoS ayarı yapalım.

## VPN Tünelimiz boyunca, Quality of Service (QoS) Yapılandırmak

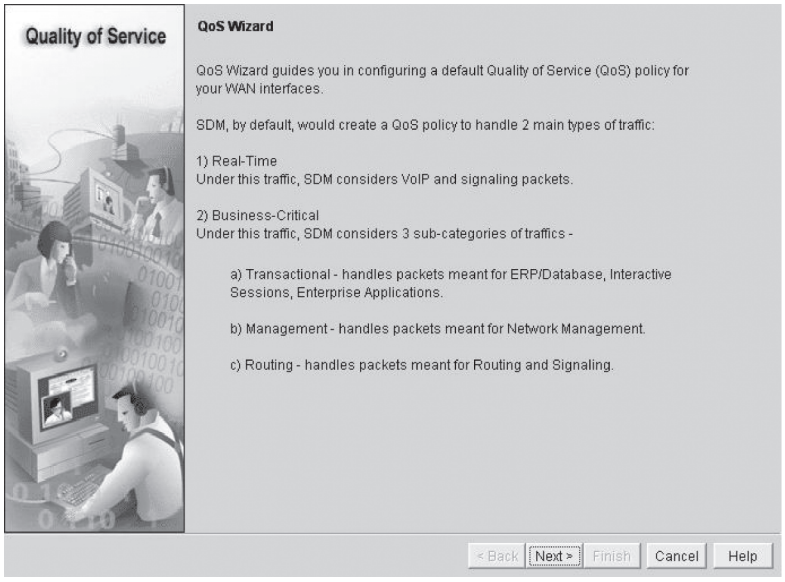
Müşteri ağlarına sahip olmamızın sebebi, uygulama ve kullanıcıların ihtiyaçlarını verimli şekilde karşılamaktır. Ayrıca, İnternet'in güçlü büyümesinin, tavşanlar gibi artan intranet'lerin beraber çalışmasının, çok bant genişliği harcayan yeni uygulamaların birçoğunun ve IP alt yapımızda dolaşan ses, veri ve video trafiğinin birleşmiş yüklerinin karışımının, onların tamamıyla en yüksek sınırına ulaştığını da biliyoruz. Aslında bu durumun sonuçlarını düşük performans ve güvensiz iletişimler şeklinde yaşıyoruz.

Bu bizi QoS'a götürür. İster inanın, ister inanmayın, QoS uygulamak gerçekten daha güvenli, düşük maliyetli, hatta günümüz ağ ortamlarında daha hızlıdır. Bundan dolayı, VPN seri linklerimizde QoS'u nasıl yapılandıracağımızı göstereceğim. Seri linklerimiz çalışan bir ağda, QoS yapılandırmak için iyi bir yerdir.

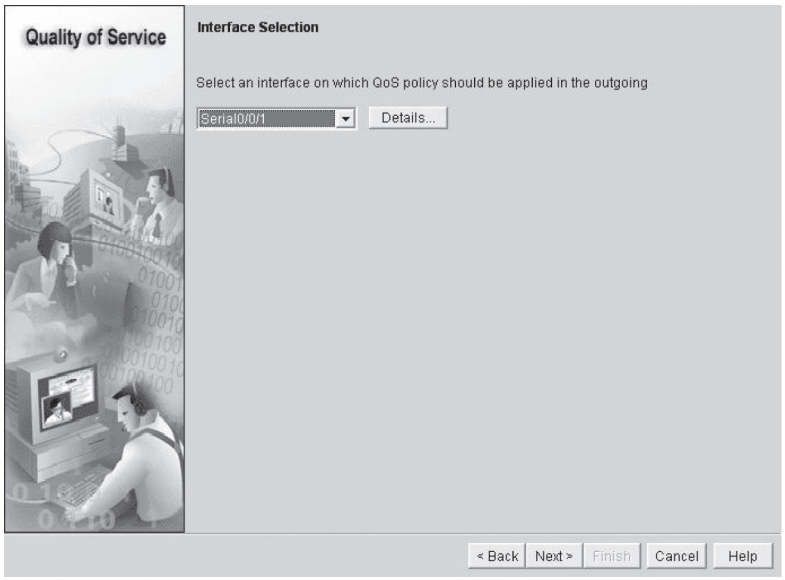
Bu sefer R3 router'ımdan başlayacağım. SDM ile bağlandıktan sonra Configure butonuna tıklayacağım ve sonra Tasks barı altındaki Quality of Service'e tıklayacağım.



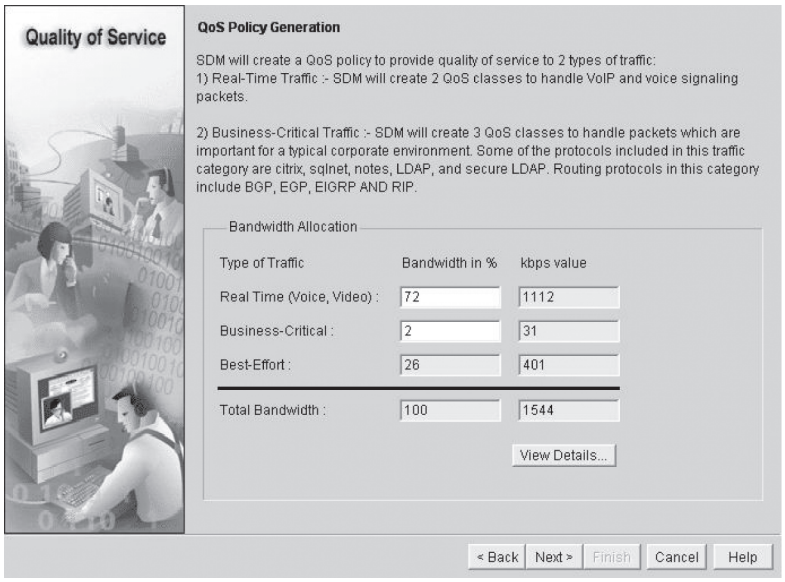

Buradan, sihirbazın ilk ekranına erişmek için,Launch QoS Wizard'a tıklayacağım.



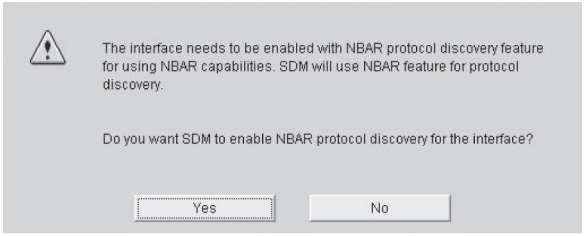

Next butonuna tıkladıktan sonra, kaynak veya çıkış portum olarak kullanmak istediğim interface'i seçeceğim ve sonra Next butonuna tıklayacağım.




QoS Policy Generation ekranından, çeşitli veri tipleri için bant genişliği tahsisi oluşturabilirim. Varsayılan olarak SDM, tipik bir ortam için üç QoS sınıfı oluşturur. Daha fazla bilgi sağlamak için View Details'e tıklayabilirsiniz. Next butonuna tıkladım.



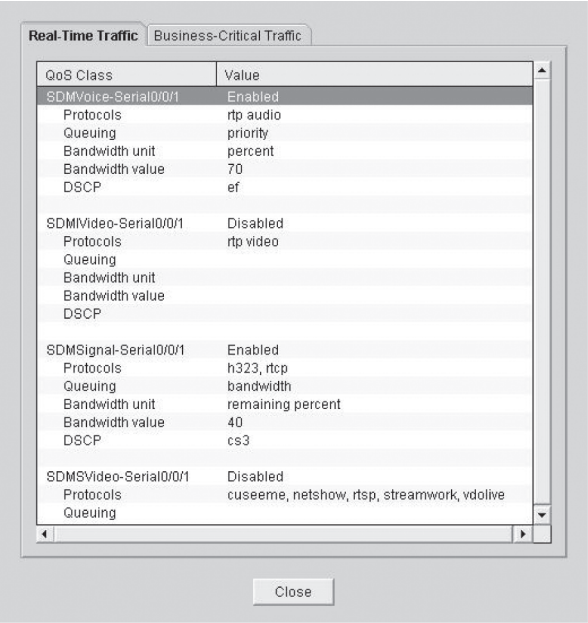

SDM, bir yolunu bulursa, ilerleyecek ve bu interface için NBAR protokol discovery özelliğini etkinleştirecektir.



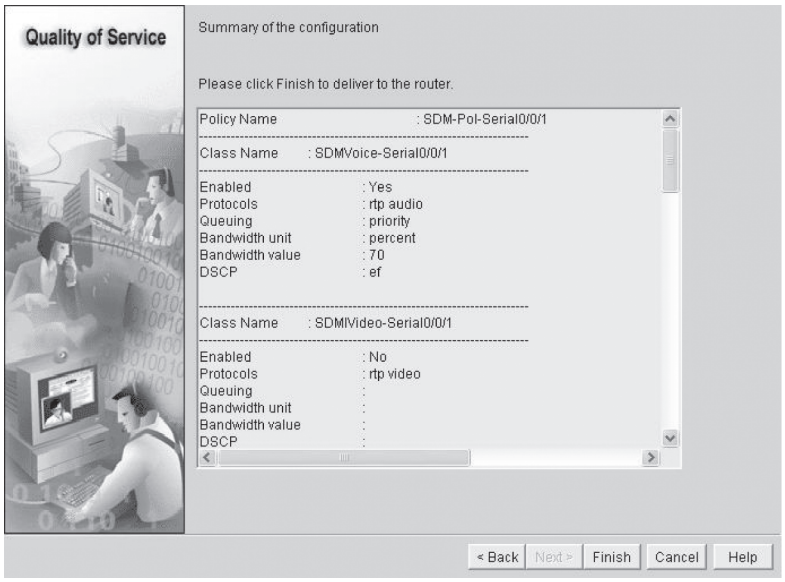

Network Based Application Recognition (NBAR), hem yaşamsal uygulamaların tanımlaması ve sınıflaması açısından hem de ERP gibi uygulamaların optimizasyonuna izin verdiğinden, oldukça iyidir. Bu sınıflandırılmış uygulamalara sahip olduktan sonra, onların gereksinim duyduğu minimum bant genişliğini kullanmalarını garanti etmelisiniz. Onlar basit olarak, policy'ler ile yönetile-

bilirler ve bu yaşamsal uygulamalar, sınıflandırılırlarsa minimum miktarda bant genişliği garanti edilebilir, ayrıca belli bir işlem için etiketlenebilirler.

Yes'e tıkladım ve sonraki ekrandan, hem QoS sınıfı ve gerçek zamanlı trafik ile iş için önemli trafik görüntülenir.




Close butonuna tıklayın. Yapılandırma özeti görünmektedir. Fakat çok uzun olduğundan, size sadece sayfanın üst kısmını göstereceğim.




Finish butonuna tıklayınca, router'a yapılandırmayı yükledi. Aslında, bu kitapta göstermek için çok fazla şey yükledi. SDM çok kullanışlı bir araçtır ve gösterdiğim gibi, bazı güçlü konfigürasyonlar için gerçekten verimli şekilde kullanılabilmektedir.

# Özet

Bu bölümde, WAN servislerini arasındaki farkları öğrendiniz: Kablo, DSL, HDLC, PPP, PPPoE ve Frame Relay. Ayrıca, bu servislerden biri çalışınca bir VPN bağlantısı kullanabileceğinizi de öğrendiniz.

High-Level Data-Link Control (HDLC) ve HDLC'nin etkin olduğunu, `show interface` komutu ile nasıl doğrulayacağınızı anlamak zorunda olduğunuzu söylemeliyim. Size, hem gerçekten önemli bazı HDLC bilgileri sağlandı hem de HDLC'nin önerdiğinden daha fazla özelliğe ihtiyacınız olduğunda veya iki farklı marka router kullandığınızda, point-to-point Protokol'ün (PPP) nasıl kullanıldığını gördünüz. Bundan dolayı Cisco'nun tescilli olduğunu ve farklı iki üreticinin router'ı arasında çalışmayacağını şimdi biliyorsunuz.

PPP bölümünü incelediğimde, hem farklı LCP seçeneklerini hem de kullanılabilecek iki farklı kimlik doğrulama yöntemini (PAP ve CHAP) işledim.

Frame Relay ve onunla kullanılan iki farklı enkapsülasyon yönteminden detaylı olarak bahsettim. Ayrıca, LMI seçenekleri, Frame Relay eşleşmeleri ve subinterface yapılandırmalarını da konuştuk. Frame Relay terim ve özelliklerine ilave olarak, Frame Relay yapılandırması ve doğrulamasını da derinlemesine gösterdim.

Bölümü, bir WAN linkini yapılandırmak için SDM'in nasıl kullanıldığını göstererek ve sonra VPN'ler ile bir VPN linkinin IPSec ile yapılandırılmasını işleyerek bitirdik.

# Sınav Gereklilikleri

**Cisco router'larda varsayılan seri enkapsülasyonu hatırlamak:** Cisco router'lar, varsayılan olarak, seri linklerinin tamamında, tescilli Level Data-Link Control (HDLC) enkapsülasyon kullanır.

**Farklı Frame Relay enkapsülasyonları anlamak:** Cisco, router'larında iki farklı Frame Relay enkapsülasyon yöntemi kullanır. Cisco varsayılandır ve router, Cisco Frame Relay switch'e bağlanıyor demektir. Internet Engineering Task Force (IETF), router'ınız herhangi bir Frame Relay switch'e bağlanıyor demektir.

**CIR'ın, Frame Relay'de ne olduğunu hatırlamak:** CIR, Frame Relay switch'in veri transferi için kabul ettiği, saniyede bit miktarı olarak ortalama değerdir.

**Frame Relay'i doğrulamak için kullanılan komutları hatırlamak:** `show frame-relay lmi` komutu, size lokal router ve Frame Relay switch arasında değiş tokuş edilen LMI trafiğini verir. `show frame pvc` komutu, yapılandırılmış tüm PVC ve DLCI numaralarını listeleyecektir.

**PPP Data Link katmanı protokollerini hatırlamak:** Üç Data Link katmanı protokolü, Network katman protokollerini belirten Network Control Protocol (NCP), noktadan-noktaya bağlantıyı oluşturup, yapılandırıp, koruyup, sonlandırmanın bir yöntemi olan Link Control Protocol (LCP) ve paketleri enkapsüle eden MAC katman protokolü olan, High-Level Data-Link Control'dür (HDLC).

**Farklı seri WAN tiplerini hatırlamak:** Çok yaygın olarak kullanılan seri WAN bağlantıları HDLC, PPP ve Frame Relay'dir.

**Virtual Private Network terimini anlamak:** İki yerleşim arasında bir VPN'nin neden ve nasıl kullanılacağını anlamanız gerekir.

# Yazılı Lab 14

Aşağıdaki WAN sorularının cevaplarını yazın:

1. Bir Cisco router'ın serial0'ındaki enkapsülasyon yöntemini görmek için kullanılan komutu yazın.
2. S0'ı, PPP enkapsülasyon ile yapılandırmak için gerekli komutu yazın.
3. PPP authentication için bir Cisco router'da kullanılan, `todd` kullanıcı adı ve `cisco` şifresi yapılandırması için gerekli komutları yazın.

4. Bir Cisco seri interface'de CHAP kimlik doğrulamayı etkinleştirmek için kullanılan komutları yazın.
5. İki seri interface (0 ve 1) için, DLCI numaraları yapılandırmak için kullanılan komutları yazın. S0 için 16, s1 için 17'yi kullanın.
6. Bir noktadan-noktaya subinterface kullanarak, uzak bir ofisi yapılandırmak için kullanılan komutları yazın. DLCI 16 ve IP adres 172.16.60.1/24'ü kullanın.
7. xDSL ve gerekli kimlik doğrulama çalıştırdığınızda, hangi protokolü kullanırsınız?
8. PPP'de belirtilen üç protokol nedir?
9. VPN tünelinizde güvenlik sağlamak için, hangi protokol ailesini kullanırsınız?
10. Tipik üç VPN kategorisi nedir?
11. (Yazılı Lab 14'ün cevapları, bu bölüm için gözden geçirme sorularına cevaptan sonra bulunabilir)

# Pratik Lab'lar

Bu bölümde, her lab'ın sağladığı şekilleri kullanarak, üç farklı WAN lab'ındaki Cisco router'ları yapılandıracaksınız (bu lab'lar, gerçek Cisco router'lar kullanmayı içerecektir).

Lab 14.1: PPP Enkapsülasyon ve Kimlik Doğrulama Yapılandırmak.

Lab 14.2: HDLC'yi Yapılandırmak ve Görüntülemek.

Lab 14.3: Frame Relay ve Subinterface'leri Yapılandırmak.

# Pratik Lab 14.1: PPP Enkapsülasyon ve Kimlik Doğrulama Yapılandırmak

Varsayılan olarak Cisco router'lar, seri linklerde noktadan noktaya enkapsülasyon yöntemi olarak High-Level Data-Link Control (HDLC) kullanır. Cisco olmayan bir ekipmana bağlanıyorsanız, iletişim için PPP enkapsülasyon yöntemi kullanabilirsiniz.

Yapılandıracağınız lab, aşağıdaki diyagramda gösterilmektedir.


1. Enkapsülasyon yöntemini görmek için RouterA ve RouterB'de **sh int s0** yazın.
2. Her router'da hostname ayarlandığından emin olun:

```
RouterA#config t
RouterA(config)#hostname RouterA

RouterB#config t
RouterB(config)#hostname RouterB
```

3. Her iki router'da varsayılan HDLC enkapsülasyonu PPP olarak değiştirmek için interface yapılandırmasında, `encapsulation` komutunu kullanın. Linklin her iki ucunda aynı enkapsülasyon yöntemi çalışmalıdır.

```
RouterA#Config t
RouterA(config)#int s0
RouterA(config-if)#Encap ppp
```

4. Şimdi RouterB'ye gidin ve serial0 interface'i PPP enkapsülasyona ayarlayın.

```
RouterB#config t
RouterB(config)#int s0
RouterB(config-if)#encap ppp
```

5. Her iki router'da `sh int s0` yazarak, konfigürasyonu doğrulayın.
6. IPCP, IPXCP ve CDPCP'ye dikkat edin. Bu, üst-katman (Network katmanı) bilgisini, MAC alt katmanındaki HDLC boyunca aktarmak için kullanılan bilgidir.
7. Her router'da kullanıcı adı ve şifre tanımlayın. Kullanıcı adının, uzak router'ın ismi olduğuna dikkat edin. Ayrıca, şifre aynı olmalıdır.

```
RouterA#config t
RouterA(config)#username RouterB password todd

RouterB#config t
RouterB(config)#username RouterA password todd
```

8. Her interface'de CHAP ve PAP kimlik doğrulamayı etkinleştirin.

```
RouterA(config)#int s0
RouterA(config-if)#ppp authentication chap

RouterB(config)#int s0
RouterB(config-if)#ppp authentication chap
```

9. Şu iki komutu kullanarak, her router'da PPP yapılandırmasının doğruluğunu kontrol edin.

```
sh int s0
debug ppp authentication
```

# Pratik Lab 14.2: HDLC'yi Yapılandırmak ve Görüntülemek

NOT *Bu lab, Lab 14.1'de kullanılan ile aynı konfigürasyonu kullanacaktır.*

Aslında HDLC için konfigürasyon yoktur, fakat Lab 14.1'i tamamladıysanız, her iki router'da PPP enkapsülasyon ayarlanmıştır. Bunun için ilk lab'a PPP koydum. Bu LAN, bir router'da HDLC enkapsülasyon yapılandırmanıza izin verecektir.

1. Her seri interface için `encapsulation hdlc` komutunu kullanarak enkapsülasyonu ayarlayın.

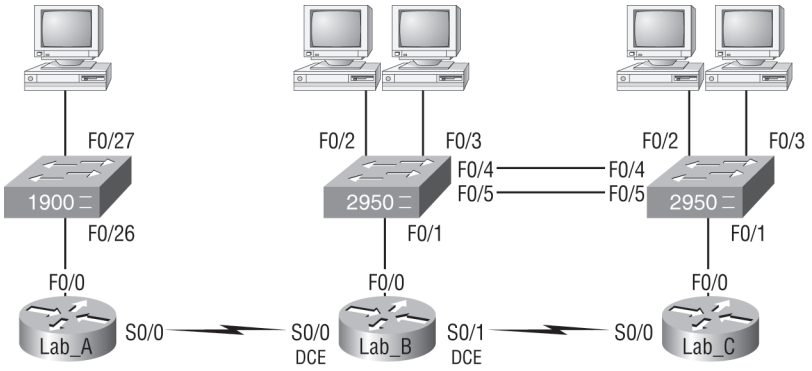
```
RouterA#config t
RouterA(config)#int s0
RouterA(config-if)#encapsulation hdlc
```

```
RouterB#config t
RouterB(config)#int s0
RouterB(config-if)#encapsulation hdlc
```

2. Her router'da `show interface s0` komutunu kullanarak HDLC enkapsülasyonu doğrulayın.

# Pratik Lab 14.3: Frame Relay ve Subinterface'leri Yapılandırmak

Bu lab'da, Frame Relay'i yapılandırmak için aşağıdaki şekli kullanacaksınız.




Bu lab'da, Lab_B router'ını bir Frame Relay switch olması için yapılandıracaksınız. Daha sonra, PVC için Lab_A ve Lab_C router'larını yapılandıracağım.

1. Hostname'i, `frame-relay switching` komutunu ve Frame Relay switch'teki her seri interface'in enkapsülasyonunu ayarlayın.

```
Router#config t
Router(config)#hostname Lab_B
Lab_B(config)#frame-relay switching [makes the router an
FR switch]
Lab_B(config)#int s0
Lab_B(config-if)#encapsulation frame-relay
Lab_B(config-if)#int s1
Lab_B(config-if)#encapsulation frame-relay
```

2. Her interface'de Frame Relay eşleşmelerini yapılandırın. Sadece bir interface'i diğerine, Frame Relay frame'leri ile anahtarladıklarından, bu interface'lerde IP adresine sahip olmak zorunda değilsiniz.

```
Lab_B(config-if)#int s0
Lab_B(config-if)#frame intf-type dce
[The above command makes this an FR DCE interface, which
is different than a router's interface being DCE]
Lab_B(config-if)#frame-relay route 102 interface
  Serial0/1 201
Lab_B(config-if)#clock rate 64000
[The above command is used if you have this as DCE, which
```

```
is different than an FR DCE]
Lab_B(config-if)#int s1
Lab_B(config-if)#frame intf-type dce
Lab_B(config-if)#frame-relay route 201 interface
  Serial0/0 102
Lab_B(config-if)#clock rate 64000 [if you have this as DCE]
```

2. Bu göründüğü kadar zor değildir. `route` komutu, PVC 102'den frame'ler aldıysanız, onları PVC 201'i kullanarak, `int s0/1`'e gönderin demektedir. `int s0/1`'den gelen herhangi bir şey, PVC 102 kullanarak serial0/0'dan gönderilecektir.
3. Lab_A'yı, point-to-point subinterface ile yapılandırın.

```
Router#config t
Router(config)#hostname Lab_A
Lab_A(config)#int s0
Lab_A(config-if)#encapsulation frame-relay
Lab_A(config-if)#int s0.102 point-to-point
Lab_A(config-if)#ip address 172.16.10.1
  255.255.255.0
Lab_A(config-if)#frame-relay interface-dlci 102
```

4. Lab_C'yi point-to-point subinterface ile yapılandırın.

```
Router#config t
Router(config)#hostname Lab_C
Lab_C(config)#int s0
Lab_C(config-if)#encapsulation frame-relay
Lab_C(config-if)#int s0.201 point-to-point
Lab_C(config-if)#ip address 172.16.10.2
  255.255.255.0
```

5. Konfigürasyonunuzu, aşağıdaki komutlarla doğrulayın:

```
Lab_A>sho frame ?
ip       show frame relay IP statistics
lmi      show frame relay lmi statistics
map      Frame-Relay map table
pvc      show frame relay pvc statistics
route    show frame relay route
traffic  Frame-Relay protocol statistics
```

6. Ayrıca, Ping ve Telnet'i bağlanırlığı kontrol etmek için kullanın.

# Gözden Geçirme Soruları

*Aşağıdaki sorular, bu bölümün materyallerini anladığınızı test etmek için tasarlanmıştır. İlave sorulara nasıl sahip olacağınızla ilgili daha fazla bilgi için bu kitabın Giriş bölümüne bakın.*

NOT

1. Hangi komut, ağdaki iki router arasında olan CHAP kimlik doğrulamasını gösterecektir?
   A. show chap authentication
   B. show interface serial 0
   C. debug ppp authentication
   D. debug chap authentication
2. Inverse ARP işlevsel değilse, bir Frame Relay ağda bağlanırlık için hangi komut gerekmektedir?
   A. frame-relay arp
   B. frame-relay map
   C. frame-relay interface-dci
   D. frame-relay lmi-type
3. Bir merkez ve altı şube ofisi olan bir müşteriniz olduğunuzu düşünün. Yakın gelecekte altı şube daha eklemeyi planlamaktadırlar. Şube ofislerinin merkeze ekonomik olarak bağlanmalarını sağlayacak ve merkezdeki yeterli boş portu olmayan router ile bir WAN teknolojisi oluşturmayı istiyorlar. Aşağıdakilerden hangisi tavsiye edilebilir?
   A. PPP
   B. HDLC
   C. Frame Relay
   D. ISDN
4. `Router#show frame-relay ?` komutu kullandığınızda, aşağıdaki seçeneklerden hangisi görüntülenmektedir?
   A. dlci
   B. neighbors
   C. lmi
   D. pvc
   E. map
5. Frame Relay ağında kullanılmakta olan bir router, routing güncellemelerini, split horizon sorunlarından korumak için, nasıl yapılandırılmalıdır?
   A. Her PVC için, benzersiz bir DLCI ile ayrı bir subinterface ve subinterface'e atanmış bir subnet yapılandır.
   B. Multicast ve broadcast trafiği desteklemesi için, her Frame Relay devresini, bir point-to-point hat olarak yapılandır.
   C. Aynı subnet'te çok sayıda subinterface yapılandır.
   D. Çoklu uzak router interface'lerine çoklu PVC bağlantıları kurmak için, tek bir subinterface yapılandır.

6. Bir seri interface'de hangi enkapsülasyonlar yapılandırılabilmektedir? (Üç şık seçin)
   A. Ethernet
   B. Token Ring
   C. HDLC
   D. Frame Relay
   E. PPP
7. Point-to-point subinterface'ler için Frame Relay kurduğunuzda, aşağıdakilerden hangisi yapılandırılmamalıdır?
   A. Fiziksel interface'de, Frame Relay enkapsülasyon
   B. Her subinterface'de, lokal DLCI
   C. Fiziksel interface'de bir IP adresi
   D. Point-to-point olarak subinterface tipi
8. Bir router, seri bir DTE interface'i kullanılarak, Frame Relay WAN linkine bağlandığında, clock rate nasıl belirlenir?
   A. CSU/DSU tarafından sağlanır.
   B. Uzak uç router tarafından.
   C. `clock rate` komutu ile.
   D. Physical katman bit stream timing tarafından.
9. Varsayılan bir Frame Relay WAN'ı, hangi fiziksel ağ tipiyle sınıflandırılır?
   A. Point-to-point
   B. Broadcast multi-access
   C. Non-broadcast multi-access
   D. Non-broadcast multipoint
10. Aşağıdakilerden hangisi, Ethernet frame'lerindeki PPP frame'lerini enkapsüle eder ve authentication, encryption ve compression gibi genel PPP özelliklerini kullanır?
    A. PPP
    B. PPPoA
    C. PPPoE
    D. Token Ring
11. Bir router'ı, Frame Relay bağlantı için, Cisco olmayan bir router'a bağlamanız gerekmektedir. Aşağıdaki komutlardan hangisi, router'ın WAN interface'ini bu bağlantı için hazırlayacaktır?
    A. Router(config-if)#encapsulation frame-relay q933a
    B. Router(config-if)#encapsulation frame-relay ansi
    C. Router(config-if)#encapsulation frame-relay ietf
    D. Router(config-if)#encapsulation frame-relay cisco
12. Acme Corporation, uzak ofis çalışanlarının, yerel ağa bağlanmasını sağlamak için, dial-up servisleri kurmaktadır. Şirket, çoklu routeable protokoller kullanmakta, kullanıcıların ağa bağlanması için, kimlik doğrulaması gerektirmekte ve bazı aramaların, uzak mesafe olmasından dolayı, callback desteği istemektedir. Aşağıdaki protokollerden hangisi, bu uzak servisler için en iyi seçimdir?

A. 802.1

B. Frame Relay

C. HDLC

D. PPP

E. PAP

13. Bir asenkron seri bağlantıda, hangi WAN enkapsülasyonları yapılandırılabilmektedir? (İki şık seçin.)

A. PPP

B. ATM

C. HDLC

D. SDLC

E. Frame Relay

14. Aşağıdakilerden hangisi, DSLAM'de sonlandırılan Data Link katman protokolü olarak, ATM kullanır?

A. DSL

B. PPPoE

C. Frame Relay

D. Dedicated T1

E. Wireless

F. POTS

15. Corp router ile Remote router arasındaki seri link neden çalışmamaktadır?

```
Corp#sh int s0/0
Serial0/0 is up, line protocol is down
  Hardware is PowerQUICC Serial
  Internet address is 10.0.1.1/24
  MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec,
     reliability 254/255, txload 1/255, rxload 1/255
  Encapsulation PPP, loopback not set

Remote#sh int s0/0
Serial0/0 is up, line protocol is down
  Hardware is PowerQUICC Serial
  Internet address is 10.0.1.2/24
  MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec,
     reliability 254/255, txload 1/255, rxload 1/255

 Encapsulation HDLC, loopback not set
```

A. Seri kablo arızalıdır.

B. IP adresleri aynı subnette değildir.

C. Subnet mask'ları doğru değildir.

D. Keepalive ayarları doğru değildir.

E. Katman 2 frame tipleri uyumlu değildir.

16. Aşağıdaki teknolojilerden hangisi, HFC terimini kullanır?

A. DSL

B. PPPoE

C. Frame Relay

D. Cable

E. Wireless

F. POTS

17. Uzak bir yerleşim, merkez ofise bağlanmıştır. Bununla beraber, uzak kullanıcılar, merkez ofisteki uygulamalara erişememektedir. Uzak kullanıcılara, merkez ofis router'ından ping atılabilmektedir. Aşağıdaki komut çıktısına göz attıktan sonra, bu problem için muhtemel sebebin ne olduğunu düşünürsünüz?

```
Central#show running-config
!
interface Serial0
 ip address 10.0.8.1 255.255.248.0
 encapsulation frame-relay
 frame-relay map ip 10.0.15.2 200
!
Router rip
Network 10.0.0.0

Remote#show running-config
!
interface Serial0
 ip address 10.0.15.2 255.255.248.0
 encapsulation frame-relay
 frame-relay map ip 10.0.8.1 100
!
Router rip
Network 10.0.0.0
```

A. Frame Relay PVC, arızalıdır.

B. Merkez/Uzak router linki arasındaki IP adreslemesi yanlıştır.

C. RIP routing bilgisi, gönderilmiyordur.

D. Frame Relay Inverse ARP, düzgün şekilde yapılandırılmamıştır

18. Aşağıdakilerden hangisi, OSI modelinin Network katmanında çalışan IP-tabanlı ağ boyunca güvenli veri aktarımı sağlayan, protokol ve algoritmaların bir endüstri standart ailesidir?

    A. HDLC

    B. Cable

    C. VPN

    D. IPSec

    E. xDSL

19. Aşağıdakilerden hangisi, TCP/IP olmayan protokollerin tünellenmesini ve gizliliğini mümkün kılarak, internet üzerinde özel ağların oluşturulmasını sağlar?

    A. HDLC

    B. Cable

    C. VPN

    D. IPSec

    E. xDSL

20. Aşağıdaki şekli referans alarak, Frame Relay DLCI, RouterA ile ilgili hangi fonksiyonu sağlar?


    A. RouterA ile Frame Relay switch arasındaki sinyalleşme standardını tanımlar.

    B. RouterB ve Frame Relay switch arasındaki devreyi tanımlar.

    C. RouterA ve RouterB arasında kullanılan enkapsülasyonu tanımlar.

    D. RouterB ile Frame Relay switch arasındaki sinyalleşme standardını tanımlar.

# Gözden Geçirme Sorularının Cevapları

1. C `debug ppp authentication` komutu size, point-to-point bağlantılar arasında, PPP'nin kullandığı kimlik doğrulama prosesini gösterecektir.
2. B şayet Frame Relay ağınızda, IARP'ı desteklemeyen bir router'ınız varsa, router'ınızda, bilinen DLCI'ı, IP adres eşleşmesi sağlayan Frame Relay eşleşmeleri oluşturmanız gerekir.
3. C Anahtar cümle, router'ınızda "yeterli boş port olmadığı''dır. Sadece Frame Relay, bir interface ile çoklu lokasyonlara bir bağlantı sağlayabilir.
4. C,D,E `show frame-relay ?` komutu birçok seçenek sağlar. Fakat bu sorudaki uygun seçenekler, lmi, pvc ve map'tir.
5. A Çoklu uzak yerleşimlere bağlı çoklu DLCI'lar ile yapılandırılmış bir seri port'a sahipseniz, split horizon kuralı (Modül 5'de anlatıldı), route güncellemelerinin, alınan interface'den gönderilmesini durdurur. Her PVC için subinterface oluşturarak, Frame Relay kullandığınızda, split horizon sorunlarından kaçınabilirsiniz.
6. C,D,E Ethernet ve Token Ring, LAN teknolojileridir ve bir seri interface'de yapılandırılamazlar. PPP, HDLC ve Frame Relay, tipik olarak bir seri interface'de yapılandırılan, katman2 WAN teknolojileridir.
7. C CCNA sınav konularına çalıştığınızda ve point-to-point subinterface'lerle bir Frame Relay yapılandırdığınızda, fiziksel interface'de bir IP adresi koyamayacağınızı hatırlamanız çok önemlidir.
8. A Bir seri interface'de saat denetimi, daima CSU/DSU (DCE cihazı) tarafından sağlanır. Bununla beraber, test ortamınızda bir CSU/DSU'e sahip değilseniz, o zaman DCE kablo bağlı router'ın seri interface'inde `clock rate` komutu ile saat denetimi sağlamanız gerekmektedir.
9. C Varsayılan olarak, Frame Relay, bir non-broadcast multi-access (NBMA) ağdır. Yani, RIP güncellemeleri gibi broadcast'ler, varsayılan olarak link boyunca iletilmeyecektir.
10. C PPPoE, Ethernet frame'lerindeki PPP frame'lerini enkapsüle eder ve authentication, encryption ve compression gibi genel PPP özelliklerini kullanır. PPPoA, ATM tarafından kullanılır.
11. C Frame Relay ağının bir tarafında Cisco router, diğer tarafında Cisco olmayan bir router'a sahipseniz, IETF enkapsülasyon yöntemini kullanmanız gerekmektedir. Cisco enkapsülasyon, varsayılandır. Yani, Frame Relay ağının her iki tarafında bir Cisco router olmalıdır.
12. D HDLC ve Frame Relay, bu iş gereklilik tiplerini desteklemediğinden, PPP tek seçeneğinizdir. PPP, dinamik adresleme, PAP ile CHAP kullanarak authentication ve callback servisleri sağlar.
13. A,B ATM cevabı için heyecanlanmayın. CCNA sınavları, ATM'i derinlemesine içermemektedir. PPP, dial-up (asenkron) servisleri için çoğunlukla kullanılmaktadır. PPP'nin oldukça verimli olmasından dolayı, aslında artık kullanılmamasına rağmen, ATM de kullanılabilir.
14. A ATM, CPE'den, DSL katman1 bağlantısı boyunca tipik olarak kullanılan Data Link katman protokolüdür ve DSL interface kartları (ATU-C'ler) içeren ATM switch olan DSLAM'de sonlandırılırlar.
15. E Remote router, varsayılan HDLC seri enkapsülasyon ve Corp router, PPP seri enkapsülasyon kullandığından, bu kolay bir sorudur. Remote router'a gitmek ve enkapsülasyonu, PPP olarak ayarlamak veya Corp router'ınkini HDLC'ye değiştirmeniz gerekir.
16. D Modern bir ağda, hybrid fibre-coaxial (HFC), broadband bir ağ oluşturmak için, fiber optik ve koaksiyel kablo'nun birlikte kullanıldığı bir telekomünikasyon endüstri terimidir.

17. C IP adresi doğru olarak görünmese de, onlar aynı subnettedir, bu nedenle B şıkkı doğru değildir. Soru, diğer tarafa ping atabildiğinizi belirtmektedir. Öyleyse, PVC aktiftir, yani A şıkkı doğru olamaz. IARP'ı yapılandıramazsınız. Bu nedenle sadece C şıkı doğru olabilir. Frame Relay ağı, varsayılan olarak bir non-broadcast multi-access network olduğundan, RIP güncellemeleri gibi broadcast'ler, `frame-relay map` komutunun sonunda, broadcast ifadesini kullanmadığınız sürece, PVC boyunca gönderilmeyecektir.
18. D IPSec, OSI modelinin katman3 Network katmanında çalışan IP-tabanlı ağ boyunca güvenli veri aktarımı sağlayan, protokol ve algoritmaların bir endüstri standartları ailesidir.
19. C Bir VPN, TCP/IP olmayan protokollerin tünellenmesi ve gizliliği mümkün kılarak, İnternet üzerinde özel ağların oluşturulmasını sağlar. Bir VPN, herhangi bir link tipi boyunca kurulabilir.
20. A Bu bölümde birçok kez belirttiğim gibi, şunu hatırlamanız gerekir: DLCI'lar sadece lokal olarak önemlidir ve sadece router'dan, switch'e devreyi belirtir. RouterA, RouterB ağlarına sahip olmak için, DLCI 100 kullanır. RouterB, RouterA ağlarına sahip olmak için DLCI 200 kullanır.

# Yazılı Lab 11'in Cevapları

1. `sh int s0`
2. `config t`
   `int s0`
   `encap ppp`
3. config t
   **`username todd password cisco`**
4. config t
   **`int bri0`**
   **`ppp authentication chap`**
5. config t
   **`int s0`**
   **`frame interface-dlci 16`**
   **`int s1`**
   **`frame interface-dlci 17`**
6. config t
   **`int s0`**
   **`encap frame`**
   **`int s0.16 point-to-point`**
   **`ip address 172.16.60.1 255.255.255.0`**
   **`frame interface-dlci 16`**
7. PPPoE
8. HDLC, LCP ve NCP
9. IPSec
10. Remote access VPN'ler, site-to-site VPN'ler ve extranet VPN'ler.

# A Terimler Sözlüğü

# Terimler Sözlüğü

**100BaseT** IEEE 802.3u standartına bağlı 100BaseT, UTP kablolamada kullanan, 100 Mbps bant tabanlı Ethernet özelliğidir. 100BaseT, trafik olmadığında, ağ üzerinde (10BaseT'de kullanılandan daha fazla bilgi içeren) link pulsları gönderir*. Ayrıca bakınız: 10BaseT, Fast Ethernet ve IEEE 802.3.*

**100BaseTX** IEEE 802.3u standartına bağlı 100BaseTX, UTP ve STP kabloların iki çiftini de kullanan, 100 Mbps bant tabanlı Fast Ethernet özelliğidir. Bir tel çifti veri alır, diğer çift gönderir. Doğru sinyal zamanlaması için 100BaseTX segmenti 100 metreden uzun olamaz.

**10BaseT** Orjinal IEEE 802.3 standardının bir parçası olan 10BaseT, Kategori 3, 4 veya 5 sarmal-çift kablonun, biri veri göndermek diğeri almak için her iki çiftini kullanan, 10 Mbps bant tabanlı Ethernet özelliğidir. *Ayrıca bakınız: Ethernet ve IEEE 802.3.*

**A&B bit sinyalleşmesi** T1 iletişim hizmetlerinde kullanılır ve bazen "24 kanal sinyalleşme" olarak tanımlanır. Bu prosedürdeki 24 T1 altkanalın her biri, kontrol sinyal bilgisini göndermek için her altıncı frame'in bir bit'ini kullanır.

**AAA** Authentication, Authorization, and Accounting: Ağ güvenliği sağlamak için Cisco tarafından geliştirilen bir sistem. *Ayrıca bakınız*: Authentication, Authorization, and Accounting

**AAL** ATM Adaptation Layer: Diğer uygulamalardan veri kabul eden ve onu 48-byte ATM veri segmentindeki ATM katmanına getiren, Data Link katmanının, servis bağımlı bir alt katmanıdır. CS ve SAR, AAL'lere şekil veren iki alt katmandır. Günümüzde, ITU-T tarafından tavsiye edilen dört tip AAL; AAL1, AAL2, AAL3/4 ve AAL5'tir. Kullandıkları kaynak-hedef zamanlaması, CBR veya VBR olmaları ve connection-oriented veya connectionless mod veri iletimi için kullanılmasına göre farklılıklar gösterirler. *Ayrıca bakınız:* AAL1, AAL2, AAL3/4, AAL5, ATM ve ATM katmanı.

**AAL1** ATM Adaptation Layer 1: ITU-T tarafından tavsiye edilen dört AAL'den biri olan AAL1, sabit zamanlı trafik ve sıkıştırılmamış video gibi sabit bit değerlerine ihtiyaç duyan connection-oriented, zaman-duyarlı servisler için kullanılır. *Ayrıca bakınız: AAL*

**AAL2** ATM Adaptation Layer 2: ITU-T tarafından tavsiye edilen dört AAL'den biri olan AAL2, sıkıştırılmış ses trafiği gibi, değişken bir bit değerini sağlayan connection-oriented servisler için kullanılır. *Ayrıca bakınız: AAL*

**AAL3/4** ATM Adaptation Layer 3/4: ITU-T tarafından tavsiye edilen dört AAL'den biri (başlangıçta farklı iki katmanın birleşimi) olan AAL3/4, connection-oriented ve connectionless hatların ikisini de destekler. İkisinin öncelikli kullanımı, ATM ağlarında SMDS paketleri gönderilmesidir. *Ayrıca bakınız: AAL*

**AAL5** ATM Adaptation Layer 5: ITU-T tarafından tavsiye edilen dört AAL'den biri olan AAL5, ATM ve LANE trafikleri üzerinde, öncelikle, klasik IP transferi için connection-oriented VBR servislerini desteklemek için kullanılır. Daha düşük bant genişliği maliyeti ve daha basit işleyiş gereksinimleri öneren, fakat aynı zamanda düşük bant genişliği ve hata-giderme kapasitesi sağlayan, en az karmaşık AAL türü olan AAL5, SEAL kullanır. *Ayrıca bakınız: AAL*

**AARP** AppleTalk Address Resolution Protocol: Ağ adreslerini, data-link adreslerine eşleştiren, AppleTalk yığınındaki bir protokoldür.

**AARP arama paketleri** Verilen bir düğüm ID'sinin, genişlememiş bir AppleTalk ağındaki diğer düğüm tarafından kullanılıp kullanılmadığını tespit etmek için AARP tarafından kullanılan paketler. Şayet düğüm ID'si kullanımda ise, düğüm başka bir ID seçecek ve daha fazla AARP arama paketi gönderecektir. Ayrıca bakınız: AARP

**ABM** Asynchronous Balanced Mode: İki istasyon bir aktarım başlattığında, ABM, bu istasyonlar arasında, eşdüzeysel ya da noktadan-noktaya bağlantıları destekleyen bir HDLC (ya da onun türevi protokoller) iletişim teknolojisidir.

**ABR** Area Border Router: Bir ya da daha fazla OSPF alanının sınırında bulunan OSPF router'ıdır. ABR'lar OSPF alanlarını, OSPF omurga alanına bağlamak için kullanılmaktadır.

**access hattı** Sadece bir sanal VAN'a (VLAN) ait switch'lerin kullandığı bir hat. Trunk hatları çoklu VLAN'lerden bilgi taşırlar.

**access hızı** Devrenin bant genişliği değerini belirtir. Örneğin, bir T1 devresi için erişim değeri 1.544Mbps'dir. Frame Relay ya da diğer teknolojilerde, örnek olarak 256Kbps gibi, bölünmüş bir T1 bağlantısı olabilir. Bununla beraber, erişim ve saat değeri hala 1.544Mbps'dir.

**access katmanı (erişim katmanı)** Cisco'nun üç-katmanlı hiyerarşik modelindeki katmanlardan biri. Erişim katmanı, kullanıcılara ağlar topluluğuna erişim sağlamaktadır.

**access list** Ağdaki çeşitli servisler için router'a giden ya da router'dan gelen ilgili trafiği tespit etmekte router'lar tarafından tutulan test durumlarının bir grubudur.

**access metodu** Ağ cihazlarının, ağın kendisine erişim sağlamak için ele aldıkları yöntem.

**access sunucusu** Ağ erişim sunucusu olarak da bilinen erişim sunucusu, asenkron cihazları, ağ ve terminal emülasyon yazılımları yardımıyla, bir LAN ya da WAN'a bağlamak ve desteklenen protokollere senkron ya da asenkron yönlendirme sağlamakta kullanılan bir iletişim prosesidir.

**Accounting** AAA'daki üç bileşenden biridir. Kimlik bilgileri tutma, güvenlik modelleri için denetim ve log tutma olanakları sağlar.

**Acknowledgment** Bir işlemin yapıldığı anlamında, bir ağ cihazından diğerine gönderilen doğrulamadır. ACK olarak kısaltılabilir*. Tersidir: NAK*

**ACR** Allowed cell rate: ATM trafiğini yönetmek için, ATM forumu tarafından tanımlanan bir değer. Tıkanıklığı kontrol ölçümleri ile dinamik olarak denetlenir. ACR minimum hücre değeri (MCR) ve en yüksek hücre değeri (PCR) arasında değişir. *Ayrıca bakınız: MCR ve PCR*

**Adjacency** Yaygın bir medya segmenti kullanan uç düğümler ve tanımlı komşu router'lar arasında routing bilgilerinin alışverişini yapmak için sağlanan ilişki.

**administrative distance (AD) – yönetimsel uzaklık** Bir routing bilgi kaynağının güvenilirlik seviyesini gösteren 0 ile 255 arasında bir numaradır. Daha yüksek güvenilirlik değeri için daha düşük numara kullanılır.

**administrative weight - yönetimsel ağırlık** Bir ağ hattına öncelik tanımlamak için ağ yöneticisi tarafından atanan bir değer. ATM ağ kaynak kullanılabilirliğini kontrol etmek için PTSP'ler tarafından değiş tokuş edilen dört hat metriğinden biridir.

**Adres çözümlemesi** Bilgisayar adresleme tasarımları arasındaki farklılıkları çözmek için kullanılan işleyişlerdir. Adres çözümlemesi, tipik olarak network katmanı (3.katman) adreslerini data link katmanı (2.katman) adresine eşleştirmek için bir yöntem belirtir. *Ayrıca bakınız: Adres eşitleme*

**adres mapping** Ağ adreslerini bir formattan diğerine çevirmedir. Bu metodoloji, farklı protokollere birbirlerinin yerine geçebilecek şekilde çalışma izni verir.

**Adres maskesi** Bir adresin, hangi parçasının ağı ya da subnet'i, hangisinin kullanıcıyı gösterdiğini belirleyen bir bit kombinasyonudur. Bazen sadece maske olarak tanımlanır. *Ayrıca bakınız: Subnet mask.*

**Adres öğrenme** Transparent bridge'ler, bir ağdaki tüm cihazların donanım adresini öğrenmek için kullanılırlar. Bundan sonra, switch'ler, donanım adresleri (MAC) bilinen ağı filterelerler.

**ADSU** ATM Data Service Unit: HSSI-uyumlu bir mekanizma yardımıyla bir ATM ağına bağlanmak için kullanılan terminal adaptörüdür.

**advertising** Routing ve servis güncellemelerinin, verilen sürelerde iletildiği, ağdaki diğer router'lara, uygun route kaydını sağlamayı mümkün kılan bir işleyiştir

**AEP** AppleTalk Echo Protocol: Birinin diğerine bir paket gönderdiği ve diğerinin cevap olarak bir eko ya da kopya aldığı iki AppleTalk düğümü arasındaki bağlantı için yapılan bir test.

**AFI** Authority and Format Identifier: Bir ATM adresinin IDI bölümünün tip ve formatını tanımlayan bir NSAP ATM adresinin parçasıdır.

**AFP** AppleTalk Filing Protocol: Kullanıcılara, bir sunucudaki dosya ve uygulamalar için paylaşımı mümkün kılan, AppleShare ve Mac OS dosya paylaşımını destekleyen, bir sunum katmanı protokolüdür

**ağ segmentasyonu** Geniş ağları daha küçük ağlara ayırmaktır. Router, switch ve bridge'ler ağ segmentasyonu için kullanılmaktadır.

**AIP (ATM Interface Processor):** AAL3/4 ve AAL5i destekleyen Cisco7000 serisi router arayüzüdür, UNI'deki performans darboğazlarını minimuma indirmektedir.

**Akış kontrolü** Gönderilen cihazdan veriyle alan birimin ambale olmadığından emin olmak için kullanılan bir metodoloji. IBM ağlarında bilindiği şekliyle, Pacing (hız denetimi) şu anlama gelmektedir; alan bir birimdeki arabellekler dolduğunda, alınan önbellekteki verinin tamamı işlenene ve önbelleğin çalışmak için tekrar hazır olana kadar aktarımları geçici olarak durdurmak için alınan birime bir mesaj aktarılır.

**Aktif durum** Bir EIGRP routing tablosuyla ilgili bir router kullanılamadığında, route aktif duruma geçecektir.

**Aktif ekran** Token ringi yönetmek için kullanılan bir mekanizmadır. Ring'teki en yüksek MAC adrese sahip ağ düğümü aktif ekran olur ve döngüleri engellemek ve token'ların kaybolmamasından emin olmak gibi yönetimsel işlerden sorumludur.

**Algoritma** Bir problemi çözmek için yapılan işlemler veya kuralların bir bütünüdür. Ağ kurulumunda, algoritmalar tipik olarak, bir kaynaktan hedefe trafik için en iyi route'u bulmak için kullanılmaktadır.

**alignment error – Hizalama hatası** Sekize bölünemeyen sayıda ekstra bit'e sahip bir frame alan Ethernet ağlarında oluşan bir hatadır. Hizalama hataları genelde, çarpışmaların (collision) sebep olduğu frame hasarlarının neticesidir.

**all-routes explorer paketi** Tüm SRB ağı boyunca hareket edebilen, belirli bir hedefe olası tüm yolları bulan bir explorer paketidir. *Ayrıca bakınız:* Explorer paketi, lokal explorer paketi ve spanning explorer paketi.

**AM** Amplitude modulation: Taşıyıcı sinyal genliğinin değişmesiyle oluşan veriyi gösteren bir modülasyon yöntemidir. *Ayrıca bakınız:* Modülasyon

**AMI** Alternate Mark Inversion: Her bir bit hücresi boyunca sıfırları 01 olarak gösteren ve her bir bit hücresi boyunca, dönüşümlü olarak, birleri 11 veya 00 olarak gösteren T1 ve E1 devrelerindeki bir bağlantı-kodu tipi. Gönderen cihaz, AMI'deki birlerin yoğunluğunu, veri akımına bağlı olarak korumak zorundadır. Ayrıca, ikili-kod, değişken işaret evirmesi olarak da bilinir. *B8ZS'nin tersidir. Ayrıca bakınız: Birlerin yoğunluğu*.

**Anahtarlamalı LAN** LAN switch'ler kullanarak gerçekleştirilen bir LAN. *Ayrıca bakınız: LAN switch.*

**Analog aktarım** Bilginin, sinyal genliği, frekansı ve fazının çeşitli kombinasyonlarda gösterildiği sinyal mesajlaşması.

**ANSI** Amerikan Ulusal Standartları Enstitüsü: Standartlarla ilgili aktiviteleri koordine eden, U.S. ulusal standartdartlarını onaylayan ve uluslararası standart organizasyonlarındaki U.S. konumunu geliştiren, kuruluş, hükümet ve gönüllü üyelerden oluşan organizasyon. ANSI, iletişim, ağ kurulumu ve çeşitli teknik alanlar gibi bilgi dallarındaki uluslararası ve U.S. standartlarının oluşturulmasına yardım eder. Vida yivinden ağ kurulumuna kadar üretim ve teknoloji alanında 13.000'in üzerinde standart yayını vardır. ANSI, International Electrotechnical Commission (IEC) ve International Organization for Standardization (ISO) üyesidir.

**Anycast** Birden fazla uç sistem tarafından paylaşılabilen bir ATM adresidir, istekleri, belirli bir servisi sağlayan düğüme yönlendirmeye izin verir.

**AppleTalk** Güncel iki versiyonu olan, Macintosh ortamında kullanmak için Apple Computer tarafından dizayn edilen iletişim protokollerinin bütünüdür. Önceki Faz 1 protokolleri, bir bölgede bulunan tek bir ağ numarasına sahip fiziksel ağı destekler. Sonraki Faz 2 protokolleri, tek fiziksel ağdaki birden fazla mantıksal ağı destekler ve ağların birden fazla bölgede bulunmasına izin verir. Ayrıca bakınız: zone.

**Application katmanı** OSI modelinin dışındaki (elektronik posta ve dosya transferi gibi) uygulama prosedürlerine servisler sağlayan, OSI referans ağ modelinin 7. katmanıdır. Bu katman, bağlantıyı yapmak için gerekli kaynaklarla iletişim ortaklarının uygunluğunu saptar ve seçer, ortaklık uygulamalarını koordine eder ve veri bütünlüğünün kontrolü ve hata giderimi için prosedürlerde bir görüş birliği oluşturur. *Ayrıca bakınız: Data Link Katmanı, Network katmanı, Physical katman, Presentation katmanı, Session katmanı ve Transport katmanı.*

**ARA** AppleTalk Remote Access: Uzak bir AppleTalk lokasyonundan, kaynak ve bilgilere erişim kurmaları için Macintosh kullanıcılarının kullandığı bir protokol.

**arama giriş kontrolü** ATM ağlarındaki trafiği yönetmek için kullanılan bir cihazdır. İstenen bir VCC için uygun bant genişliği içeren bir yolun uygunluğunu saptar.

**Arama kurulum zamanı** DTE cihazları arasındaki bir anahtar aramasını yerine getirmek için gerekli zaman miktarı.

**Arama kurulumu** Bir kaynak ve hedef cihazının aralarında nasıl bir bağlantı kurulacağını açıklayan anlaşma planı

**Arama önceliği** Devre-anahtarlı sistemlerde, her oluşturulan porta verilen önceliklerin tanımlanmasıdır. Aramaların hangi sıra ile tekrar kurulacağını belirtmektedir. İlave olarak, arama önceliği, hangi aramaların bir bant genişliği rezervasyonu sırasında kabul edileceğini belirlemektedir.

**arama tesisi** Arama devam ederken, bir ISDN arama kurulum düzenleme başvurusunda kullanılır.

**Area - Alan** Fizikselden ziyade mantıksal olan, kendilerine bağlı cihazlarla birlikte (CLNS, DECnet veya OSPF tabanlı) segmentlerin bir bütünü. Alanlar yaygın olarak, tek bir otonom sistem oluşturmak için router kullanarak diğer alanlara bağlanırlar. *Ayrıca bakınız:* Otonom sistem

**ARM** Asynchronous Response Mode: Aktarımın, birincil ya da ikincil ünitelerden birinden başlatılabildiği, bir ana ve en az bir ilave istasyon kullanan HDLC iletişim modudur.

**ARP** Address Resolution Protocol: RFC 826 ile tanımlanan, IP adresinden MAC adresini bulan protokoldür. *Ayrıca bakınız: RARP.*

**AS** Autonomous system: Aynı routing metodolojisini paylaşan ortak yönetim altındaki ağ grubu. Otonom sistemler area'lar ile bölünmüştür ve IANA tarafından özel 16-bit bir numara atanmalıdır. *Ayrıca bakınız: Area*

**AS path prepending - AS yol eklemesi** Hatalı ASN'ler ekleyerek otonom sistem yolunu uzatmak için BGP'deki yol haritalarının kullanımıdır.

**ASBR** Autonomous System Boundary Router: RIP gibi ek bir routing protokolü ve OSPF çalıştıran, OSPF olmayan bir ağ ile bir OSPF otonom sistemi arasına yerleştirilen Area Border Router'ıdır. ASBR'lar, stub olmayan OSPF area'larına yerleştirilmelidir. *Ayrıca bakınız:* ABR, stub olmayan area ve OSPF.

**ASCII** American Standard Code for Information Interchange: Karakterleri simgelemek için 7 veri biti ile 1 eşitleme bitinden oluşan 8-bitlik kod.

**Asenkron aktarım** Genelde, her bir karakterin başını ve sonunu gösteren (başlatma ve durdurma bitleri olarak bilinen), kontrol bitlerindeki farklı frekans ve faz karakterleriyle, düzensiz gönderilen Dijital sinyallerdir. Isochronous aktarım ve synchronous aktarım ile zıttır.

**ASIC** Application-specific integrated circuits: Filtreleme kararları için 2.katman switch'lerde kullanılmaktadır. ASIC, MAC adresi filtreleme tablosuna bakar ve alınan paketi hedef donanım adresine göndermek için hangi portun kullanılacağına karar verir. Frame'in sadece bu tek segmente gönderilmesine izin verilir. Şayet donanım adresi bilinmiyorsa, frame tüm portlardan gönderilir.

**ASN.1** Abstract Syntax Notation One: Bilgisayar yapıları ve tanımlanan yöntemlerden bağımsız, veri tiplerini açıklamak için kullanılan bir OSI dilidir. ISO Uluslararsı Standardı 8824 ile açıklanmıştır.

**ASP** AppleTalk Session Protocol: İstekleri sıralamakla beraber, oturumları kurmak, devam ettirmek ve kapatmak için ATP'de çalışan bir protokol. *Ayrıca bakınız: ATP*

**AST** Automatic Spanning Tree: SRB ağlarında spanning tree'nin otomatik belirlenmesini destekleyen, ağdaki bir düğümden diğerine giden spanning explorer frame'leri için bir yol sağlayan fonksiyondur. AST, IEEE 802.1 standartına dayanmaktadır. *Ayrıca bakınız: IEEE 802.1 ve SRB.*

**atanmış hat** Bir bant genişliğini paylaşmayan point-to-point bağlantı.

**ATCP** AppleTalk Control Program: RFC 1378'de tanımlı, PPP üzerinde AppleTalk kurulumu ve yapılandırması için kullanılan protokol. *Ayrıca bakınız: PPP*

**ATDM** Asynchronous Time-Division Multiplexing: Bilgi göndermek için kullanılan bir tekniktir. Belirli vericilere, önceden belirlemekten ziyade ihtiyaç olduğunda zaman dilimi tanımlanarak, normal TDM'lerden farklılık gösterir. FDM, istatistiksel multiplexing ve TDM ile zıttır.

**ATG** Address Translation Gateway: Router'lara çoklu, bağımsız DECnet ağlarınına bir route belirlemek ve ağlar arasında seçili düğümler için kullanıcı-atanmış bir adres çevirisi oluşturmak için imkan veren Cisco DECnet routing yazılımındaki mekanizmadır.

**ATM** Asynchronous Transfer Mode: Sabit 53-byte hücrelerle tespit edilen, ses, video ya da veri gibi çoklu servis sistemlerindeki hücrelerin aktarılmasında kullanılan, uluslarası bir standarttır. Sabit boyutlu hücreler işlemlerin donanımda olmasına izin verdiğinden aktarma gecikmeleri azalmaktadır. ATM, SONET, E3 ve T3 gibi yüksek-hız aktarım ortamlarının avantajlarını en üst seviyeye çıkarmak için tasarlanmıştır.

**ATM ARP sunucusu** Adres çözümleme servisleriyle, ATM üzerinde klasik IP koşan mantıksal subnetleri sağlayan bir cihazdır.

**ATM Forumu** ATM teknolojisi için, standartlar-bazlı kurulum anlaşmalarını geliştirmek ve yükseltmek için 1991 de Northern Telekom, Sprint, Cisco Sistemleri ve NET/ADAPTIVE tarafından ortaklaşa kurulan uluslararası organizasyon. ATM Forumu, ANSI ve ITU-T tarafından geliştirilen resmi standartları geliştirir ve resmi standartlar yayınlanmadan önce kurulum anlaşmaları oluşturur.

**ATM katmanı** ATM ağında veri bağlantı katmanının servis bağımsız bir alt katmanıdır. Standart 53-byte ATM hücresi oluşturmak için ATM katmanı, AAL'dan 48-byte segmentler alır ve her birine 5-byte başlık ekler. Sonra bu hücreler, fiziksel ortamlar üzerinden aktarım için physical katmana gönderilir. *Ayrıca bakınız: AAL*

**ATM uçnoktası (endpoint)** Bir ATM ağında bağlantıyı başlatma ve sonlandırma. ATM uçnoktaları, sunucular, iş istasyonları, ATM-to-LAN switch'leri ve ATM router'larını içerirler.

**ATM user-user bağlantısı** ATMM prosesleri gibi en az iki ATM servis kullanıcısı arasında iletişim sağlamak için ATM katmanı tarafından kurulan bir bağlantı.

**ATM üzerinde klasik IP** RFC 1577 ile tanımlanan, ATM özelliklerini arttıran, ATM üzerinde IP koşması hususunda düzenleme.

**ATMM** ATM Management: ATM switch'lerde çalışan, hız uygulamalarını ve VCI çevirilerini yöneten bir prosedür. *Ayrıca bakınız: ATM*

**ATP** AppleTalk Transaction Protocol: Birinin, diğerinden verilen bir işi yerine getirmesini ve sonuçları rapor etmesini istediği, iki soket arasında güvenli işlemleri mümkün kılan bir transport-katmanı protokolüdür. ATP talep-cevap eşlerinin kayıpsız alışverişinden emin olmak için talepleri ve cevapları birbirine bağlar.

**Attenuation (zayıflama)** İletişimde, tipik olarak mesafenin sebep olduğu, sinyal enerjisinin zayıflığı veya kaybolması.

**AURP** AppleTalk Update-based Routing Protocol: Bir AppleTalk WAN oluşturmak için, farklı bir ağ boyunca (TCP/IP gibi) bitişik olmayan enaz iki AppleTalk ağ topluluğunun bağlantısına izin veren farklı protokolün başlığındaki AppleTalk trafiğini enkapsüle etmek için bir teknik. Yapılan bağlantı, bir AURP tüneli olarak belirtilir. Harici router'lar arasındaki routing bilgisinin değiş tokuş edilmesiye, AURP tüm AppleTalk WAN'ı için routing tablolarını muhafaza eder. *Ayrıca bakınız: AURP tüneli.*

**AURP tüneli** Bir TCP/IP ağı gibi farklı ağ tarafından fiziksel, bölünmüş AppleTalk ağ toplulukları arasında tek bir sanal link gibi davranan, AURP WAN'ında yapılan bir bağlantı.

**Authentication (kimlik doğrulama)** AAA modelindeki ilk bileşen. Kullanıcılara tipik olarak, kendilerini eşsiz tanımlamakta kullanılan bir kullanıcı adı ve şifresi üzerinden kimlik doğrulaması yapılır

**Authorization (yetkilendirme)** AAA modelinde, kimlik denetimi bilgisine dayanan bir kaynağa erişime izin verme eylemi.

**automatic call reconnect** Arızalı bir trunk hattından otomatik tekrar çağrı yönlendirmeyi mümkün kılan bir özellik.

**auxiliary port (yardımcı port)** Size, bir modeme bağlanma, router'ı arama ve konsol yapılandırma ayarları yapma izin veren, Cisco router'ların arkasındaki konsol portu.

**B kanalı** Bearer channel: Kulanıcı verisini aktaran, ISDN'deki 64Kbps, full-duplex bir kanal. *Mukayese ediniz: D kanalı, E kanalı ve H kanalı.*

**B8ZS** Binary 8-Zero Substitution: T1 ve E1 devreleri üzerinde, 8 ardışık sıfırın aktarıldığı herzaman özel bir kod ikamesi kullanan, bağlantının uzak ucunda çevrilen bir hat-kodu türü. Bu teknik, veri akışından bağımsız olarak birlerin yoğunluğunu garanti eder. Ayrıca, bipolar 8-sıfır ikamesi olarak da bilinir. *Tersidir: AMI. Ayrıca bakınız: birlerin yoğunluğu.*

**Back end (geri uç)** Bir front end'e (ileri uca) servisler sağlayan bir düğüm ya da yazılım programı. *Ayrıca bakınız. Sunucu.*

**Backbone (anaomurga)** Diğer ağlara gönderilen ve diğer ağlardan başlatılan trafik için öncelikli yol sağlayan ağın temel bölümü.

**bandwidth on demand (BoD)** Bu fonksiyon, belirli bir bağlantı için uygun bant genişliği miktarını artırnak için ilave bir B kanalının kullanılmasına izin verir.

**Bant genişliği** Ağ sinyalleri arasında kullanılan en yüksek ve en düşük frekanslar arasındaki aralık. Yaygın olarak, bir ağ protokolü veya ortamının, ölçülen throughput (yapılan iş) kapasitesine işaret eder.

**Bant-içi yönetim** Bant-içi yönetim, ağ içindeki bir ağ cihazının yönetimidir. Örnekler, lokal LAN üzerinden doğruca Simple Network Management Protocol (SNMP) ve Telnet kullanmayı içerir. *Mukayese ediniz: Bant-dışı yönetim.*

**Bant-içi sinyalleşme** Bant-içi sinyalleşme, analog POTS hatlarındaki arama bekleme gibi, sinyalleşmeyi taşımak için taşıyıcı kanal kullanımıdır. Bu, bir ISDN devredeki mevcut ikinci aktif bir aramayı kullanan D kanalının durumundaki gibi, bant-dışı sinyalleşmenin tersidir.

**Baseband** Sadece bir taşıyıcı frekansı kullanan, ağ teknolojisinin bir özelliği. Ayrıca, darbant olarak da bilinir. *Mukayese ediniz: Genişbant*.

**Baseline (anahat)** Baseline bilgisi, ağ ve rutin kullanım bilgisi hakkında geçmişle ilgili veri içerir. Bu bilgi, problemin çözümüne en kısa sürede katkıda bulunabilecek, ağa en son yapılan değişikliklerin belirlenmesi için kullanılabilir.

**Baud** Her sinyal öğesinin, 1 bit gösterildiğinde, saniyedeki bit'le eşanlamlıdır. Saniyede aktarılan ayrı sinyal öğelerinin sayısına eşit bir sinyalleşme hız birimidir.

**BDR** Backup designated router: Bu, arıza durumunda, designated router'ı yedeklemek için OSPF ağında kullanılmaktadır.

**Beacon** Kablo kopması gibi, ringdeki ciddi bir probleme işaret eden FDDI veya Token Ring frame'i. Beacon frame'i, arızalı olduğu düşünülen istasyonun adresini taşır. *Ayrıca bakınız: Failure domain.*

**BECN** Backward Explicit Congestion Notification: BECN, bir Frame Relay ağı tarafından ayarlanan, tıkanık bir yola yönlenmiş frame'lerden uzaklaşan frame'lerdeki bittir. BECN ile frame'leri alan bir DTE, gerekli akış kontrol ölçümlerini yüksek-seviye protokollerinden almak isteyebilir. *Mukayese ediniz: FECN.*

**BGP eşleri (peers)** Bakınız: BGP komşuları.

**BGP komşuları** Dinamik routing bilgisini değiş tokuş etmek için bir iletişim işlemi başlatan, BGP çalıştıran iki router. OSI referans modelinin 4.katmanında bir TCP portu kullanırlar. Özellikle, 179. TCP portu kullanılmaktadır. Ayrıca BGP eşleri olarak da bilinir.

**BGP konuşmacı (speaker)** Önek ve route'larını yayınlayan bir router.

**BGP tanımlayıcı** Bu alan, BGP konuşmacısını (speaker) belirleyen bir değer içerir. Bu, bir OPEN mesajı gönderildiğinde, BGP router tarafından seçilen rastgele bir değerdir.

**BGP4** BGP version 4: En yaygın olarak İnternette kullanılan domain'ler arası routing protokolünün 4. versiyonu. BGP4, CIDR'ı destekler ve routing tablosunun boyutunu azaltmak için route-sayma mekanizması kullanır. *Ayrıca bakınız: CIDR.*

**BIP** Bit Interleaved Parity: bir linkteki hataları görüntülemek için ATM'de kullanılan bir yöntem. Önceki blok veya frame için linkteki ek yükü tespitinde bir kontrol biti veya mesajı gönderir. Bu, aktarımdaki bit hatalarının bulunmasına ve bakım bilgisi olarak taşınmasına izin verir.

**BISDN** Broadband ISDN: Video gibi yüksek-bant genişliği teknolojilerini yönetmek için oluşturulan ITU-T standartıdır. BISDN, şimdilik, SONET-tabanlı aktarım devreleri boyunca ATM teknolojisi kullanır. Tipik olarak 155Mbps ve 622Mbps arasında, hatta günümüzde (çok paranız varsa) gigabyte seviyesinde veri oranı sağlar. *Ayrıca bakınız: BRI, ISDN ve PRI.*

**bidirectional shared tree (çiftyönlü paylaşım ağacı)** Shared tree multicast göndermenin bir yöntemi. Bu yöntem, ne kadar yakın olursa olsun, kaynak veya RP'den veri almak için grup üyelerine izin verir. *Ayrıca bakınız: RP (rendezvous point.)*

**Binary** Birleri ve sıfırları kullanan, iki-karakter bir numaralama yöntemi. İkili numaralama sistemi, bilgilerin tamamen dijital gösterimi temeline dayanır.

**Binding** Bir LAN'da belirli bir frame tipini kullanan Ağ katmanı protokolünü yapılandırmakdır.

Bir PVC'deki güvenli hızdır ve ağ tıkanıklığı olduğunda, iptal olması için trafik denetimi fonksiyonu tarafından etiketlenmez. Bu güvenli burst, byte'lar ve hücreler için atanmaktadır.

**Bir yoğunluğu** Pulse yoğunluğu olarak da bilinen bir sinyal saat denetimi yöntemidir. CSU/DSU, üzerinden geçen saat denetim bilgisine erişir. Bu düzenlemenin çalışması için verinin, aktarılan her bir 8 bit için en az bir binary 1 içermesi için şifrelenmesi gerekir.

**Birleşik metric** IGRP ve EIGRP gibi uzak bir ağa en iyi yolu bulmak için birden fazla metrik kullanan, routing protokolleridir. IGRP ve EIGRP'nin her ikiside, varsayılan, bir hattın bant genişliğini ve gecikmesini kullanır. Bununla beraber, maximum transmission unit (MTU), yükleme ve bir hattın güvenilirliği de kullanılabilir.

**Bit** Bir 1 veya 0 olan ikili sayı. Sekiz bit, bir byte yapar.

**Bit-tabanlı protokol** Frame'in içeriğine bakmaksızın, frame'leri aktaran data link katmanı iletişim protokolü sınıfı. Bit-tabanlı protokoller, byte-tabanlı olanlarla karşılaştırıldığında, daha verimli ve güvenli full-dupplex çalışma sağlarlar. *Mukayese ediniz: Byte-tabanlı protokol.*

**Blok boyutu** Bir subnet'te kullanılabilecek kullanıcıların sayısı. Blok boyutları, tipik olarak, 4, 8, 16, 32, 64 ve 128'in artımında kullanılabilir.

**Boot ROM** Routerı bootstrap (önyükleme) moduna çekmek için kullanılır. Daha sonra, bootstrap modu, cihazı bir işletim sistemiyle açar. ROM ayrıca küçük bir Cisco IOS tutabilir.

**Boot sequence (sırası)** Bir routerın nasıl boot edileceğini açıklar. Yapılandırma kayıt defteri (register) router'a, hem IOS'u nereden boot edeceğini hem de yapılandırmayı nasıl yükleyeceğini söyler.

**Bootstrap protokolü** İstemcilere, dinamik olarak IP adresi ve ağ geçidi atamak için kullanılan bir protokol.

**Border gateway** Farklı otonom sistemlerdeki router'larla iletişimi kolaylaştıran bir router.

**Border peer (sınır eşi)** Bir peer grubun yetkisindeki cihaz. Bir hiyerarşik tasarımda kenarda bulunur. Peer grubun herhangi bir üyesi, bir kaynağın yerini bulmak istediğinde, sınır eşine bir explorer gönderir. Bundan sonra sınır eşi, bu isteği istemci router adına gönderir, bu trafiğin artmasına engel olur.

**Border router** Tipik olarak, bir alanı backbone (ana omurga) alanına bağlayan bir router olarak Open Shortest Path First'te (OSPF) tanımlıdır. Bununla beraber, border router, bir firmayı İnternete bağlayan bir router'da olabilir. *Ayrıca bakınız: OSPF.*

**BPDU** Bridge Protocol Data Unit: Ağdaki bridge'ler arasında bilginin değiş tokuş edilmesi amacıyla tanımlanabilir aralıklarda gönderilen bir Spanning Tree Protokolü başlatma paketi.

**BRI** Basic Rate Interface: Video, veri ve ses arasında devre-anahtarlamalı iletişime yardım eden, ISDN arayüzü. İki B kanalı (her biri 64kbps) ve bir D kanalından (16Kbps) oluşmaktadır. *Mukayese ediniz: PRI. Ayrıca bakınız: BISDN.*

**Bridge belirleyici (identifier)** Bir 2.katman anahtar ağ topluluğunda, root bridge'i seçmek için kullanılır. Bridge ID'si, bridge önceliği ve temel MAC adresinin bir kombinasyonudur.

**Bridge** Bir ağdaki iki segmenti bağlamak ve aralarında paketleri aktarmak için bir cihaz. Her iki segment, iletişim kurabilmek için aynı protokolleri kullanmak zorundadır. Bridge'ler, OSI referans modelinin 2.katmanı, Veri Hattı katmanında çalışırlar. Bridge'in amacı, gelen bir frame'i, MAC adresine bağlı olarak filtrelemek, göndermek ve yaymaktır.

**Bridge grubu** Köprülemenin router yapılandırmasında kullanılır. Bridge grupları, eşsiz bir numara ile tanımlanmaktadır. Ağ trafiği, aynı bridge grubuna üye olan tüm arayüzler arasında köprülenir.

**Bridge önceliği** Bridge'in STP önceliğini ayarlar. Tüm bridge öncelikleri, varsayılan olarak, 32768'e ayarlanmıştır.

**Bridging kısır döngüsü (bridging loop)** Bir ağa birden fazla link olduğunda ve STP protokolünün açık olmadığında, bir bridge ağında kısır döngüler oluşur.

**Broadband (genişbant)** Telekomünikasyonda, genişbant, 4Khz (tipik ses seviyesi) den daha büyük bant genişliğine sahip bir kanal olarak sınıflandırılmaktadır. LAN terminolojisinde, analog sinyalleşme kullanılan bir koaksiyel kablo olarak sınıflandırılır. Wideband olarak da bilinir.

**Broadcast (multi-access) ağlar** Ethernet gibi Broadcast (multiaccess) ağlar, birçok cihazla, aynı ağa bağlanmak (ya da erişmek) için imkan verirler, ayrıca, bir paketi bir ağdaki tüm düğümlere taşıyan bir broadcast kabiliyeti sağlar.

**Broadcast domain** Gruptaki herhangi bir cihazdan başlatılan broadcast frame'lerini alan cihazların bir grubudur. Router'lar broadcast frame'lerini iletmediklerinden, broadcast domain'leri bir broadcast'ten diğerine iletilmezler.

**Broadcast fırtınası** Ağ segmenti üzerinde çok sayıda broadcastin eşzamanlı aktarımının neden olduğu, ağda istenmeyen bir durumdur. Böyle bir olay, ağ bant genişliğini etkileyebilir ve zaman aşımı ile neticelenebilir.

**Broadcast** Mantıksal ve donanımsal adreslemenin her ikisinde de kullanılır. Mantıksal adreslemede, kullanıcı adresleri, tamamen birlerden olacaktır. Donanımsal adreslemeyle, donanım adresi, ikili sistemde tamamen birlerden olacaktır (hexadecimal'de tamamen F'lerden).

**Buffer Arabellek** Aktarım esnasında verinin tutulması için atanmış bir depolama alanıdır. Arabellekler, işlem hızındaki değişimleri dengelemek, genelde daha hızlı cihazlardan alınan, düzensiz veri yoğunluklarının alınması/saklanması için kullanılmaktadır. Gönderilen veriden önceki her şey alınana kadar gelen bilgi saklanmaktadır. Ayrıca bir bilgi arabelleği olarak da bilinir.

**Bursting** ATM ve Frame Relay içeren bazı teknolojiler, burstable olmayı göze alırlar. Bunun anlamı şudur: kullanıcı verisi, bağlantı için normalde ayrılan bant genişliğini aşabilir, bununla beraber, port hızı limitinin üzerine çıkamaz. Buna örnek, bir T1'deki 128Kbps Frame Relay CIR olabilir. Üreticiye bağlı olarak, kısa bir sure için 128Kbps'dan daha hızlı göndermesi mümkün olabilir.

**Bus** Bir dijital sinyalin, bilgisayarın bir parçasından diğerine veri göndermek için kullanılabildiği, tipik olarak tel veya bakır kablo olan herhangi bir yaygın fiziksel yoldur.

**BUS** Broadcast and unknown servers: LAN emülasyonunda, bilinmeyen (register olmamış) adresli tüm broadcast ve paketleri, ATM için gerekli noktadan-noktaya sanal devrelere çözümlemekten sorumlu donanım veya yazılımdır. *Ayrıca bakınız: LANE, LEC, LECS ve LES.*

**Bus topoloji** Ağdaki çeşitli istasyonlardan aktarımların ortam uzunluğunca tekrarlandığı ve diğer tüm istasyonlarca kabul edildiği, doğrusal bir LAN mimarisidir. *Mukayese ediniz: Ring topoloji ve star topoloji.*

**BX.25** X.25'in AT&T kullanımı. *Ayrıca bakınız: X.25*

**Bypass gecikmesi** Token ringdeki belirli bir arayüzün kapatılmasına ve fiilen ringden çıkılmasına imkan veren bir cihaz.

**Bypass modu** Bir arayüzü silen, FDDI ve Token Ring ağ işlemi.

**Byte** Sekiz bit. Ayrıca bakınız: Octet.

**Byte-tabanlı protokol** Frame'lerin sınırlarını işaretlemek için kullanıcı karakter setinden özel bir karakter kullanan, data-link iletişim protokolünün bir türü. Bit-tabanlı protokoller bu protokollerin yerini almaktadır. *Mukayese ediniz: Bit-tabanlı protocol.*

**CAC** Connection Admission Control: Bir bağlantı isteğinin, mevcut bağlantılardaki QoS garantilerini bozup bozmadığını anlamak için bağlantı kurulumu gerçekleştirilirken, her ATM switch'i tarafından çalıştırılan eylemlerin sırasıdır. Aynı zamanda CAC, bir ATM ağı boyunca bir bağlantı isteğine bir route oluşturmak için kullanılmaktadir.

**CBR** Constant bit rate: ATM ağlarında kullanmak için tasarlanan QoS sınıfı bir ATM Forumu. CBR, güvenli teslimatı garanti etmek için hassas saat denetimine dayalı bağlantılar için kullanılmaktadır. *CBR ve PCR.*

**CD** Carrier detect: Bir arayüzün aktif olduğunu ya da modemle oluşturulan bir bağlantı kurulduğunu işaret eden bir sinyal.

**CDP** Cisco Discoveri Protokol: Donanım tipi, yazılım versiyonu ve aktif arayüzler hakkında komşu bir Cisco cihaza bilgi veren, Cisco'nun geliştirdiği protokoldür. Cihazlar arasında bir SNAP frame'i kullanır ve yönlendirilemez.

**CDP holdtime** Komşu tarafından güncellenmeyen bir bilginin atılmasından önce, komşu bir router tarafından alınan Cisco Discoveri Protokol bilgisinin bir router tarafından tutulacağı süre. Bu sayaç, varsayılan olarak 180 saniyeye ayarlanmıştır.

**CDP sayacı** Tüm router arayüzlerinden iletilen Cisco Discoveri Protokol yayınları arasındaki varsayılan zaman miktarı. CP sayacı, varsayılan olarak 90 saniyedir.

**CDT** Cell Transfer Delay: ATM'de verilen bir bağlantı için kaynak kullanıcı- ağ arayüzündeki bir hücre çıkış olayı ile hedefteki ilgili hücre giriş olayı arasındaki zaman periyodu. Bu noktalar arasındaki CDT, toplam ATM'ler arası aktarım gecikmesi ve toplam ATM işleyiş gecikmesinin ortalamasıdır.

**CDTV** Cell Delay Variation Tolerance: Bir bağlantı kurulduğunda, belirtilen ATM ağlarındaki trafik yönetimi için bir QoS parametresidir. CDTV tarafından tanımlanan CBR aktarımlarındaki PCR tarafından alınan veri örnekleri için kabul edilebilen dalgalanma seviyeleridir. *Ayrıca bakınız:*

**cell payload scrambling** Bir ATM switch'in, bazı orta-hız kenar ve trunk arayüzlerinde (T3 ve E3 devreleri) frame'lere ayırma yöntemi. Cell payload scrambling, belirli yaygın bit örneğiyle hat senkronizasyonunu sağlamak için bir hücrenin veri parçasını tekrar düzenler.

**Centrex** Bir on-site PBX'i andıran, lokal anahtarlama sağlayan bir lokal santral taşıyıcı servisidir. Centrex, on-site anahtarlama yeteneğine sahip değildir. Bu nedenle, tüm müşteri bağlantıları merkez ofise (CO) geri döner. *Ayrıca bakınız: CO.*

**CER** Cell error ratio: ATM'de, belirli bir yayılma süresinde, hatalara sahip iletilmiş hücrelerin, toplam iletilmiş hücrelere oranıdır.

**CGMP** Cisco Grup Yönetim Protokolü: Cisco tarafından geliştirilen tescilli bir protokoldür. Router, Catalyst switch'lere multicast üyelik komutları göndermek için CGMP kullanır.

**Channalized E1** 2.048Mbps'da çalışan, DDR, Frame Relay ve X.25'ı destekleyen, 29 B kanalı ve bir D kanalına bölünmüş bir giriş hattıdır. *Mukayese ediniz: Channelized T1.*

**channelized T1** 1.544Mbps'da çalışan, DDR, Frame Rellay ve X.25'i destekleyen, tekli veya grup kanallarının çeşitli hedeflere bağlandığı, 23 B kanalı ve 64Kbps tek bir D kanalına bölünmüş bir giriş hattıdır. *Mukayese ediniz: Channelized E1.*

**CHAP** Challenge Handshake Authentication Protocol: PPP enkapsülasyon kullanan hatlarda desteklenmektedir. Yetkili olmayan kullanıcıların girişini engellemeye yardım eden, uzak uçları tespit eden bir güvenlik özelliğidir. CHAP uygulandıktan sonra, router ya da access sunucusu, belirtilen kullanıcının girişine izin verilip verilmeyeceğine karar verir. Yeni, PAP'tan daha güvenli bir protokoldür. *Mukayese ediniz: PAP.*

**Checksum** Gönderilen verinin bütünlüğünden emin olmak için yapılan bir test. Matematik fonksiyonlarının silsilesiyle alınan bir değer serisinden hesap edilen bir numaradır. Tipik olarak, hesaplanan verinin sonuna yerleştirilir ve sonra doğrulama için alınan uçta tekrar hesaplanır. *Mukayese ediniz: CRC*

**Choke paketi** Tıkanıklık olduğunda, bir vericiye, gönderme hızını düşürmesi gerektiği bilgisini gönderen bir pakettir.

**CIDR** Classless Inter-Domain Routing: Bir IP ağ grubunun, diğer ağlara, birleşik, daha geniş olarak görünmesine izin verir. CIDR'da, IP adresleri ve subnet mask'lar, noktalarla ayrılmış 4 oktet ve sonuna maskelenen bitleri temsil eden rakamın eklendiği bir adres olarak yazılmaktadır (bir subneti temsil eden kısaltma şekli). *Ayrıca bakınız: BGP4.*

**CIP** Channel Interface Processor: Bir kullanıcı ana bilgisayarını, bir kontrol ünitesine bağlayan, Cisco 7000 serisi router'larda kullanmak için bir kanal ilave arayüzü. Bu cihaz, kanal eklemek için bir FBP ihtiyacını ortadan kaldırır.

**CIR** Committed information rate: Minimum yayılma zamanı ortalamasıdır ve bps ile ölçülür. Bir Frame Relay ağının üzerinde mutabık kalınan minimum bilgi aktarım değeridir.

**Cisco FRAD** Cisco Frame Relay Access Device: Mevcut bir LAN gerekmeksizin, SDLC cihazlarını Frame Relay'e bağlamak için Cisco IPS Frame Relay SNA servislerini destekleyen bir Cisco ürünüdür. Tamamıyla çalışır çokluport router'a yükseltilebilir. SDLC'den Ethernet ve Token Ring'e dönüştürme aktif edilebilir, fakat eklenen LAN'ları desteklemez. *Ayrıca bakınız: FRAD.*

**Cisco IOS** Cisco İnternet İşletim Sistemi yazılımı. CiscoFusion mimarisi altındaki tüm ürünler için ortak işlevsellik, ölçeklenebilirlik ve güvenlik sağlayan, router ve switch'lerin, Cisco satır çekirdeği. *Ayrıca bakınız: CiscoFusion.*

**CiscoFusion** Cisco'nun, Cisco IOS çalışan ağlararası iletişim mimarisinin adıdır. Sahip olunan router ve switch'lerin farklı yığınlarının kapasitelerini birleştirmek için dizayn edilmiştir.

**CiscoView** Dinamik durum istatistikleri ve yapılandırma bilgilerinin kapsamlarını mümkün kılan Cisco ağ cihazları için GUI-tabanlı yönetim yazılımı. Cisco cihazın şasisinin fiziksel görünümünü görüntüler ve cihaz-görüntüleme fonksiyonları ile önemli hata giderme özellikleri sağlar. Çok sayıda SNMP-tabanlı ağ yönetim platformu ile birleştirilebilir.

**Classful routing** Bir route güncellemesi gönderildiğinde, subnet mask'a bilgisi gönderilmeyen routing protokolü.

**Classless routing** Routing güncellemelerinde subnet mask bilgileri gönderen routing. Değişen-Uzunlukta Subnet Maskelemesi (VLSM) ve suppernetting, classless routing'e imkan verir. Classless routing'i destekleyen routing protokolleri, RIP ver2, EIGRP ve OSPF'tir.

**CLI** Command-line interface: Maksimum esneklikle Cisco router ve switch'leri yapılandırmanıza izin verir.

**CLP** Cell Loss Priority: Ağ tıkanıklığı esnasında iptal olan bir hücre olasılığını tanımlayan ATM hücre başlığındaki alan. CLP=0'lı hücreler güvenli kabul edilir ve iptal olma ihtimali yoktur. Tıkanık bölümler boyunca iptal olabilen CLP=1'li hücreler, en iyi çalışan trafik olarak kabul edilirler ve güvenli trafiği kullanmak için daha çok kaynak taşırlar.

**CLR** Cell Loss Ratio: ATM'de, iptal olmuş hücrelerin, başarıyla taşınan hücrelere oranıdır. Bir bağlantı kurulduğunda CLR, bir QoS parametresine atanabilir.

**CO** Central office: Belirli bir alandaki tüm döngülerin bağlandığı ve abone hatlarının devre anahtarlı olduğu lokal telefon firma ofisi.

**collapsed omurga** Bir ağlararası iletişim cihazı ile tüm ağ segmentlerinin birbirine bağlandığı, dağıtık olmayan bir omurga. Collapsed omurga, bir router, hub veya switch gibi bir cihazda çalışan sanal bir ağ segmenti olabilir

**collision domain** Çarpışan frame'lerin tespit edildiği ethernet'teki ağ alanı. Collision'lar, hub ve repeater'lar ile yayınlanır, fakat LAN switch ve router'larla yayınlanmazlar. *Ayrıca bakınız: Collision.*

**Collision** Ethernet'te aynı anda aktarım gönderen iki düğümün etkisi. Fiziksel ortamda karşılaştıklarında, her bir düğümden frame'ler çarpışacak ve hasar görecektir. *Ayrıca bakınız*: *Collision domain.*

**Configuration register - yapılandırma kayıt defteri** İlk kullanıma hazırlama sırasında, Cisco router'ların nasıl çalışacağına karar veren, donanım veya yazılımda saklanan, yapılandırılabilir 16-bit bir değer. Donanımda, bit konumu, bir jumper kullanarak ayarlanır. Yazılımda, başlangıç seçeneklerini ayarlamak için kullanılan özel bit örneklerininin belirtilmesiyle ayarlanır ve yapılandırma komutları ile hexadecimal bir değer kullanarak yapılandırılır.

**Congestion (tıkanıklık)** Ağın kaldırabileceğinden daha fazla trafik olması.

**congestion avoidance** Gecikmeleri minimize etmek için, sisteme girildiğinde, trafiği kontrol etmek için bir ağ kullanım yöntemi. Göstergeler taşınamaz olarak işaret ettiğinde, düşük-öncelikli trafik, ağın sonunda atılır, böylece kaynaklar verimli kullanılır.

**congestion çökmesi** Hedef noktalarına az ya da hiçbir trafiğin ulaşmadığı ATM ağlarında paketlerin tekrar aktarılması sonucu oluşan durum. Genelde, zayıf paket atılımı veya ABR tıkanıklık geribesleme mekanizmasıyla birleştirilmiş etkisiz ve yetersiz önbelleğe alma kapasiteli switch'lerle kurulan ağlarda olmaktadır.

**connection ID** Bir router'a yapılan her bir Telnet oturumuna verilen kimliklerdir. "show sessions" size, lokal bir router'ın uzak bir router'a yaptığı bağlantıları verecektir. "show users" komutu, lokal router'ınıza telnet yapan kullanıcıların ID'lerini gösterecektir.

**Connectionless (bağlantısız)** Bir sanal devre yaratmaksızın olan veri transferidir. Düşük ek-yüke sahiptir, en güçlü taşımayı kullanır ve güvenli değildir. Connection-oriented ile zıttır. *Ayrıca bakınız: Sanal devre*

**Connectionless Network Service (CLNS)** Bakınız: *Connectionless.*

**connection-oriented (bağlantı-tabanlı)** Herhangi bir veri transfer edilmeden önce sanal bir devre oluşturan, veri transfer yöntemi. Güvenli veri transferi için onay ve akış kontrolü kullanır. Connectionless ile zıttır. *Ayrıca bakınız: Sanal devre.*

**control direct VCC** Faz 1 LAN emülasyonuyla tanımlanan iki kontrol bağlantısının biridir. Bir LEC ile LES'e, ATM'de kurulan iki yönlü bir sanal kontrol bağlantısıdır (VCC). *Ayrıca bakınız: Control distribute VCC.*

**control distribute VCC**Faz 1 LAN emülasyonuyla tanımlanan iki kontrol bağlantısının biridir. Bir LEC ile LES'e, ATM'de kurulan tek yönlü bir sanal kontrol bağlantısıdır (VCC). Genel olarak, VCC, point-to-multipoint bağlantısıdır. *Ayrıca bakınız: Control direct VCC.*

**Convergence** En uygun yolları kullanan ağın uygun bir görünümünü oluşturmak ve routing tablolarını güncellemek için bir ağtopluluğundaki tüm router'lar için gerekli işleyiş. Bir STP convergence sırasında, kullanıcı verileri iletilmez.

**core katmanı** Cisco hiyerarşik ağlarını tasarlamak, kurmak ve devamlılığını sağlamak için size yardım eden, Cisco üç-katmanlı hiyerarşik modelinin en üstteki katmanıdır. Bu katmanda, paket filtrelemesi olmamalıdır.

**Cost** Ayrıca "path cost" olarak da bilinen, tipik olarak bir ağ yöneticisi tarafından atanan ve bir ağlar topluluğundaki farklı route'ları mukayese etmek için routing protokolü tarafından kullanılan, hop count, bant genişliği ve diğer hesaplamalara dayalı isteğe bağlı bir değerdir.

**count to infinity** Router'ların, belirli ağlara hop sayılarının artmaya devam ettiği, convergence'in yavaş olduğu routing algoritmalarında olan bir problemdir. Bu problemden kaçınmak için, farklı routing protokolleri için çeşitli çözümler geliştirilmiştir. Bu çözümlerin bazıları, maksimum hop sayısı (sonsuzluk tanımı) tanımlamak, route poising, poisin reverse ve split horizon'dur.

**CPCS** Common Part Convergence Sublayer: Servis bağımsız iki AAL altkatmanından biri. CS ve SAR altkatmanlarına bölünmüştür. CPCS, ATM ağları boyunca aktarım için veri hazırlar. ATM katmanına gönderilen 48-byte veri hücreleri oluşturur. *Ayrıca bakınız: AAL ve ATM katmanı.*

**CPE** Customer premises equipment: Müşteri lokasyonlarında kurulan ve servis sağlayıcı ağına bağlanan telefon, modem ve terminal gibi öğelerdir.

**Crankback** ATM'de, seçilen bir yolda herhangi bir yerdeki düğüm, bir bağlantı kurulum isteğini kabul etmediğinde, isteği blokladığında kullanılan bir düzeltme tekniği. Yol, aradaki bir düğüme geri döndürülür, sonra son hedefe alternatif bir yol bulmaya çalışmak için GCAC kullanır.

**CRC** Cyclic redundancy check: Frame alıcısının, frame içeriklerini asal bir binary bölen ile bölüp bir hesaplama yaparak ve kalanı, gönderilen düğüm ile frame'de saklanan bir değeri kıyaslayarak hataları tespit eden bir metodolijidir. *Zıt anlamlıdır: Checksum*

**Crossover kablo** Bir switch'i switch'e, kullanıcıyı-kullanıcıya, hub'ı hub'a veya switch'i hub'a bağlayan Ethernet kablo çeşididir.

**CSMA/CD** Carrier Sense Multiple Access with Collision Detection: Ethernet IEEE 802.3 komitesi tarafından tanımlanan bir teknoloji. Aktarımdan önce, her bir cihaz, dijital bir sinyal için kablo algılar. Aynı zamanda, CSMA/CD, ağdaki tüm cihazların, her seferinde biri olmak suretiyle, aynı kabloyu paylaşmasına izin verir. Şayet iki cihaz aynı anda aktarırsa, bir frame çarpışması olur ve bir çarpışma örneği gönderilir; cihazlar aktarımı durduracak, önceden belirlenen, ayrıca seçilen rastgele bir süre beklenecek ve daha sonra tekrar aktarmaya çalışacaktır.

**CSU** Channel service unit: Son kullanıcı ekipmanlarını lokal dijital telefon döngülerine bağlayan dijital bir mekanizma. Genelde, CSU/DSU gibi veri servis ünitesi ile beraber konuşulur. *Ayrıca bakınız: DSU*

**CSU/DSU** Channel service unit/data service unit: WAN ağlarında, CPE dijital sinyallerini servis sağlayıcının switch'inin anlayacağı dile çevirmek için kullanılan Physical katman cihazıdır. Sınır noktası olarak da bilinen bir CSU/DSU, tipik olarak bir RJ-45 (8-pin modüler) fişine takılı bir cihazdır.

**cut-through frame switching** Bir switch'ten verinin aktığı, paketin geldiği porta girişi tamamlanmadan önce ilk gelen parçanın switch'in çıkış portundan çıktığı bir frame-switching tekniğidir. Frame'in hedef adresi kesinleştirilip çıktı portu belirlenir belirlenmez frame'ler, cut-through switching kullanan cihazlar tarafından okunup, işlemden geçip yönlendirilecektir.

**D kanalı**(1) Data kanalı: Bir full-duplex, 16Kbps (BRI) veya 64Kbps (PRI) ISDN kanalı. Mukayese ediniz: B kanalı, E kanalı ve H kanalı. (2) SNA de, herhangi bir çevrebirimi ile işlemci ve ana bellek arasında bir bağlantı sağlayan kanal.

**Dahili EIGRP route** Bunlar, aynı otonom sistemlerinin üyesi EIGRP router'ları tarafından özel bir otonom sisteminde oluşturulan route'lardır.

**Data Circuit-terminating Ekipmanları (DCE)** DCE, DTE ekipmanlarına saat denetimi sağlamak için kullanılmaktadır.

**data direct VCC** ATM deki iki LEC ve Faz 1 LAN emülasyonu tarafından tanımlı üç veri bağlantısından biri arasındaki çiftönlü point-to-point bir sanal kontrol bağlantısıdır (VCC). Data direct VCC'ler QoS'u garanti etmediğinden, genellikle, UBR ve ABR bağlantıları için reserve edilirler. *Mukayese ediniz: control distribute VCC* ve *control direct VCC.*

**Data Link Control Katmanı** SNA mimari modelinin 2. katmanı, kullanılan bir fiziksel hat üzerindeki aktarımından ve bir bakıma OSI modelinin Data Link katmanı ile mukayese yapmaktan sorumludur.

**Data Link Katmanı** OSI referans modelinin 2. katmanı, fiziksel bir hat üzerinde güvenli veri aktarımından emin olur ve öncelikle, fiziksel adresleme, hat disiplini, ağ topolojisi, hata uyarısı, istenilen frame'lerin taşınması ve akış kontrolu sağlar. IEEE, bu katmanı, MAC altkatmanı ve LLC altkatmanı olarak daha çok sayıda bölümlemiştir. Aynı zamanda Link Katmanı olarak da bilinir. SNA modelinin Data Link Control katmanı ile mukayese edilebilir. *Ayrıca bakınız: Application katmanı, LLC, MAC, Network katmanı, Physical katman, Presentations katmanı, Session katmanı ve Transport katmanı.*

**Data Terminal Ekipmanı** (data terminal equipment) Bakınız: DTE

**Datagram** Önceden kurulu bir sanal devre olmaksızın bir ortam üzerinde bir Network katmanı birimi olarak aktarılan bilginin mantıksal bir yığını. IP datagramları, İnternetin öncelikli bilgi birimleridir. OSI referans modelinin çeşitli katmanlarında, cell, frame, mesaj, paket ve segment terimleri, aynı zamanda bu mantıksal bilgi gruplarını tanımlar.

**DCC** Data Country Code: ATM Forumu tarafından geliştirilen, özel ağlar tarafından kullanılmak için tasarlanmış iki ATM adres formatından biri. *Mukayese ediniz: ICD*

**DCE** Data communications equipment (EIA tarafından tanımlandığı gibi) veya data circuit-terminating equipment (ITU-T tarafından tanımlandığı gibi): Modemler gibi kullanıcı-ağ arayüzünün ağ bölümünü oluşturan bir iletişim ağ mekanizması veya bağlantısıdır. DCE, ağlara fiziksel bağlantı sağlar, trafiği iletir ve DTE ve DCE cihazları arasındaki veri aktarımını senkronize etmek için bir saat denetimi sinyali sağlar. *Mukayese ediniz: DTE*

**DDP** Datagram Delivery Protocol: Bir internetwork boyunca gönderilen datagramlardan sorumlu, bir connectionless protokol gibi AppleTalk ailesinde kullanılan protokol.

**DDR** Dial-on-demand routing: Bir router'a, gönderen istasyonun gereksinimlerine göre bir devre-anahtarlamalı oturumu otomatik başlatma ve sonlandırma izni veren bir teknik. Keepalive'leri taklit ederek router oturumu aktif gibi davranarak son kullanıcıyı kandırır. DDR, bir modem veya harici ISDN terminal adaptörü yardımıyla ISDN veya telefon hattı üzerinde routing'e izin verir.

**DE** Discard Eligibility: Switch çok yoğunken bir frame'in tercihli olarak atılabileceğini bir switch'e söylemek için Frame Relay ağlarında kullanılır. DE, committed information rate (CIR) gerekenden büyükse veya sıfıra ayarlandıysa, aktarım router'ları tarafından açılan, frame'deki bir alandır.

**deencapsulation** Bir katmanın, alt katmandan gelen Protocol Data Unit'deki (PDU) başlık bilgisini sildiği, katmanlaşmış protokoller tarafından kullanılan teknik. *Bakınız: Encapsulation.*

**default route** Routing tablosunda bir sonraki hop'un belirtilmediği frame'leri yöneltmek için kullanılan static routing tablo girişi.

**Değişken route'lar** Hata olduğunda yedek route'lar (static route'lar) oluşturmak için dinamik routing'de kullanılır.

**Demarc** Customer premises equipment (CPE) ile Telco'nun taşıma ekipmanı arasındaki sınır noktası.

**Demodülasyon** Bir module olmuş sinyali orjinal şekline getiren adımlar dizisi. Kullanıldığında, bir modem analog bir sinyali onun orjinal dijital şekline demodüle eder (ve, tersine, dijital veriyi bir analog sinyale module eder). *Bakınız: modülasyon.*

**Demultiplexing** Birden fazla giriş akımından oluşan multiplex sinyali, ayrı çıktı akımlarına çevirme prosesidir. *Ayrıca bakınız: Multiplexing.*

**designated bridge** Root bridge'e bir segmentten bir frame routing işlemindeki, en düşük root yol değerine sahip bridge.

**designated port** Routing portları atamak için Spanning Tree Protocol (STP) ile kullanılır. Şayet aynı ağa iki hat varsa STP, ağ döngüsünü durdurmak için bir portu kapatacaktır.

**designated router (DR)** Bir multi-access ağ için LSa'ler oluşturan ve OSPF operasyonundaki diğer özel hizmetleri çalıştırmak için gerekli bir OSPF router'ıdır. Bağlı iki router'ın minimumunu sağlayan Multi-access OSPF ağları, OSPF Hello Protokolü tarafından seçilen bir router'ı tespit eder, bu bir multi-access ağda gerekli komşulukların sayısında bir düşüşe olanak verir. Bu routing protokol trafiği yoğunluğunu ve veritabanının fiziksel boyutunu düşürür.

**Desktop katmanı:** Bazen desktop katmanı olarak belirtilen access katmanı. Access katmanı, kullanıcı ve çalışma gruplarının ağlar topluluğu kaynaklarına girişlerini kontrol eder.

**Devre anahtarlaması** PPP ve ISDN gibi dial-up ağlarla kullanılmaktadır. Veriyi geçirir, fakat ilk olarak, telefon araması yapar gibi, bir bağlantı kurulması gerekir.

**DHCP** Dynamic Host Configuration Protocol: DHCP, BootP protokolünün bir süpersetidir. Bunun anlamı, BootP gibi aynı protokol yapısını kullanır, fakat arttırılmış özelliklere sahiptir. Bu protokollerin her ikisi de, istendiğinde kullanıcıları dinamik yapılandıran sunucular kullanırlar. İki temel; adres havuzları ve kiralama süreleridir.

**Dial-Up Back-up** Dial-up arama bağlantıları, tipik olarak Frame Relay bağlantılarına yedekleme sağlamak için kullanılmaktadır. Yedekleme hattı, bir analog modem veya ISDN üzerinde aktif edilir.

**Dinamik girişler** Dinamik olarak, bir donanım veya mantıksal adres tablosu oluşturmak için 2. ve 3. katman cihazlarda kullanılır.

**Dinamik routing** "Adaptive routing" olarak da bilinen bu teknik, trafik ve fiziksel ağ değişikliklerini otomatik olarak uyarlar.

**dinamik VLAN** Bir yönetici, özel bir sunucuda, internetwork'teki tüm cihazların donanım adresleriyle girişler oluşturur. Sunucu daha sonra, yeni cihazları donanım adresi-tabanlı isteyen bir switch'e, ilgili VLAN'leri bildirir.

**directed broadcast** Uzak bir ağ segmentindeki özel bir düğüm grubuna aktarılan bir veri frame'i ya da paketi. Directed broadcastler, bütün kullanıcı bitleri açık olan bir hedef subnet adresi olan broadcast adresleri ile bilinirler.

**discovery mod** Dinamik yapılandırma olarak bilinen bu teknik, eklenmiş bir ağ hakkında çalışan bir düğümden bilgi elde etmek için bir AppleTalk arayüzü tarafından kullanılmaktadır.

**distance-vector protokolleri** Distance-vector protokolü, uzaklığı sorgulayarak uzak bir ağa en iyi yolu bulur. Bir paketin bir router boyunca her gidişi, bir hop olarak bilinir. Hedef ağa en düşük sayıda hop'la gidilen yol, en iyi yol olarak belirtilir. Bununla beraber, Cisco'nun IGRP'si, distance vector'ü dikkate alır ve uzak bir ağa en iyi yola karar vermek için hattın bant genişliği ve gecikmesinin birleşimi bir metrik kullanır.

**distance-vector routing algoritması** En kısa yolu bulmak için, routing algoritmalarının bu türü, her bir routerın tam güncellemesiyle tüm routing tablosunu göndermesi için gerekli, verilen bir yoldaki hop'ların sayısını, sadece kendi komşularına bildirir. Bu çeşit routing algoritmaları, kısır döngüler üretmek eğilimindedir, fakat, genelde link-state örneklerinden daha basittirler*. Ayrıca bakınız: Link-state routing algorithm ve SPF.*

**distribution katmanı** Cisco hiyerarşik ağlarını tasarımı, kurulumu ve devamlılığı için size yardım eden Cisco 3-katman hiyerarşik modelinin orta katmanı. Distribution katmanı, access katman cihazlarının bağlandığı noktadır. Routing bu katmanda çalışmaktadır.

**DLCI** Data Link Connection Identifier: Bir Frame Relay ağındaki sanal devreleri tespit etmek için kullanılmaktadır.

**DLSw** Data Link Switching: IBM, router-tabanlı ağlarda SNA(Systems Network Architecture) ve NETBIOS protokollerine destek sağlamak için 1992'de Data Link Switching'i (DLSw) geliştirdi. SNA ve NetBIOS protokolleri, herhangi bir 3.katman ağ bilgisi içermeyen yönlendirilemez protokollerdir. DLSw, yönlendirilebilmesi ve Remote Source-Route Bridging (RSRB) için bir alternatif olması için bu protokolleri TCP/IP mesajlarına enkapsüle eder.

**DLSw+** Cisco'nun DLSw kurulumu. RFC standartları için desteğe ilave olarak, Cisco, ölçeklenebilirliği arttırmak ve performans ve kullanılabilirliği geliştirmek için tasarlanan ilaveler eklemiştir.

**DNS** Domain Name System: Host isimlerini IP adreslerine çözmek için kullanılır.

**Döngüden kaçınma** Şayet, switch'ler arası çoklu bağlantılar, yedekleme amacıyla oluşturulursa, Spanning Tree Protokolü (STP), hala yedekleme izni verilirken ağ kısır döngülerini durdurmak için kullanılır.

**DSAP** Destination Service Access Point: Bir paketin hedef alanında belirtilen, bir ağ düğümünün servis giriş noktasıdır.

**DSR** Data Set Ready: Bir DCE açıldığı ve çalışmaya hazır olduğunda, bu EIA/TIA-232 arayüz devresi de bağlıdır.

**DSU** Data service unit: Bu cihaz, bir data terminal equipment (DTE) mekanizmasındaki fiziksel arayüzü T1 veya E1 gibi bir aktarım hizmetine uyarlamak için kullanılmaktadır ve aynı zamanda sinyal zamanlamasından sorumludur. Genellikle, channel service unit (CSU) ile gruplandırılır ve CSU/DSU olarak belirtilir*. Ayrıca bakınız: CSU.*

**DTE** Data terminal equipment: Bir hedef, kaynak ya da her ikisi olarak hizmet veren bir kullanıcı-ağ arayüzünün sonundaki kullanıcıda bulunan bir cihaz. DTE, multiplexer, router, protokol çevirici ve bilgisayar gibi bir cihaz olabilir. Bir veri ağına bağlantı, modem gibi bir data communication equipment (DCE) ile yapılmaktadır, bu cihaz tarafından üretilen saat denetleme sinyali kullanır. *Ayrıca bakınız: DCE.*

**DTR** Data Terminal Ready: DCE ye bağlanan, aktif bir EIA/TIA-232 devresinin DTE'ye bir veri iletmeye veya almaya hazırlık durumu.

**DUAL** Diffusing Update Algorithm: Enhanced IGRP'de kullanılan bu convergence algoritması, bütün bir route hesaplaması boyunca kısır döngü oluşmadan çalışılmasını sağlar. DUAL değişiklikten etkilenmeyen router'ları içermeden, topoloji değişikliğinin kapsayan router'lara aynı anda senkronize olmaları olanağı verir. *Ayrıca bakınız: Enhanced IGRP.*

**Durum geçişleri** Bit hücresinin ortasındaki dijital sinyalin durumunu okuyan dijital sinyalleşme düzenlemesi. Şayet o beş volt ise hücre bir olarak okunur. Şayet dijital sinyalin durumu sıfır volt ise bit, sıfır olarak okunur.

**Düğüm adresi** Bir ağ topluluğunda belirli bir cihazı tanımlamak için kullanılır. Network interface kartına yazılan bir donanım adresi veya bir yönetici ya da sunucu tarafından bir düğüme atanan mantıksal ağ adresi olabilir.

**DVMRP** Distance Vector Multicast Routing Protocol: Temel olarak Routing Information Protocol (RIP) tabanlı bu İnternet gateway protokolü, genel condensed-mode IP multicast planı geliştirir, komşuları arasında routing datagramlarını transfer etmek için IGMP kullanır. *Ayrıca bakınız: IGMP.*

**DXI** Data Exchange Interface: DXI, paket enkapsülasyonunu için DSU kullanarak bir ATM ağına, bir FEP gibi davranması için bir router, bridge veya hub gibi bir ağ cihazının etkisini tanımlar.

**E kanalı** Echo channel: Circuit-switching için kullanılan bir 64Kbps ISDN kontrol kanalı. Bu kanal için özel düzenleme, 1984 ITU-T ISDN teknik düzenlemesinden bulunabilir, fakat 1988 versiyonundan kaldırılmıştır. Ayrıca bakınız: *B kanalı, D kanalı* ve *H kanalı.*

**E.164** (1) Standart telefon numaralandırma sisteminden türetilen, özellikle ISDN, SMDS ve BISDN de, uluslararası telekomünikasyon numaralandırması için ITU-T tarafından tavsiye edilen standart. (2) E.164 formatında numaralar içeren bir ATM adresindeki alanın etiketi.

**E1** 2.048Mbps'de veri taşıyan, genelde Avrupa'da kullanılan, bir wide-area aktarım planlamasıdır. E1 aktarım hatları, özel kullanımlar için yaygın taşıyıcılardan kiralamak için uygundur.

**eBGP** External Border Gateway Protocol: Farklı otonom sistemleri arasında route bilgisi alışverişi yapmak için kullanılır.

**EEPROM** Electronically erasable programmable read-only memory: Üretimi sonrası programlanan bu kalıcı hafıza çipleri gerektiğinde, elektrik ve tekrar programlama kullanılarak silinebilmektedir. *Ayrıca bakınız: EPROM ve PROM.*

**EFCI** Explicit Forward Congestion Indication: Bir ATM ağındaki ABR servisiyle kabul edilen bir tıkanma geri besleme modu. EFCI, hazır veya kesin tıkanma durumundaki herhangi bir ağ elemanı tarafından kurulabilir. Hedefteki uç system, EFCI değerine bağlı bağlantı hücre (cell) hızlarını düşüren ve ayarlayan bir protokol çalıştırabilir. *Ayrıca bakınız: ABR.*

**EIGRP** Bakınız: *Enhanced IGRP*

**EIP** Ethernet Interface Processor: Ethernet versiyon1, Ethernet versiyon2 ve diğer arayüz işlemcilerine hızlı bir veri yolu ile IEE 802.3 arayüzlerini desteklemek için 10Mbps AUI portları sağlayan bir Cisco 7000 serisi router arayüz işlemci kartı.

**ELAN** Emulated LAN: Bir Ethernet ya da Token Ring LAN'ını daha iyi yapmak için bir kullanıcı/sunucu modeli kullanarak yapılandırılan bir ATM ağı. Çoklu ELAN'lar, tekli bir ATM ağında aynı anda bulunabilirler ve bir LAN emulation client (LEC), bir LAN emulation server (LES), bir broadcast and unknown server (BUS) ve bir LAN emulation configuration server'ın (LECS) yerine konabilmektedir. ELAN'lar, LANE düzenlemesi ile tanımlanmaktadır. *Ayrıca bakınız*: *LANE, LEC, LECS ve LES.*

**ELAP** EtherTalk Link Access Protocol: Bir EtherTalk ağında, standart Ethernet Data-link katmanı üzerinde kurulan bağlantı erişim protokolüdür.

**Enhanced IGRP (EIGRP)** Enhanced Interior Gateway Routing Protocol: Link-state ve distance-vector protokollerinin avantajlarının birleştiği, Cisco tarafından oluşturulmuş gelişmiş bir routing protokolüdür. Enhanced IGRP, yüksek operasyon kabiliyeti içeren, üstün convergence niteliğine sahiptir. *Ayrıca bakınız: IGP, OSPF ve RIP.*

**Enkapsülasyon** Bir katmanın, üzerindeki bir katmandan Protocol Data Unit'e (PDU) başlık bilgisini eklediği, katmanlı protokol tarafından kullanılan teknik. Örneğin, internet terminolojisinde, bir paket, Network katmanından (IP) bir başlık, Transport katmanından (TCP) bir başlık, uygulama protokol bilgisi ilave edilen bir Data link katmanı başlığı içerebilir.

**enterprise network** Geniş bir firma veya organizasyonda en büyük lokasyonları bağlayan, özel olarak sahip olunan ve çalıştırılan bir ağ.

**EPROM** Erasable programmable read-only memory: Üretimlerinden sonra programlanmış bu kalıcı hafıza çipleri istendiğinde, yüksek –enerjili ışık kullanılarak silinebilir ve tekrar programlanabilir. *Ayrıca bakınız: EEPROM ve PROM.*

**ESF** Extended Superframe: Her biri 192 bit olan 24 frame'den oluşur, 193. bit zamanlamayı içeren başka fonksiyonlar sağlar. Bu, SF'in gelişmiş bir versiyonudur. *Ayrıca bakınız: SF.*

**Ethernet** Xerox Corporation tarafından oluşturulan ve daha sonra Xerox, Digital Equipment Corporation ve Intel'in müşterek katkılarıyla geliştirilen banttabanlı bir düzenlemedir. Ethernet, IEEE 802.3 serisi standarda benzerdir ve CSMA/CD kullanarak, 10Mbps'de çeşitli kablo türleri üzerinde çalışır. Aynı zamanda, DIX (Digital/Intel/Xerox) Ethernet olarak bilinir. *Ayrıca bakınız: 10BaseT, Fast Ethernet ve IEEE.*

**EtherTalk** Ethernet'le AppleTalk ağlarına bağlanmasına izin verilen, Apple Computer'den bir data-link ürünü.

**Etiketli trafik** Cell loss priority- CLP biti 1'e ayarlı ATM cell'leri. Ayrıca, Frame Relay ağlarında Discard Eligible (DE) trafiğe işaret eder. Etiketli trafik, şayet trafik tıkanık ise yüksek öncelikli trafiğin hatasız taşınmasından emin olmak için gözardı edilebilir. *Ayrıca bakınız: CLP.*

**excess burst değeri** Kullanıcıların, tanımlı limiti aşabileceği trafik miktarı.

**Excess değeri** ATM ağ kurulumunda, trafik aşımında bir bağlantının garanti edilmiş değeri. Ağ kaynaklarının kullanılabilirliğine bağlı olarak, aşırı trafik, tıkanıklık boyunca iptal edilebilir. *Mukayese ediniz*: *Maximum rate.*

**EXEC oturumu** Komut-satırı arayüzünü açıklamak için kullanılan bir Cisco terimidir.User ve privileged modunda EXEC oturumu mevcuttur .

**explorer frame** Bir frame aktarılmadan önce, uzak köprü ağına route bulmak için kaynak route köprülemeyle kullanılmaktadır.

**explorer paketi** Bir kaynak-route-köprülü ağ üzerinde bir yol bulmak için bir kaynak Token Ring cihazının aktardığı bir SNA paketi.

**extended IP access listesi** Mantıksal adres, Network katmanı başlığındaki protokol alanı ve hatta Transport katmanı başlığındaki port alanı ile ağı filtreleyen IP access listesi.

**extended IPX access listesi** Mantıksal IPX adresi, Network katmanı başlığındaki protokol alanı ve hatta Transport katmanı başlığındaki soket numarası ile ağı filtreleyen IPX access listesi.

**Extended Setup** Basic Setup modundan daha fazla detayla router'ı yapılandırmak için kurulum modunda kullanılır.

**failure domain** Token ring ağlarında bir hatanın olduğu bölge. Bir istasyon, ağda, kablo kırılması gibi ciddi bir sorun olduğu bilgisine ulaştığında, hatayı NAUN'u ve aradaki her bilginin olduğu istasyon raporunu içeren bir işaret frame'i gönderir. Bu failure domain olarak tanımlanır. Daha sonra işaretleme, otoyapılandırma olarak bilinen işlemi başlatır. *Ayrıca bakınız: Autoreconfiguration* ve *beacon.*

**Fallback** ATM ağlarında, bu mekanizma, kullanılan alışılmış bir yöntem uygulanamıyorsa, bir yola erişmek için kullanılmaktadır.

**Fast Ethernet** 100Mbps hızında bir Ethernet düzenlemesidir. Fast Ethernet, MAC mekanizmaları, MTU ve frame formatı gibi özellikler korunurak, 10BaseT den 10 kat daha hızlıdır. Bu benzerlikler onu, Fast Ethernet ağlarında, mevcut 10BaseT uygulamaları ve yönetim araçları ile uyumlu kılar. Fast Ethernet, IEEE 802.3 düzenlemesinin bir uzantısına dayanmaktadır. (IEEE 802.3u). *Mukayese ediniz: Ethernet, Ayrıca bakınız: 100BaseT, 100BaseTX ve IEEE.*

**fast switching – hızlı anahtarlama** Bir router boyunca hızlı paket anahtarlama yapmak için bir route önbelleği kullanan Cisco özelliği. Process switching ile zıttır.

**FDDI** Fiber Distributed Data Interface: 200Mbps'a kadar hızda çalışabilen ve fiber-optik kabloda token-passing ortam girişi kullanan, ANSI X3T9.5 tarafından tanımlı bir LAN standardıdır. Yedekleme için FDDI bir dual-ring mimarisi kullanabilir.

**FDM** Frequency-Division Multiplexing: Frekansa bağlı bir kabloda bant genişliği atanabilmesi için farklı kanallardan bilgiyi mümkün kılan bir tekniktir. *Ayrıca bakınız: TDM, ATDM ve and statistical multiplexing.*

**FECN** Forward Explicit Congestion Notification: Kaynaktan hedefe bir yol boyunca tıkanıklığa rastlandığını DTE reseptörünü bilgilendiren, Frame Relay ağı boyunca bir bit seti. FECN bit setiyle frame alan bir cihaz, gerektiğinde akış-kontrolü başlatmak için yüksek-öncelikli protokoller isteyebilir. *Ayrıca bakınız: BECN.*

**FEIP** Fast Ethernet Interface Processor: İki 100Mbps 100BaseT portuna kadar destekleyen, Cisco 7000 serisi router'larda çalışan bir interface işlemcisi.

**Filtreleme** Access listler ile ağdaki güvenliği sağlamak için kullanılmaktadır.

**Firewall** Özel ağın güvenliğinden emin olmak için access listler ve diğer yöntemleri kullanan – bir router veya access server ya da birkaç router veya access sunucunun yerine geçen- bağlı genel ağlar ile özel bir ağ arasında kurulan bir engel.

**Flapping** Açılıp kapanan bir seri arayüzü açıklamak için kullanılan terim.

**Flash** Electronically erasable programmable read-only memory (EEPROM). Varsayılan, Cisco IOS'u saklamak için kullanılır.

**Flash memory** Intel tarafından geliştirilen ve diğer yarıiletken üreticilerine lisanslanan kalıcı olmayan depolama ürünü, elektronik olarak silinebilir ve programlanabilir, fiziki olarak bir EEPROM çipine yerleştirilir. Flash bellek, istendiğinde, yazılım imajlarının saklanmasını, önyüklenmesini ve tekrar yazılmasını mümkün kılar. Cisco router ve switch'ler, varsayılan, IOS'u tutmak için flash bellek kullanır. *Ayrıca bakınız: EPROM ve EEPROM.*

**Flat network** Geniş bir collision domain ve geniş broadcast domain olan ağ.

**Flooding** Bir arayüzden bir trafik alındığında, bu, trafiğin başladığı cihaza bağlı arayüz dışındaki tüm arayüzlerden aktarılır. Bu teknik, ağ boyunca bridge ve switch'ler yardımıyla trafik transferi için kullanılmaktadır.

**forward/filter kararları** Arayüzden bir frame alındığında, switch, hedef donanım adresine bakar ve MAC veritabanındaki çıkış arayüzünü bulur. Frame sadece belirlenen hedef portundan gönderilir.

**FQDN** Fully qualified domain name: İnternette isimden-IP adresine çözümleme sağlamak için DNS domain yapısında kullanılır. Bir FQDN örneği, bob.acme.com'dur.

**FRAD** Frame Relay access device: LAN ve Frame Relay WAN arasında bir bağlantı sağlayan herhangi bir cihaz. *Ayrıca bakınız: Cisco FRAD ve FRAS.*

**Fragment** Bilerek daha ufak parçalara bölünen geniş bir paketin bir parçası. Bir paket parçası, mutlaka bir hata belirtmez ve kasıtlı olabilir.

**Fragmentation** Daha geniş paket boyutunu desteklemeyen orta büyüklükte bir ağ ortamına veri gönderildiğinde bir paketin kasten daha ufak parçalara bölünmesi işlemi.

**FragmentFree** Parçalanma olmadığından emin olmak için frame'in veri bölümünü okuyan LAN switch türüdür. Bazen, modified-cut-through olarak söylenir.

**Frame** Bir aktarım ortamına Data Link katmanı tarafından gönderilen bilginin mantıksal bir birimi.

**Frame çeşitleri** Hangi paketin lokal ağda çalışacağını tespit etmek için LAN'larda kullanılır. Ethernet, dört farklı frame çeşidi sağlar. Bunlar birbirleriyle uyumlu değillerdir, bu nedenle iki kullanıcının iletişime geçmesi için aynı frame türünü kullanması gerekmektedir.

**Frame filtreleme** Daha fazla bant genişliği sağlamak için 2.katman switch'lerde kullanılan Frame filtrelemesi. Bir switch, frame'in hedef donanım adresini okur ve switch tarafından oluşturulmuş filtre tablosuna bakar. Daha sonra, frame'i sadece donanım adresinin bulunduğu porttan gönderir ve diğer portlar frame'i görmezler.

**Frame kimliği (frame tagging)** VLAN'ler birçok bağlı switche arasında yayılabilir, Cisco buna switch fabric demektedir. Bu switch fabric içerisindeki switch'ler frame'lerin izlerini, switch'lerin portlarından alınıyor gibi saklamak zorundadır ve frameler, bu switch fabric'ten geçiyorlar gibi, ait oldukları VLAN'lerin izlerini saklamak zorundadırlar.

**Frame Relay bridging** RFC 1490 ile tanımlı bu köprüleme yöntemi, diğer köprüleme işlemleri gibi aynı spanning-tree algoritması kullanır, fakat bir Frame Relay ağı boyunca aktarım için paketlerin enkapsüle edilmelerini mümkün kılar.

**Frame Relay switching** Bir servis sağlayıcı tarafından sağlanan, Frame Relay switch'leri için paket anahtarlaması.

**Frame Relay** X25 protokolünün, daha kullanışlı bir yenilemesi (Veri taşınmasını garanti eden bir ilişkisiz paket relay teknolojisi). Bağlı mekanizmalar arasında çoklu sanal devreler ve protokoller sunan Frame Relay, bir endüstri standardıdır, paylaşımlı giriş, çok iyi performans, anahtarlanmış Data link katmanı enkapsülasyonu sağlar.

**Frame tagging** Bakınız: Frame identification

**framing** OSI modelinin Data Link katmanında enkapsülasyonu. Paketler, başlık ve kuyrukları ile beraber enkapsüle edildiklerinden, framing olarak adlandırılır.

**FRAS** Frame Relay Access Support: Bir Frame Relay ağında, SDLC, Ethernet, Token Ring ve Frame Relay-ekli IBM cihazlarına diğer IBM mekanizmalarıyla bağlanma imkanı veren bir Cisco IOS yazılım özelliği.

**Frekans** Hertz ile ölçülen (saniyedeki devir), zaman birimi başına değişen sinyal devir sayısı.

**FSIP** Fast Serial Interface Processor: Dört veya sekiz yüksek-hızda seri portu sağlayan, Cisco 7000 router'ların varsayılan seri interface işlemcisi.

**FTP** File Transfer Protocol: Ağ düğümleri arasında dosya aktarılması için kullanılan bir TCP/IP protokolüdür. Geniş bir dosya çeşidi aralığını destekler ve RFC 959'la tanımlıdır. *Ayrıca bakınız: TFTP.*

**Full duplex** Aynı zamanda, gönderen istasyonla alan bir birim arasında bilgi aktarmak için kapasite.

**Full mesh** Her düğümün, onu diğer ağ düğümlerine bağlayan bir fiziksel ya da sanal devreye sahip olduğu bir ağ topoloji türü. Bir full mesh, yedekleme için çok iyi bir olanak sağlar, fakat maliyetinden dolayı, tipik olarak, ağ omurgaları için kullanılır. *Ayrıca bakınız: Partial mesh.*

**Gecikme** Göndericinin bir işlem başlatması ile aldıkları ilk cevap arasında geçen zaman. Aynı zamanda, bir yol üzerinde, bir paketin kaynağından hedefine kadar hareketi için gereken zaman. *Ayrıca bakınız: Latency.*

**Gecikme süresi (latency)** Yaygın olarak, bir lokasyondan diğerine bir veri paketi almak için geçen zamandır. Özel ağ kurulumu bağlamında, (1) bir cihazla bir ağa erişim için bir isteğin uygulanması ile gerçekten mekanizmanın aktarımı kabul ettiğinde geçen zamandır veya (2) bir mekanizmanın bir frame alması ve frame'in hedef portundan gönderilmesi arasında geçen zamandır.

**genişleme** Bir algoritmayla sıkıştırılmış verinin yöneltildiği, bilginin orjinal boyutuna getirildiği işleyiş.

**Genlik** Analog veya dijital bir dalganın en yüksek değeri.

**global komut** Router yapılandırmasını değiştirmek için kullanılan komutları tanımlamak için kullanılan Cisco terimidir ve tüm router'ı etkiler. Tersine, bir interface komutu sadece yapılandırıldığı interface'i etkiler.

**GMII** Gigabit MMI: Veri transferinin bir anında 8 bit sağlayan Media Independent Interface.

**GNS** Get Nearest Server: Bir IPX ağında, Verilen türde en yakın aktif sunucunun lokasyonunu belirlemek için bir müşteri tarafından gönderilen bir istek paketi. Bir IPX ağ kullanıcısı, bağlı bir sunucudan direk bir cevap ya da internetwork'teki servisin lokasyonunu söyleyen bir router'dan cevap almak için bir GNS'e ulaşır. GNS, IPX ve SAP'nin bir parçasıdır. *Ayrıca bakınız: IPX ve SAP.*

**Grafting** Pruning işlemi ile deaktif edilen bir interface'i aktif hale getiren bir işlem.

**GRE** Generic Routing Encapsulation: Uzak noktadaki bir IP ağı üzerinde Cisco router'lara point-to-point bir sanal bağlantı oluşturarak, IP tünellerine, geniş bir protokol türünü enkapsüle kapasitesiyle Cisco tarafından oluşturulan bir tünelleme protokolü. GRE kullanan IP tünellemesi, tekli-protokol bir omurga ortamında, çoklu-protokol altağları bağlantılarıyla tekli-protokol omurga ortamı üzerinde ağ genişlemesini mümkün kılar.

**Guardband** İki iletişim kanalı arasında bulunan, kullanılmayan frekans alanı. İkisi arasında parazitten kaçınmak için gerekli alanı sağlar.

**Güvenli burst** Bir ATM ağında, en geniş, geçici onaylanan veri burst arttırımıdır.

**Güvenli multicast** EIGRP, multicast trafiği gönderdiğinde, class D adresi 224.0.0.10'u kullanır. Her router, komşularının kim olduğuyla ilgilenir ve gönderdiği her multicast için yanıt gönderen komşularının listesini tutar. Şayet EIGRP, bir komşusundan cevap alamazsa, aynı veriyi unicast kullanarak anahtarlayacaktır. Şayet, 16 unicast denemesinden sonra hala bir cevap almıyorsa, komşu ölü olarak bildirilir. İnsanlar sıkça, bu işlemi güvenli multicast olarak belirtirler.

**H kanalı** High-speed channel: 384Kbps hızda çalışan, full-dupplex bir ISDN birincil hız kanalı.

**Half duplex** Gönderen ve alan birim arasında bir seferde yalnız bir yönde veri transferi kapasitesi.

**Handshake** Senkronize operasyonlardan emin olmak için ağdaki bir veya daha fazla cihaz arasında gidip gelen aktarımların bir serisi.

**harici EIGRP route** Normal olarak, bir EIGRP route'un administrative distance'ı 90'dır, fakat bu sadece, bir dahili EIGRP route olarak bilinen tipi için doğrudur. Bunlar, aynı otonom sisteminin üyesi olan EIGRP router'larıyla özel bir otonom sistemden çıkan route'lardır. Diğer route çeşidi, harici EIGRP route olarak bilinir ve 170 administrative distance'a sahiptir. Bu routle'lar, elle veya otomatik redistribution yöntemi ile EIGRP route tablosunda görünürler ve EIGRP otonom sisteminin dışından gelen ağları gösterirler.

**Hata töleransı** İletişimde kesinti olmaksızın bir ağ cihazı veya bir iletişim hattının başarısız olabileceği miktar. Hata töleransı, uzak bir ağa ikinci bir route eklenerek sağlanabilmektedir.

**HDLC** High-Level Data-Link Control: Frame karakterleri kullanan, checksum'lar içeren HDLC, senkron seri hatlarında veri enkapsülasyonu için bir yöntem getirir ve Cisco router'lar için varsayılan enkapsülasyondur. HDLC, ISO tarafından oluşturulan bir bit-tabanlı senkron Data Link katman protokolüdür ve SDLC'den türemiştir. Bununla beraber, birçok HDLC üretici uygulamaları (Cisco'nunkileri içeren) tescillidir. *Ayrıca bakınız: SDLC.*

**Hedef adresi**Bir paket alacak ağ cihaz(lar)ının adresi.

**Helper adresi** Bir sunucuya yöneltilmiş bir unicast'e bir servis için kullanıcının lokal broadcast isteğini değiştirmek için Cisco router'ı yapılandırmada kullanılan unicast adresi.

**HIP** HSSI Interface Processor: T3 veya E3'e kadar hızlarda ATM, SMDS, Frame Relay veya özel hatlara bağlantıları destekleyen bir HSSI portu sağlayan, Cisco 7000 serisi router'larda kullanılan bir interface işlemcisi.

**Hızlandırılmış taşıma** Farklı bir ağ cihazında, diğer katmanlarla ya da aynı protokol katmanıyla iletişime geçen bir protokol katmanı ile belirtilir. Daha hızlı işlemden geçecek, tanımlı veri gerektiren bir opsiyondur.

**Hibrit routing protokolü** Distance-vector ve link-state'in her ikisinin niteliklerini kullanan routing protokolüdür. Enhanced Interior Gateway Routing Protocol (Enhanced IGRP).

**hiyerarşi** IP adreslemesinin tanımlanmasında kullanılan terim; hiyerarşik adreslemede, bazı bitler ağlar ve bazı bitler kullanıcı adreslemesi için kullanılır. Aynı zamanda, DNS yapısında ve Cisco tasarım modelinde kullanılır.

**Hiyerarşik adresleme** Lokasyonu saptamak için mantıksal bir komut zinciri çalıştıran bir adres planlaması. IP adresleri, uygun hedeflere paketleri yöneltmek için ağ adresleri, subnet numaraları ve kullanıcı numaralarının bir hiyerarşisinden oluşturulmaktadır.

**Holddown** Tanımlı bir zaman aralığında, ne route yayınlayan ne de yayınları kabul eden bir router'da olan bir route durumu. Holddown'lar, geçersiz bilgi almaktan kaçınmak için kullanılmaktadır. Asıl bilgi geçerli olabilir, fakat güvenilmezdir. Bağlantılarından biri çalışmadığında, bir route, genellikle holddown'da yer almaktadır.

**Hop** Herhangi iki ağ düğümü arasındaki bir paketin hareketi. Ayrıca bakınız: hop sayısı.

**Hop sayısı** Yoldaki router'ların sayısına bağlı, bir kaynak ve bir hedef arasındaki uzaklığı hesap eden bir routing metriği. RIP, tek metriği olarak hop sayısını kullanır. *Ayrıca bakınız: hop ve RIP.*

**Host-to-Host layer** OSI modelinin Transport katmanına eşdeğer İnternet Protokol ailesindeki katman.

**HSCI** High-Speed Communication Interface: Cisco tarafından geliştirilen, 52 Mbps'a kadar hızlarda full-duplex senkron seri iletişim kapasitesi sağlayan interface

**HSRP** Hot Standby Router Protocol: Yönetici müdahelesi olmaksızın yüksek ağ kullanılabilirliği ve neredeyse anlık donanım yedekliliği sağlayan bir protokol.Öncü router içeren bir Hot Standby router grubu oluşturulur. Şayet öncü router çalışmazsa, yerine, diğer router'lardan biri – standby router'lar- geçecektir.

**HSSI** High-Speed Serial Interface: 52Mbps'a kadar hızlardaki bir WAN üzerinde yüksek-hızda seri bağlantı için bir ağ standart fiziksel konnektörü.

**Hub'lar** Gerçekte sadece çok portlu repeater olan physical katman cihazlarıdır. Bir porttan elektronik dijital bir sinyal alındığında, sinyal, tekrar kuvvetlendirilir, tekrar üretilir ve sinyalin alındığı segment dışındaki tüm segmentlere gönderilir.

**hücre** ATM ağ kurulumunda, switching ve çoklama için temel veri birimidir. Hücreler, hücrenin veri akımını gösteren 5-byte başlığı ve 48-byte yükü içeren tanımlı bir 53-byte uzunluğa sahiptir. Ayrıca bakınız: hücre gecikmesi

**Hücre gecikmesi** Hücre olarak bilinen sabit boyuttaki ufak paketleri kullanan bir teknolojidir. Sabit uzunlukları, hücrelerin işlemden geçmelerine ve yüksek hızlardaki donanımlarda anahtarlanmalarına imkan verir. Bu, ATM ve diğer yüksek-hız ağ protokolleri için bu teknolojiyi temel kılar. *Ayrıca bakınız: Hücre.*

**ICD** International Code Designator: Adresleme altağ modelinden uyarlanmıştır. Bu, ağ katmanı adreslerini ATM adreslerine adreslemeyi belirler. ICD, özel ağlarla kullanılmak için ATM Forumu tarafından oluşturulan adreslemede kullanılan iki ATM formatından biridir. *Ayrıca bakınız: DCC.*

**ICMP** Internet Control Message Protocol: RFC 792 de belgelenen, Hataları raporlama ve IP paket işlemlerine uygun bilgi sağlama amacı için bir Network katmanı internet protokolüdür.

**IEEE 802.1** Bridge grubunu tanımlayan, IEEE komisyon düzenlemesi. STP (Spanning Tree Protokolü) için düzenleme, IEEE 802.1D'dir. STP, bridge kullanan ağlarda ağ kısır döngülerini bulmak ve engellemek için STA (spanning-tree algoritması ) kullanır. VLAN trunking için düzenleme, IEE 802.1Q'dur.

**IEEE 802.3** Özellikle orjinal 10Mbps standardı olan, Ethernet grubunu açıklayan IEEE komisyon düzenlemesidir. Ethernet, physical katman ve MAC altkatman ortam erişimini belirten bir LAN protokolüdür. IEEE 802.3, aynı ağdaki birçok cihaz için erişim sağlamak için CSMA/CD kullanır. FastEthernet, 802.3U ve GigabitEthernet, 802.3Q olarak tanımlanmıştır. *Ayrıca bakınız: CSMA/CD.*

**IEEE 802.5** Token Ring ortam erişimini açıklayan IEEE komisyonu.

**IEEE** Institute of Electrical and Electronics Engineers: Diğer aktiviteleri arasında, ağ kurulumu ve haberleşmeyi de içeren, bilgi işlem ve elektronikteki birçok alanın standartlarını tanımlayan, profesyonel bir organizasyondur. IEEE standartları, endüstride bugün kullanılan hakim LAN standartlarıdır. Birçok protokol, yaygın olarak, uygun IEEE standartının referans numarasıyla bilinmektedir.

**IGMP** Internet Group Management Protocol: IP kullanıcıları tarafından kullanılan, komşu bir multicast router'a, multicast grup üyeliklerini raporlayan protokol.

**IGP** Interior gateway protocol: Bağımsız bir sistemdeki routing verisini değiş tokuş etmek için bir ağtopluluğu tarafından kullanılan protokol. Örnekleri RIP, IGRP ve OSPF'tir.

**ILMI** Integrated (veya Interim) Lokal Management Interface: ATM INI'ye ağ-yönetim kabiliyetini eklemek için görevlendirilmiş, ATM Forumu tarafından oluşturulmuş bir düzenleme. Entegre Lokal Yönetim Arayüzü hücreleri, ATM sistemleri arasında otomatik yapılandırma sağlar. LAN emülasyonunda, ILMI, bir LECS bulmak için ATM uç istasyonu için yeterli bilgiyi sağlayabilir. ILMI, uç istasyonlara, ATM NSAP (Network Service Access Point) prefix bilgisi sağlar.

**Inside Network ( İç ağ)** NAT terminolojisinde, iç ağ, aktarıma tabi, ağların grubudur. Dış ağlar, diğer tüm adresleri belirtir – bunlar genelde İnternette bulunurlar.

**Interface işlemcisi** Cisco 7000 serisi router'larda kullanılan çeşitli işlemci modüllerinden biri.

**Interface yapılandırma modu** Bir IP adresi veya maskesi gibi özel bilgilerle bir Cisco router veya switch'i yapılandırmanız için olanak veren mod.

**Intermediate System to Intermediate System (IS-IS)** Intermediate System–to–Intermediate System: Bir OSI link-state hiyerarşik routing protokolüdür.

**Internet katmanı** Bir internetwork boyunca ağ adreslemesi ve routing sağlayan protokollerin katmanı.

**Internet** Popülaritesi, 1990'ların ortasında yayılmaya başlayan, evrensel ağların ağıdır. Orijinal olarak, ortak akademik araştırma için bir araç olan internet, tüm bilgi çeşitlerinin değiş tokuş edildiği ve dağıtıldığı bir ortam oldu. İnternetin, tamamen farklı bilgisayar platform ve teknolojilerine bağlanma ihtiyacı, tek tip protokol ve standart gelişimine öncülük yaptı ki aynı zamanda firma LAN'larında yaygın olarak kullanılması başladı. *Ayrıca bakınız: TCP/IP ve MBONE.*

**Internet protokolü (IP)** TCP/IP protokol yığınına ait bir protokol. *Ayrıca bakınız: TCP/IP.*

**Inverse ARP** Inverse Address Resolution Protocol: Mantıksal ağ adresininin yerini belirlemek ve onu kalıcı bir sanal devre (PVC) ile ilişkilendirmek için, router gibi bir ağ cihazına izin veren, bir ağda dinamik adres tespitinin yapıldığı bir teknik. Yaygın olarak, lokal DLCI üzerinde reverse ARP istekleri göndererek uzak-uç düğümünün TCP/IP adresini belirlemek için Frame Relay'de kullanılır.

**IP adresi** Sık olarak bir internet adresi olarak belirtilir. Bu, internetteki (veya herhangi bir TCP/IP ağında) herhangi bir cihazı (host) eşsiz olarak tanımlayan bir adrestir. Her bir adres, noktalarla ayrılan desimal numaralar olarak gösterilen dört oktetten oluşmaktadır ("noktalı-desimal" olarak bilinen bir format). Her adres, bir ağ numarası, isteğe bağlı bir altağ ve bir host numarasından oluşmuştur. Host adresi, ağ veya altağdaki özel bir düğümü adreslerken, ağ ve altağ numaraları, beraber, routing için kullanılmaktadır. Ağ ve altağ bilgisi, subnet maskesi kullanılarak IP adresinden çıkarılır. A'dan C'ye klasların, adreslerin ağ, altağ ve host bölümlerine farklı bitler ayırdığı, beş IP adres klası vardır (A-E). Ayrıca bakınız: CIDR, IP ve subnet maskesi.

**IP** Internet Protokolü: RFC 791 de tanımlı, TCP/IP yığının parçası ve bağlantısız servis öneren bir Ağ katmanı protokolüdür. IP, adresleme, servis tipi teknik özellikleri, parçalama ve tekrar biraraya getirme ile güvenlik için özelliklerin düzenlenmesini sağlar.

**IP multicast** Bir kaynaktan çeşitli uç noktalara veya çoklu kaynaklardan birçok hedefe IP trafiğinin tekrar çoğaltılmasını mümkün kılan bir routing tekniği. Her özel hedef noktasına bir paket yerine, grup için sadece bir IP uçnoktası belirtilen bir multicast grubuna bir paket gönderilir.

**IPCP** IP Control Program: PPP üzerinde IP yapılandırmak ve kurmak için kullanılan protokol. *Ayrıca bakınız: IP ve PPP.*

**IPX** Internetwork Packet Exchange: Sunuculardan, iş istasyonlarına bilgi transfer edilmesi için Novell NetWare ağlarında kullanılan Network katmanı (3.katman) protokolüdür. IP ve XNS'e benzer. *Ayrıca bakınız: IPX ve PPP.*

**IPXCP** IPX Control Protocol: PPP üzerinde IPX yapılandırmak ve kurmak için kullanılan protokol.

**IPXWAN** IPX kullanılan bir bağlantıda hat tercihlerini görüşmek ve sağlamak için yeni bir WAN bağlantısında kullanılan protokol. Bağlantı çalışır olduktan ve tercihler, iki uçtan-uca bağlantı tarafından üzerinde mutabık kalındıktan sonra, normal IPX aktarımı başlar.

**ISDN** Integrated Services Digital Network: Telefon ağlarında, veri, ses ve diğer dijital trafiği taşımaya izin veren bir iletişim protokolü olan ISDN, telefon firmaları tarafından bir servis olarak önerilir. *Ayrıca bakınız: BISDN, BRI ve PRI.*

**IS-IS** *Bakınız: Intermediate System–to–Intermediate System (IS-IS)*

**ISL routing** Inter-Switch Link routing: Switch kullanılan bir ağda frame etiketlemenin Cisco tescilli bir yöntemi. Frame etiketleme (tagging), switch'ler kullanılan bir ağ topluluğunda,switch'ler arası geçişlerde bir frame'in VLAN üyeliğini belirlemenin bir yoludur.

**ITU-T** International Telecommunication Union-Telecommunication Standardization Sector: Bu, telekomünikasyon teknolojileri için dünyada geçerli standartlar geliştiren bir mühendisler topluluğudur.

**interarea routing** İki veya daha fazla mantıksal area arasındaki routing. *Tersi: Alan-içi routing, Ayrıca bakınız: Area.*

**internet** Internetin yayılmasından önce, bu küçük harfli şekli, genel anlamda "internetwork" için bir kısaltmaydı. *Ayrıca bakınız: Internetwork.*

**Internetwork (Ağlartopluluğu)** Tipik olarak, tek bir yapı gibi çalışan, router veya diğer mekanizmalarla birbirine bağlı ağların bir grubu.

**internetwork iletişim** Yaygın olarak, ağları birbirine bağlayan genel görevlerle ilgili her şey. Teknolojileri, işleyişleri ve ürünleri içeren terim. Ağları, bir router'a bağladığınızda, bir ağtopluluğu oluşturuyorsunuz.

**intra-area routing** Bir mantıksal alanda olan routing. *Tersidir: Inter-area routing.*

**isochronous aktarım** Güvenli aktarım için sabit bir bit oranı gerektiren, senkron bir veri hattı üzerinde asenkron veri transferi.

**İstatistiksel multiplexing** Genelde multiplexing çoklu mantıksal kanallardan alınan bilgiyi, tek bir fiziksel kanal üzerinden göndermeye izin veren bir tekniktir. İstatistiksel multiplexing, dinamik olarak aktif olan giriş kanallarına bant genişliği tanımlar. Uygun bant genişliğini optimize eder, böylece diğer multiplexing teknikleriyle olandan daha fazla cihaz bağlanabilir. Ayrıca, istatistiksel time-division multiplexing veya stat mux olarak da bilinir.

**Kablo aralığı** Genişletilmiş bir AppleTalk ağında, ağdaki mevcut düğümler tarafından kullanmak için tahsis edilmiş numara aralığı. Kablo aralık değeri, tek bir ağ numarasından, farklı büyüklükte ağ numara sırasına kadar olabilir. Düğüm adresleri, kablo aralık değerleri ile tanımlanmaktadır.

**Katman 3 switch** *Ayrıca bakınız: Çok katmanlı switch.*

**Katman** Ağdaki aktarım için verinin enkapsüle edilmesinde, OSI modelinin nasıl çalıştığını hiyerarşik olarak belirtmek için kullanılan terim.

**Katmanlı mimari** Bir ağda çalışması için geliştirilen uygulamaların, endüstri standardı yaklaşımı. Katmanlı mimari, uygulama geliştiricilerin, tüm program yerine sadece bir katmanda değişiklik yapmalarına izin verir.

**Kenar cihaz** (Ethernet ve Token Ring gibi) eski arayüzler ile Data-link ve Network katmanına dayalı ATM arayüzleri arasında paket routing'ine olanak veren bir cihaz. Bir kenar cihazı, bir

Network katmanı routing protokolü çalışmasının parçası olmaz; seyrek olarak, gereken iletim bilgisini almak için route açıklama protokolü kullanır.

**Kerberos** Verinin, ağın dışında izlenememesinden emin olmak için Cisco router tarafından kullanılabilen bir kimlik denetleme ve şifreleme yöntemidir. Kerberos, MIT'de geliştirildi ve Data Encryption Standard (DES) şifreleme algoritması kullanılarak güçlü güvenlik sağlamak için tasarlandı.

**Kısmi mesh** Bazı ağ düğümlerinin, bir tam mesh (her düğümün, diğer tüm ağ düğümlerine bağlandığı fiziksel ya da sanal devreye sahip) oluşturduğu, fakat diğerlerinin, sadece ağdaki bir veya iki düğüme bağlandığı bir ağ topolojisi türüdür.

**Kiralık hat (leased-line)** Telefon firmalarından kiralanan iki nokta arasındaki kalıcı bağlantı.

**Klas A ağ** IP'nin hiyerarşik adresleme planının bir parçasıdır. Klas A ağlar, ağları belirtmek için sadece 8 bit ve her bir ağdaki kullanıcı ve subneti belirtmek için 24 bite sahiptir.

**Klas B ağ** IP'ninhiyerarşik adresleme planının bir parçasıdır. Klas A ağlar, ağları belirtmek için 16 bit ve her bir ağdaki kullanıcı ve subnet belirtmek için 16 bite sahiptir.

**Klas C ağ** IP'ninhiyerarşik adresleme planının bir parçasıdır. Klas A ağlar, ağları belirtmek için 24 bit ve her bir ağdaki kullanıcı ve subnet belirtmek için sadece 8 bite sahiptir.

**Konsol portu** Tipik olarak, komut-satırı arayüz özelliğine izin veren, bir Cisco router ve switchteki RJ-45 (8 pin modüler) port.

**Kullanıcı adresi** Bir cihazda, sunucu ya da bir yönetici tarafından yapılandırılan mantıksal adres.

**Kuyruk** Yaygın olarak, bir sinema salonuna girmek için bekleyen insanların bir sırası gibi, işleyiş için hazır ve düzenli olarak yerleştirilmiş öğelerin listesi. Routing'de, bir router interface'de aktarılmak için hat üzerinde bekleyen bilgi paketlerinin birikimine işaret eder.

**LAN** Lokal area network: Yaygın olarak, limitli coğrafi bir alandaki (birkaç kilometreye kadar) ilgili cihazlar ve iki ya da daha fazla bilgisayarı bağlayan bir ağ. LAN'lar, tipik olarak, bir firmadaki, yüksek-hızlı, düşük-hatlı ağlardır. OSI'nin Fiziksel ve Veri Hattı katmanlarındaki kablolama ve sinyalleşme, LAN standartları tarafından dikte edilmektedir. Ethernet, FDDI ve Token Ring, en yaygın LAN teknolojilerindendir. *Mukayese ediniz: MAN*

**LAN switch** Genel olarak, özellikle bir Ethernet switch olarak belirtilen, veri hattı segmentleri arasındaki paketleri aktaran, yüksek-hızlı, çoklu interface'i olan bir köprüleme mekanizmasıdır. LAN switch'ler, MAC adres tabanlı trafiği transfer ederler. *Ayrıca bakınız: Çokkatmanlı switch ve store-and-forward paket anahtarlama.*

**LANE** LAN emulation: ATM ağlarına, bir LAN omurgası gibi çalışma imkanı veren teknoloji. Bunu yapmak için, ATM ağlarının, multicast ve broadcast desteği, adres eşleştirme (MAC'ten ATM'e) ve SVC yönetimi, ek olarak, çalışabilir bir paket formatı sağlaması gerekmektedir. LANE, Ethernet ve Token Ring ELAN'larını belirtir. *Ayrıca bakınız: ELAN.*

**LAPB** Link Accessed Procedure, Balanced: X.25'in parçası ve kökeni SDLC olan bir bit-tabanlı Data link katmanı protokolüdür*. Ayrıca bakınız: SDLC ve X.25.*

**LAPD** Link Access Procedure on the D channel: Özellikle D kanalları için kullanılan ve ITU-T Recommendations Q.920 ve Q.921 ile tanımlı, ISDN Data link katmanı protokolüdür. LAPD, LAPB'den türemiştir ve ISDN temel erişim sinyalleşme gereksinimlerine uymak için oluşturulmuştur.

**LCP** Link Control Protocol: PPP tarafından kullanmak için veri-hattı bağlantılarını kurmak, yapılandırmak ve test etmek için tasarlanmış protokol. *Ayrıca bakınız: PPP.*

**LE ARP** LAN Emulation Address Resolution Protocol: Bir MAC adresine uygun ATM adresi sağlayan protokol.

**leaky bucket** Bir kullanıcı veya ağdan hücre akışının uygunluğunu control etmek için ATM ağlarında kullanılan generic cell rate algorithm (GCRA) için bir örnekleme. Bucket'in (kovanın) "deliği", hücrelerin uyum sağlayabileceği uzatılmış değer olarak anlaşılır ve "derinlik", belirli bir zaman periyodunda hücre patlaması için bir töleranstır.

**learning bridge** Transparan olarak , dinamik bir MAC adresi veritabanı ve her bir adres ile ilgili arayüzleri kuran bir bridge'tir. Transparan bridge, ağdaki trafik tıkanıklığını düşürmeye yardımcı olur.

**LEC** LAN emulation client: Tüm üst-seviye protokol ve uygulamaların çalışması ve iletişiminin devam etmesine izin veren, link katmanı interface'in emülasyonunu sağlayan yazılım. LEC; istemci, sunucu, bridge ve router'ları içeren tüm ATM cihazlarında çalışır. *Ayrıca bakınız: ELAN ve LES.*

**LECS** LAN emulation configuration server: Emüle LAN servislerinin önemli bir parçası. LES'ten gelen istekler üzerine kurulan yapılandırma verisi sağlar. Bu servisler, Integrated Lokal Management Interface (ILMI) desteği, LES adresleri ve onların uyumlu emüle LAN tanımlayıcısı için yapılandırma desteği ve emüle LAN'a bir interface için adres kaydı içerir.

**LES** LAN emulation server: LEC bağlantısı için başlatma yapılandırma verisi sağlayan merkezi LANE bileşeni. LES tipik olarak, ya ATM-uyumlu bir router'da ya da bir switch'te bulunmaktadır. LES'in sorumlulukları, LEC için destek ve yapılandırma, LEC için adres kaydı, veritabanı depolama, ATM adresleriyle ilgili yanıtlar ve emüle LAN'lara interface oluşturmayı içerir. *Ayrıca bakınız: ELAN, LEC ve LECS.*

**Link** Belirli bir ağa atanmış bir ağ ya da router arayüzüdür. Bir interface, OSPF işlemine eklendiğinde, o, OSPF tarafından bir link olarak kabul edilir. Bu link, ya da interface, hem bir veya daha çok IP adresi hem de onun up veya down olması ile ilgili durum bilgisine sahiptir.

**Link-state algoritmaları** İnternetwork'teki her düğüme, komşularının ulaşma maliyeti hakkında bilgiyi broadcast veya multicast etmek için her router'a izin veren bir routimg algoritmasıdır. Link-state algoritmaları, ağın sürekli görünüşünü sağlar ve buna rağmen routing kısır döngülerine karşı savunmasız değildir. Bununla birlikte, bu kısır döngü olmayan ağ, hesaplamada oldukça fazla dikkatlilikle başarılmaktadır ve daha yaygın trafiklerdir (distance-vector routing algoritmaları ile mukayese edilirler). *Ayrıca bakınız: Distance-vector routing algoritması.*

**Link-state protokolleri** Shortest-path-first protokolü olarak da bilinen Link-state protokollerinde, router'ların her biri, üç ayrı tablo oluşturur. Bunlardan birisi, direk bağlı komşuların izini korur, biri, tüm ağtopluluğunun topolojisini belirtir ve diğeri, routing tablosu olarak kullanılır. Link-state router'lar, ağtopluluğu hakkında, herhangi bir distance-vector routing protokolünden daha fazla bilgiye sahiptirler.

**LLAP** Lokal Talk Link Access Ptotocol: Bir LokalTalk ortamında, verinin düğümden düğüme taşınmasını yöneten veri link-katmanı protokolüdür. Bu protokol, bus erişim yönetimi ile düğüm adreslemesi sağlar ve paket uzunluğuyla bütünlüğünden emin olmak için gönderilen ve alınan verilerin kontrolünü de yapar.

**LLC** Logical Link Control: IEEE tarafından tanımlanan, iki Veri Hattı katmanı alt katmanının üstte olanı. LLC, hata tespitinden (fakat düzeltmesinden değil), akış kontrölünden, çerçeveleme (framing) ve yazılım altkatman adreslemesinden sorumludur. Etkin LLC protokolü, IEEE 802.2, connectionless ve connection-oriented operasyonların her ikisini de açıklar.

**LMI** Lokal Management Interface: Orijinal Frame Relay düzenlemesi için bir geliştirmedir. Sağladığı özellikler arasında; keepalive mekanizması, multicast mekanizması, global adresleme ve bir durum mekanizması vardır.

**LNNI** LAN Emulation Network-to-Network Interface: Faz 2 LANE düzenlemesinde, bir ELAN'daki sunucu bileşenleri arasındaki iletişimi destekleyen bir arayüzdür.

**Load (yük)** IGRP gibi EIGRP, varsayılan olarak uzak bir ağa en iyi yolu belirlemek için sadece bant genişliği ve gecikmeyi (delay) kullanır. Bununla beraber, EIGRP, uzak bir ağa en iyi yolu bulma arayışında bant genişliği, gecikme, yük ve güvenirliğin bir kombinasyonunu da kullanabilir.

**load balancing (yük dengeleme)** Aynı uzak ağa çoklu linkler üzerinde paket yük dengelemesi hareketi.

**Lokal döngü** Switching ofisine en yakın bir sınır noktasından bağlantı.

**Lokal explorer paketi** Bir Token Ring SRB ağında, lokal ringe bağlı bir kullanıcıyı bulmak için bir uç sistem tarafından oluşturulan paket. Şayet lokal kullanıcı bulunamıyorsa, uç sistem iki çözüm üretecektir; bir spanning explorer paketi veya tüm-route'ları keşif paketi.

**LokalTalk** CSMA/CD çalışması, 230.4Kbps hızda veri aktarımı desteğine ek olarak, LokalTalk, OSI referans modelinin Data Link ve Physical katmanlarında çalışan, AppleTalk Computer'un tescilli temel bant protokolüdür.

**LSA** Link-State Advertisement: Link-state paketlerinde (LSP) taşınır. Link-state protokollerinin kullandığı bu yayınlar genel olarak multicast paketleridir. Komşu ve path cost'ları hakkında bilgi içerirler. Alıcı router'lar, link-state veritabanlarını ve routing tablolarını korumak için LSA'leri kullanırlar.

**LUNI** LAN Emulation User-to-Network Interface: LEC ile LES arasındaki arayüzü tanımlamaktadır. LUNI, ATM ağlarında LAN emülasyonu için ATM Forumu standartıdır. *Ayrıca bakınız: LES ve LECS*

**MAC adresi** Bir LAN segmentine bağlanmak için gerekli tüm port ve cihazların bir Data link katmanı donanım adresi. Bu cihazlar mantıksal adreslerin doğru lokasyonu için ağdaki cihazlar tarafından kullanılmaktadır. MAC adresleri, IEEE standartları tarafından tanımlanmaktadır ve tipik olarak lokal LAN arayüzlerinde burned-in address (BIA) kullanılır, 48 bit uzunluğundadır. Farklı şekillerde adlandırılan donanım adresleri, fiziksel adres, burned-in adresi veya MAC katmanı adresidir.

**MAC** Media Access Control: Donanım adreslemesi, ortam erişimi ve frameler için hata tespitinden sorumlu, Data link katmanının altkatmanı. *Ayrıca bakınız: Data link katmanı ve LLC.*

**MacIP** AppleTalk'ta Datagram Delivery Protocol (DDP) paketlerindeki IP paketlerini enkapsüle eden Network katmanı protokolü. MacIP, aynı zamanda yedek ARP servislerini destekler.

**Maksimum burst** Byte veya hücrelerle belirtilen, kısa bir zaman için bir ATM kalıcı sanal bağlantısında izin verilen garantili değeri geçen bilginin en geniş burst değeri. *Mukayese ediniz: garantili burst. Ayrıca bakınız: maksimum değer.*

**Maksimum değer** Trafiğin kaynağından, garantili ve garantisiz trafiğin toplamına eşit, belirli sanal bir devrede izin verilen maksimum veri throughput değeri. Trafik tıkanıklığı olduğunda, garantili bilgi, yoldan silinebilir. Saniyedeki bit veya hücre ile ölçülen maksimum değer, sanal devrenin asla taşıyamayacağı ve ortam araçlarının değerini aşamayacağı bilginin en yüksek throughput değerini belirtir.

**Maksimum hop sayısı** Sonlandırılmadan önce, bir paketin aktarılmasına izin verilen router'ların sayısı. Bu, bir paketin bir ağda sonsuza kadar dönmesini engellemek için oluşturulmuştur.

**MAN** Metropolitan area network: Bir metropolitan alanını içine alan bir ağ. Tipik olarak, LAN'dan geniş fakat, WAN'dan küçük bir alan.

**Manchester encoding** İlk bit süresinin yarılandığı esnada yüksek bir voltaj seviyesiyle belirtilen bir (1) ve saat denetimi için çalıştırılan bir orta bit süresi geçişindeki dijital kodlama yöntemi. Bu düzenleme, Ethernet ve IEEE 802.2 tarafından kullanılmaktadır.

**Mantıksal adres** Bir ağdan diğerine bir verinin nasıl gönderileceğini belirten Network katmanı adresidir. IP ve IPX, mantıksal adres örnekleridir.

**MBONE** İnternetin multicast omurgasıdır, multicast LAN'lardan oluşan sanal bir multicast ağıdır. Onları bağlayan point-to-point tünelleri içerir.

**MBS** Maximum Burst Size: Bir ATM sinyalleşme mesajında, hücre sayısı olarak kodlanan bu metrik, burst töleransını aktarmak için kullanılır.

**MCDT** Maximum Cell Transfer Delay: Bir ATM ağında, maksimum hücre gecikmesi ile link ve düğümdeki sabit gecikmenin toplamıdır. MCDT, bir ATM ağındaki uygun kaynakları doğrularken PNNI topoloji durum paketleri için kullanılan maksimum hücre gecikme değişimi toplamıdır. Her trafik sınıfına atanmış bir MCDT değeri vardır. *Ayrıca bakınız: MCDV.*

**MCDV** Maximum Cell Delay Variation: Bir ATM ağında, belirli servis katagorileri için bir link veya düğümdeki maksimum iki-nokta CDV objesi.

**MCLR** Maximum Cell Loss Ratio: Link veya düğümde ulaşan toplam hücre sayısı ile mukayese edilen, aktarmak için bir link veya düğümün başarısız olduğu, ATM ağındaki maksimum hücre oranı. MCLR, bir ATM ağının uygun kaynaklarını doğrulamak için PTSP'leri kullanmak için başvurulan dört metrikten biridir. MCLR, CLP biti sıfıra ayarlanan VBR ve CBR trafik sınıflarındaki hücrelere uygulanır. *Ayrıca bakınız: CBR,CLP ve VBR.*

**MCR** Minimum cell rate: ATM ağlarının trafik yönetimi için ATM Forumu tarafından belirtilen bir parametre. MCR, özellikle ABR aktarımları için tanımlanmaktadır ve allowed cell rate (ACR) için minimum değeri tanımlar. *Ayrıca bakınız: ACR ve PCR.*

**MIB** Management Information Base: Uzak cihazlardan bilgi toplamak için SNMP yönetim yazılımı ile kullanılmaktadır. Yönetim istasyonu, bilgi için uzak cihazı sorgulayabilir veya uzak istasyonda çalışan MIB, düzenli aralıklarda bilgi göndermek için programlanabilir.

**MII** Media Independent Interface: Bir seferde 4 ve 8 bit değerinden daha hızlı bit transferi sağlamak için Fast Ethernet ve Gigabit Ethernet'te kullanılmaktadır.

**MIP** Multichannel Interface Processor: Cisco 7000 serisi router'lardaki yerleşik interface işlemcisi. Bir CSU'ya bağlı seri kablolar tarafından iki kanallı T1 veya E1 bağlantıları sağlar. İki denetleyici, her grubun sisteme tek başına yapılandırılabilen seri bir interface olarak tanıtılmasıyla, 24 T1 veya 30 E1 kanal grubu sağlayacak kapasitededir.

**Mips** Millions of instructions per second: İşlemci hızının oranı.

**MLP** Multilink PPP: Birçok veri linkindeki datagramları bölmek, tekrar birleştirmek ve sıraya koymak için kullanılan bir tekniktir.

**MMP** Multichassis Multilink PPP: Çoklu router ve access sunucularında MLP'yi destekleyen bir protokol. MMP, birden fazla router ve access sunucularının, bir ağ adresi ve ISDN giriş numarası ile tek, geniş dial-up havuzu gibi çalışmasına izin verir. MMP, kullanıcı bağlantısı iki fiziksel erişim cihazına ayrıldığında paketleri bölme ve tekrar birleştirmeyi başarıyla destekler.

**Modem eliminator** Komutları ve gerekli sinyalleri taklit ederek modemsiz iki DTE cihazı arasındaki bir bağlantıyı mümkün kılan bir mekanizma.

**Modem** Modulator-demodulator: Dijital sinyalleri analoğa çeviren veya tersini yapan cihaz, böylece dijital bilgiler, ses telefon hatları gibi, analog iletişim olanaklarıyla aktarılabilirler. Bu, kaynaktaki

dijital sinyallerin, aktarım için analoğa çevrilmesi ve analog sinyallerin hedefteki dijital formuna tekrar dönüştürülmesi ile başarılır. *Ayrıca bakınız: Modülasyon ve demodülasyon.*

**Modülasyon** Dijital ve analog bilgileri göstermek için, genlik ve frekans gibi bazı elektrik sinyal parametrelerini değiştirme işlemi. *Ayrıca bakınız: AM.*

**MOSF** Multicast OSPF: Domainde IP multicast routing'e imkan veren OSPF unicast protokolünün bir uzantısı. *Ayrıca bakınız: OSPF.*

**MPOA** Multiprotocol over ATM: Direk bağlı kullanıcı, router ve çok katmanlı LAN switch'leriyle bir ATM ağında çalışan IP, IPv6, AppleTalk ve IPX gibi Network katmanı protokollerinin mevcut durumunu ve geleceğini standart hale getirmek için ATM Forumu tarafından yapılan bir çalışma.

**MTU** Maximum transmission unit: Bir interface'in yönetebileceği, byte'larla ölçülen en geniş paket boyutu.

**Multicast adresi** Belirli multicast protokolünde aktarılan, özel, varolmayan bir MAC adresiyle ağda birden fazla cihazı işaret eden tek bir adres.

**Multicast adresi** Multicast protokolünde aktarılan, özel ve var olmayan bir MAC adresiyle ağda birden fazla cihazı işaret eden tek bir adres.

**Multicast grup** Multicast, IP multicast grup adreslerine mesaj veya veri göndererek çalışır. Grup, multicast ile gönderilen veriyi okuması ve göndermesi için verilen kullanıcı veya istemcilerden oluşur.

**Multicast send VCC** Bir BUS'a LEC tarafından ayarlanan, iki-yönlü point-to-point bir virtual control connection'dır (VCC). Faz1 LANE tarafından belirtilen bilgilendirme linklerinin üç çeşidinden biridir.

**Multicast** Yaygın olarak, tek bir verici ile çoklu alıcılar arasındaki bir bağlantıdır. Broadcast adreslerin aksine, ağdaki tüm adreslere gönderilen, multicast mesajlar, ağ adreslerinin tanımlı bir altkümesine gönderilir, bu altküme, paketin hedef adresinde gösterilen bir grup multicast adresine sahiptir. *Ayrıca bakınız: Broadcast ve directed broadcast.*

**Multi-layer Switch (Çok katmanlı switch)** Oldukça geliştirilmiş, yüksek-hızlı, donanım bazlı bir LAN router çeşidi. Cihaz, katman2 ve katman3 adreslerine bağlı olarak paketleri filtreler ve gönderir. Katman4'ü okuyabilmesi dahi mükündür. Bazen, 3.katman switch olarak tanımlanmaktadır. *Ayrıca bakınız: LAN switch.*

**Multi-link** Birleşik bant genişliği sağlamak için çoklu asenkron veya ISDN linklerini birleştirmekte kullanılmaktadır.

**Multiplexing** Tek fiziksel kanal üzerinden aktarım için çoklu mantıksal sinyalleri fiziksel bir sinyale çevirme işlemi.

**NACK** Negative acknowledgment: Bir alıcıdan gönderilen cevap. Vericiye bilginin alınmadığını ya da hata içerdiğini söyler. *Mukayese ediniz: Acknowledgment .*

**Named access list** Numaraları kullanmak yerine listelere isim vermenize izin vererek access listlerin yönetimine yardım etmek için hem standart hem de extended listlerde kullanılmaktadır. Bu, normalde numaralı access listlerde mümkün olmayan, tek bir access list satırının değiştirilmesi için de size izin verir.

**NAT** Network Address Translation: Evrensel benzersiz IP adres gereksinimlerini minimize etmekte faydalı bir algoritma. Kuruluşlara, public olmayan adreslerini, evrensel yönlenebilir adres aralığına çevirmeleri imkanı verir.

**Native VLAN** Cisco switch'lerin hepsi, VLAN1 olarak adlandırılan bir native VLAN'e sahiptir. Bu, herhangi bir yolla silinemez ve değiştirilemez. Tüm switch portları, varsayılan olarak VLAN1'dedir.

**NBP** Name Binding Protocol: AppleTalk'ta, bir karakter dizgisi olarak girilen soket istemci ismini, uygun DDP adresine çeviren transport-seviyesi protokolü. NBP, AppleTalk protokollerine, isimlerini uygun soket adreslerine eşleştiren çeviri tablolarını gösteren ve koruyan kullanıcı tanımlı bölgeleri ve mekanizma isimlerini seçmek için kapasite sağlar.

**Neighbor router'lar** OSPF'te, genel ağa bakan interface sahibi iki router. Çok erişimli ağlarda, bu komşu router'lar, dinamik olarak, OSPF'in Hello protokolünü kullanarak tespit edilirler.

**Neighbors (komşular)** Her router diğerinin Hello paketlerini gördüğünde, EIGRP ve OSPF router'ları komşu olurlar.

**Neighborship (Komşuluk) tablosu** OSPF ve EIGRP routing protokolünde, her router, bitişik komşuları hakkında durum bilgilerini tutar. Yeni tespit edilen komşular öğrenildiğinde, komşunun interface ve adres bilgisi kaydedilir. Bu bilgi, komşu veri yapısında saklanır ve neighbor tablosu bu girdileri tutar. Neighborship tablosu, neighbor tablosu veya neighborship veritabanı olarak da belirtilebilir.

**NetBEUI** NetBIOS Extended User Interface: LAN Manager, Windows NT, LAN Server ve Windows for Workgroups içeren ağ işletim sistemlerinin bazılarında kullanılan NetBIOS'un gelişmiş bir versiyonudur, OSI LLC2 protokolü uygular. NetBEUI, NetBIOS'ta standartlaşmamış transport frame'lerine biçim verir ve daha fazla fonksiyon ekler. *Ayrıca bakınız: OSI.*

**NetBIOS** Network Basic Input/Output System: Düşük-seviye ağ işlemlerinden, oturum sonlandırma veya bilgi transferi gibi servisler istemek için bir IBM LAN'ında bulunan uygulamalar ile çalışan API.

**NetView** SNA (Systems Network Architecture) ağını görüntülemek için kullanılan IBM'den mainframe bir ağ ürünü. Bir VTAM (Virtual Telecommunications Access Method) uygulaması olarak çalışır.

**NetWare** Novell tarafından oluşturulan, yaygın kullanılan NOS. Uzak dosya erişimi ve dağıtık ağ servislerinin çoğunu sağlar.

**Networj katmanı** OSI referans modelinin, routing ve iki uç sistem arasında bağlantı ve yol seçimi yapılan 3.katmanıdır. Ayrıca bakınız: Application katmanı, Data Link katmanı, Physical katman, Presentation katmanı, Session katmanı ve Transport katmanı.

**Network Access katmanı** Paketlere ortam erişimi sağlayan, Internet protokol ailesinde alt katman.

**Network adresi** Bir ağ topluluğunda, ağ segmentini belirtmek için mantıksal network adresiyle kullanılır. Mantıksal adresler, doğası itibarıyla hiyerarşiktir ve ağ ve istemci olarak en az iki bölüme sahiptir. Hiyerarşik adrese bir örnek 172.16.10.5'dir. 172.16, network adresi, 10.5, istemci adresidir.

**Network control protokolü** Farklı network katmanı protokolü kurmanın ve yapılandırmanın bir yöntemi. NCP, çoklu Network katmanı protokolünün eşzamanlı kullanılmasına izin vermek için dizayn edilmiştir. Buradaki bazı protokoller, IPCP (Internet Protocol Control Protocol) ve IPXCP'dir (Internetwork Packet Exchange Control Protocol).

**NFS** Network File System: Bir ağ boyunca uzak dosya erişimine izin veren, Sun Microsystem'in yaygın olarak kullanılan dosya sistem protokol ailesindeki protokollerden birisidir. Bu isim, seyrek de olsa ayrıca RPC, XDR (External Data Representation) ve diğer protokolleride içeren tüm Sun protokol ailesini belirtmek için de kullanılır.

**NHRP** Next Hop Resulation Protocol: Bir NBMA (nonbroadcast multi-access) ağında, çeşitli kullanıcı ve router'ların MAC adreslerini dinamik olarak yerleştirmek için router'lar tarafından kullanılan protokol. Ara bir hop gerektirmeksizin, sistemleri direk bağlamayı mümkün kılar. Böylece ATM, Frame Relay, X.25 ve SMDS sistemlerinde performans artışını kolaylaştırır.

**NHS** Next Hop Server: NHRP protokolü tarafından tanımlanan bu sunucu, next-hop çözümleme önbellek tablolarını korur. NHS tarafından hizmet veren router'lar sayesinde ulaşılabilen düğümler ve ilgili düğümlerin IP'den ATM'e adres haritalarını listeler.

**NIC** Network interface card: Bilgisayara yerleştirilen bir elektronik devre. NIC, bir LAN'a ağ bağlantısı sağlar.

**Nibble** Dört bit.

**NLSP** NetWare Link Services Protocol: IS-IS tabanlı, novell'in link-state routing protokolü.

**NMP** NetWare Link Services Protocol: Switch'i kontrol etmek ve yönetmek için kullanılan bir Catalyst 5000 switch işlemci modülü.

**non-broadcast multi-access (NBMA) ağları** Non-broadcast multi-access (NBMA) ağları, Frame Relay, X.25 ve Asynchronous Transfer Mode (ATM) türüdür. Bu ağlar, çoklu-erişim için kabul edilirler, fakat Ethernet gibi broadcast kabiliyeti yoktur. Bu nedenle NBMA ağları tam olarak çalışmak için özel bir OSPF yapılndırmasına ihtiyaç duyar ve komşu ilişkisi tanımlanması gerekir.

**non-designated port** Bir switching kısır döngüsünün engellemek için frame'leri iletmeyen bir switch portudur. Spanning Tree Protocol (STP), bir portun atanmış (ileten) veya atanmamış (bloklayan) olduğuna karar vermekten sorumludur.

**non-stub area** OSPF'te, bir default route, alanlararası (interarea) route, alaniçi (intra-area) route, statik route ve harici route taşıyan kaynak-tüketici bir alan. Non-stub alanlar sadece üzerinde yapılandırılmış sanal linkler olabilen ve yalnızca bir autonomous system border router (ASBR) içeren alanlardır. *Mukayese ediniz. Stub area. Ayrıca bakınız: ASBR ve OSPF.*

**NRZ** Nonreturn to zero: Dijital veri aktarmak için çeşitli şifreleme tasarımlarından biri. NRZ sinyalleri, bir bit aralığı esnasında sinyal değişimi olmadan (sıfır-voltaj seviyesine dönmeden) değişmez voltaj değeri sağlar. Şayet aynı değerde (1 veya 0) bit serisi varsa, durum değişikliği olmayacaktır. Sinyal, kendi saat denetimine sahip değildir. *Ayrıca bakınız: NRZI.*

**NRZI** Nonreturn to zero inverted: Dijital veri aktarmak için çeşitli şifreleme tasarımlarından biri. Bir bit aralığının başındaki voltaj seviyesindeki (yüksekten düşüğe veya tersi) geçiş, 1 gibi bir değere çevrilmektedir. Geçişin yokluğu, 0 olarak çevrilmektedir. Böylece her değerdeki voltaj sürekli olarak tersine çevrilmektedir. NRZI sinyalleri, kendi saat denetimine sahip değildir. *Ayrıca bakınız. NRZ.*

**NT** Network termination: ISDN ağındaki bir nokta. *Bakınız: NT1 ve NT2.*

**NT1** NT1, iki telli "U" arayüzünü dört-telli "S/T" ye çeviren cihazdır.

**NT2** NT2, "S/T" bus'ı, elektriksel eşit iki farklı interface'e ayıran, PBX gib, ISDN-uyumlu switching cihazıdır. "S" interface'i TE1 cihazına bağlanırken, T interface'i NT1'e bağlanır,

**NVRAM** Nonvolatile RAM: Güç kapatılırken, içeriğini eksiksiz koruyan RAM'dir.

**OC** Optical Carrier: SONET optik sinyal aktarımları için, OC-1, OC-2, OC-3, vs. olarak atanan, fiziksel protokollerin bir serisi. OC sinyal seviyeleri, en düşüğü 51,84 Mbps (OC-1) olan çeşitli hızlardaki multimode bir fiber-optik hatta STS frame'leri yerleştirir. Her sonraki protokol, 51,84 ile bölünebilen bir hızda çalışır. *Ayrıca bakınız: SONET.*

**Oktet** Noktalı, ondalıklı bir IP adresinin bir bölümünü tanımlamak için kullanılan 8-tabanlı numaralama sistemi. Ayrıca, byte olarak da belirtilir.

**Ortam çevirisi** İki farklı LAN türünün iletişimine izin veren bir router özelliği. Örneğin, Ethernet'i Token Ring'e.

**OSI** Open Systems Interconnection: Farklı üretici ekipmanlarının birlikte çalışabilirliğini gerçekleştiren, veri ağı kurulumu standartlarının gelişimi için ISO ve ITU-T tarafından geliştirilen uluslararası standartlaştırma programı.

**OSI referans modeli** Open Systems Interconnection referans modeli: Cihazların herhangi bir kombinasyonunun, iletişim amacıyla nasıl bağlanabileceğini açıklayan, International Organization for Standardization (ISO) tarafından tanımlanan kavramsal bir modeldir. OSI modeli, görevleri, en üstte uygulamaların, en altta fiziksel ortamın bulunduğu bir hiyerarşi oluşturan yedi fonksiyonel katmana böler ve her bir katmanın sağladığı fonksiyonları tanımlar. *Ayrıca bakınız:Application katmanı, Data Link katmanı, Network katmanı, Physical katman, Presentation katmanı, Session katmanı ve Transport katmanı.*

**OSPF area** Bir OSPF area, bitişik ağ ve router'ların bir grubudur. Aynı area'daki tüm router'lar genel bir Area ID'sini paylaşır. Aynı anda bir router birden fazla area'ya üye olabildiğinden, Area ID'si router'daki belirli interface ile bağlantılıdır. Bu, bazı interface'ler area 0'a aitken, diğerlerinin area 1'e ait olmalarına izin verir. Aynı alandaki tüm router'lar, aynı topoloji tablosuna sahiptir.

**OSPF** Open Shortest Path : Çoklu-yol routing, yük dengeleme ve en düşük-cost routing özellikleri içeren IS-IS protokolünün önceki versiyonlarından türeyen, bir link-state, hiyerarşik routing algoritmasıdır. OSPF, internet ortamında RIP'in tavsiye edilen varisidir. *Ayrıca bakınız*: *Enhanced IGRP, IGP ve IP.*

**Oto duplex (Auto-duplex)** Bir switch veya hub portunun duplex'ini otomatik olarak ayarlayan katman1 ve katman2'deki bir ayarlama.

**Oto konfigürasyon** Düğümlerin otomatik olarak sistem kontrolü yaptığı, bir token ring arızalı domaininde düğümler tarafından çalıştırılan bir işlem. Problemli alanların yakınındaki ağı tekrar yapılandırmaya çalışır.

**Otomatik-algılama mekanizması (Auto-sense)** Ethernet switch, hub ve kartlarda hızı belirlemek amacıyla kullanılmaktadır.

**Otonom konfederasyonu** Diğer sistem ve gruplardan alınan bilgilerden ziyade kendi ağ erişilebilirlik ve routing bilgilerine bağlı olan, kendi kendini-yöneten sistemlerin bir kolleksiyonu.

**Otonom switching** Sistem işlemcisinden bağımsız olarak, paketleri anahtarlamada ciscoBus kullanarak paketlerin daha hızlı işlemden geçirilmeleri için Cisco router'ların kabiliyeti.

**Otonom system** Bakınız: AS.

**OUI** Organizationally unique identifier: Network Interface kartı yapan bir kuruluşa IEEE tarafından tahsis edilen koddur. Daha sonra kuruluş bu OUI'yı ürettiği tüm kartlara koyar. OUI, 3 byte (24 bit) uzunluğundadır. Sonra üretici, istemciyi eşsiz olarak tanımlamak için 3-byte tanıtıcı ekler. Adresin toplam uzunluğu 48 bittir (6 byte) ve donanım ya da MAC adresi olarak bilinir.

**Out-of-band sinyalleşmesi** Bir ağdaki, genellikle veri aktarımında kullanılandan ayrı frekans ve fiziksel kanalları kullanan herhangi bir aktarım.

**Out-of-band yönetimi**Ağın fiziksel kanallarının dışarıdan yönetmek. Örneğin, lokal LAN, WAN veya dial-in modeme doğrudan interface ile bağlı olmayan bir konsol bağlantısı kullanmak. *Mukayese ediniz: In-band yönetimi.*

**Outside network (Dış ağ)** NAT terminolojisinde, iç ağ (inside network), dönüştürmeye konu olan ağların grubudur. Dış ağ (outside network), genel olarak internette bulunan, diğer tüm adresleri belirtir.

**örnekleme değeri** Belirli bir zaman diliminde toplanan belirli bir dalgaboyu genliği örneğindeki değer.

**Paket switch** Bir iletişim kanalını çeşitli bağlantılara paylaştımayı mümkün kılan fiziksel bir cihaz. Onun fonksiyonu, paketler için en etkin aktarım yolunu bulmayı içerir.

**Paket switching** Paketlerle veri aktarım tabanlı bir ağ kurulumu teknolojisidir. Sürekli bir iletişimi ufak birimlere - paketlere - bölmek, bir ağdaki çoklu cihazlardaki veriyi, eşzamanlı aynı iletişim kanallarına paylaştırmayı mümkün kılar, fakat ayrıca tam routing bilgisinin kullanımı gerekmektedir.

**Paket** Veri iletişiminde, transfer edilen bilginin temel mantıksal birimidir. Bir paket, paketin nereden geldiği, nereye gittiği ve saire hakkında bilgiler içeren başlık ve/veya kuyrukta paketlenmiş ya da enkapsüle edilmiş, belirli sayıda veri byte'larından oluşmaktadır. Bir iletim göndermekle görevli çeşitli protokoller, daha sonra alıcı cihazlardaki ilgili protokollerin işledikleri, kendi katmanlarının başlık bilgilerini eklerler.

Paketleri diğer protokolün kısıtlamalarından binding yaparak kaçırmanın ve wrapper protokolünü destekleyen bir ağ üzerinde bu enkapsülasyona tabi paketi aktarmanın bir yöntemi.

**PAP** Password Authentication Protocol: Point-to-Point Protocol (PPP) ağlarında, istenen bağlantıları doğrulamanın bir yöntemi. Bağlantıya çalışıldığında, istemci (uzak) cihaz, lokal router'a bir şifre ve ID içeren kimlik denetimi isteği göndermelidir. Daha güvenli CHAP'ün (Challenge Handshake Authentication Protocol) tersine, PAP kriptolanmamış şifre yollar ve kullanıcının istenen kaynağa erişim izni olup olmadığını doğrulamaya çalışmaz. Sadece uzak ucu tespit eder. *Ayrıca bakınız: CHAP.*

**Parity kontrolü** Veri aktarımında bir hata kontrol yöntemi. Bir ekstra bit (parity biti), her bir karakter veya veri sözcüğüne eklenmektedir. Böylece, bitlerin ortalaması, bir tek sayı (tek parity'de) veya bir çift sayı (çift parity'de) olabilir.

**Pasif durum** Bir EIGRP routing tablosu hususunda, router bir route'un işlerliğini gerçekleştiremediğinde, route'un, pasif durumda olduğu kabul edilir.

**PAT** Port Address Translation: Bu işlem, tek bir IP adresini, kaynak TCP veya UDP port numaralarını değiştirerek çoklu kaynak olarak göstermeye izin verir.

**PCM** Pulse code modulation: Bir analog sinyali dijital bilgiye dönüştürme işlemi.

**PCR** Peak cell rate: ATM Forumunun tanımladığı gibi bir kaynağın aktarabildiği, saniyedeki cell maksimum hızının belirtildiği parametre.

**PDN** Public data network: Genellikle bir ücret karşılığında, bir PDN, özel şirketler ve hükümet acenteleri tarafından çalıştırılan bir bilgisayar iletişim ağına genel erişim önerir. Küçük organizasyonlar, uzak-mesafe ekipman ve devrelerine yatırım yapmaksızın WAN'lar oluşturmak için yardım eden PDN'nin fırsatlarından yaralanabilirler.

**PDU** Protocol Data Unit: OSI modelinin her bir katmanındaki işlemlerdir. Transport katmanınaki PDU'lar segment, Network katmanındaki PDU'lar paket veya datagram ve Data Link katmanındaki PDU'lar frame olarak tanımlanmaktadır. Physical katman bitleri kullanır.

**PGP** Pretty Good Privacy: Dosya ve mesajların korumalı transferini öneren popüler bir public-key/private-key şifreleme uygulaması.

**Phantom router** Kullanıcılara bir default gateway sağlamak için Hot Standby Routing Protocol (HSRP) ağında kullanılmaktadır.

**Physical katman** OSI referans katmanındaki en alttaki katman (katman 1). Veri frame'lerini, Data Link katmanından (katman 2) elektrik sinyaline çevirmekten sorumludur. Physical katman protokolleri ve standartları, örneğin, pin atamaları ile 0 ve 1 değerinde sinyalleşme için şifreleme planlaması içeren, kablo ve konnektörleri tanımlar. *Ayrıca bakınız:Application katmanı, Data Link katmanı, Network katmanı, Presentation katmanı, Session katmanı ve Transort katmanı.*

**PIM** Protocol Independent Multicast: IGMP isteklerini ve de multicast veri iletme isteklerini ele alan bir multicast protokolüdür.

**PIM-DM** Protocol Independent Multicast Dense Mode: PIM-DM, unicast route tablosu kullanır ve multicast veri aktarımı için kaynak root dağıtım mimarisine güvenmektedir.

**PIM-SM** Protocol Independent Multicast Sparse Mode: PIM-DM, unicast route tablosu kullanır ve multicast veri aktarımı için paylaşımlı root dağıtım mimarisine güvenmektedir.

**Ping** Packet Internet Groper: IP ağındaki belirli bir cihaza erişebilirliği test etmek için gönderilen bir mesajdan oluşan Unix-tabanlı bir internet kontrol aracı. Terimin kısa adı denizaltı radarına benzer çalışmayı işaret eder. Radar operatörünün bir sinyal gönderdiği ve bir sualtı nesnesinden eko (ping) işitmeyi bekledikleri gibi ağ kullanıcısı, ağdaki diğer düğümü pingleyebilir ve yanıt verip vermediğini görmek için bekler.

**Pinhole tıkanıklığı** Uzak bir ağa bilinen, farklı bant genişliğinde, birden fazla bağlantı olduğunda, distance-vector routing protokolleriyle ilgili bir problem.

**Plesiochronous** Senkron aktarımlardaki gibi sinyalde gömülü olması yerine saat denetiminin bir dış kaynaktan gelmesi dışında neredeyse senkron demektir.

**PLP** Packet Level Protocol: Bazen X.25 seviye3 veya X.25 Protokolü olarak da bilinen, X.25 ailesinin parçası, bir Network katmanı protokolüdür.

**PNNI** Private Network-Network Interface: Switch'ler veya switch grupları arasında, ağdaki yolların hesaplanması için kullanılan topoloji verisi öneren bir ATM Forumu düzenlemesi.

**Point-to-multipoint bağlantısı** ATM'de, sadece tek yönde giden bir iletişim. Root düğümü olarak bilinen, başlangıç noktasındaki tek bir sistemi, leaves olarak bilinen çoklu hedef noktalarındaki sistemlere bağlar. *Ayrıca bakınız: Point-to-point bağlantı.*

**Point-to-Point bağlantı** ATM'de, iki ATM uç sitemi arasında bir ya da iki yola yönlendirilebilen bir iletişim kanalı. Aynı zamanda bir point-to-point WAN seri bağlantısına işaret eder. *Ayrıca bakınız: Point-to-point bağlantısı.*

**Poison reverse güncellemeleri** Bu güncelleme, route zehirlenmesi olduktan sonra, bir router tarafından ilk oluşturana geri aktarılır (böylece split-horizon kuralı gözardı edilir). Tipik olarak, geniş routing kısır döngülerinin üstesinden gelmek ve bir subnet veya ağ erişilemez olduğunda (güncellemelere dahil etmeyerek ağa erişilemez olduğunu önermek yerine) kesin bilgi sunmak için DV routing protokolleri ile kullanılmaktadır. *Ayrıca bakınız: Route zehirlenmesi.*

**Polling** İkincil cihazların aktarım için veriye sahip olup olmadığını anlamak için birincil bir ağ mekanizması tarafından kullanılan düzenli sorgu işlemi. Her bir ikincile, ikincile aktarım hakkı veren bir mesaj gönderilir.

**POP** (1) Point of presence: Bir uzak-mesafe telefon firmasının, bir lokal telefon firmasıyla bağlanmak için ekipmanlarını yerleştirdiği fiziksel lokasyondur. (2) Post Office Protocol: Bir posta sunucusundan posta kurtarmak için kullanıcı e-posta uygulamaları tarafından kullanılan bir protokoldür.

**Port güvenliği** Bir takım güvenlik sağlamak için katman 2 switch'lerle kullanılır. Yönetmesi zor olduğundan, tipik olarak sık kullanılmaz. Sadece belirli frame'lere, yönetici tarafından atanmış segmentlerden geçme izni verir.

**Port numaraları** Kullanıcıdan-kullanıcıya sanal devrelerin izini tutmak için TCP ve UDP ile transport katmanında kullanılır.

**positive acknowledgment with retransmission** Belirli bir zaman frame'inde alıcı bilgisayar tarafından onaylanmayan verinin onaylanmasını ve tekrar aktarımını sağlayan bir bağlantı-tabanlı oturumdur.

**POTS** Plain old telephone service: Bu, son kuruluşlarda bulunan, geleneksel analog telefon servisini işaret eder.

**PPP** Point-to-Point Protocol: Önceki SLIP'in yerini alan, dial-up internet erişimi için kullanılan, en yaygın protokol. Özellikleri, adres bildirimini, CHAP ve PAP üzerinden kimlik denetimini, çoklu protokol desteğini ve link kontrolünü içerir. PPP iki katmana sahiptir: Link Control Protocol (LCP) bir link kurar, yapılandırır ve test eder ve sonra çeşitli Network Control Protocol'lerinden (NCP) biri için örneğin, IPX gibi belirli bir protokol ailesi için trafik taşır. *Ayrıca bakınız: CHAP, PAP ve SLIP.*

**Prefix (önek) routing'i** Kaç bitin subnet'te kullanıldığını ve bu bilginin, routing güncellemesinde nasıl gönderildiğini açıklayan yöntem. Örneğin, RIP versiyon 1, route güncellemelerinde subnet maskesi bilgisini göndermez. Bununla beraber RIP versiyon 2, gönderir. Bunun anlamı; RIPv2 güncellemeleri bir route güncellemesi ile /24, /25, /26 vs. gönderecek, RIP v1 göndermeyecektir.

**Presentation katmanı** OSI referans modelinin 6.katmanı, Application katmanındaki yazılım tarafından kullanılmak için verinin nasıl formatlanacağını, gösterileceğini, kodlanacağını ve dönüştürüleceğini açıklar. *Ayrıca bakınız:Application katmanı, Data Link katmanı, Network katmanı, Physical katman, Session katmanı ve Transort katmanı.*

**PRI** Primary Rate Interface: Tek bir 64 Kbps D kanalı, ilave olarak 23(T1) veya 30(E1) B bandından oluşan, bir PBX ve uzak-mesafe telefon firması arasındaki ISDN bağlantı çeşidi. *Ayrıca bakınız: ISDN.*

**Priority queuing** Bir interface çıkış kuyruğunda geçici yerleşen frame'lere, paket boyutları ve interface tipleri gibi özelliklere bağlı önceliklerin atandığı, bir routing fonksiyonu.

**Priviledge modu** Yapılandırmaların gözden geçirilmesi ve değiştirilmesini sağlayan Cisco router ve switch'lerde kullanılan komut-satırı EXEC modudur.

**Process/Application katmanı** Internet protokol ailesinde üst katman. Ağ servislerinden sorumludur.

**PROM** Programmable read-only memory: Özel ekipman kullanılarak, sadece bir defa programlanabilen ROM.

**Proses anahtarlaması** Bir paket, yönlendirilmek için bir router'a ulaştığında, router'ın işlemci arabelleğine kopyalanır ve router 3.katman adreslerinde bir arama işlemi gerçekleştirir. Route tablosu kullanarak, bir çıkış arayüzü, hedef adresiyle ilişkilendirilir. Router, hızlı-anahtarlama belleği başlatırken, işlemci, ilave yeni bilgiyle paketi çıkış arayüzüne gönderir. Aynı hedef adresi için tutulan sonraki paketler, ilk paket gibi aynı yolu takip eder.

**Protokol** Ağ kurulumunda, belirli bir iletişim türü için bir kurallar bütününün düzenlemesi. Terim aynı zamanda, bir protokol oluşturan yazılımı belirtmek içinde kullanılır.

**Protokol yığını** İlgili protokollerin bir kolleksiyonu.

**Protokol-bağımlı modüller** EIGRP routing protokollerinde kullanılan protokol-bağımlı modüller, network katmanı için IP, IPX ve AppleTalk için çoklu protokol desteğine izin veren protokol-özel gerekliliklerden sorumludur.

**Proxy Address Resolution Protocol** Proxy ARP: Bir kullanıcıda, yapılandırılmış default gateway hatasında yedeklemeye izin vermek için kullanılır. Proxy ARP, router gibi bir ara cihazın, bir uç düğüm yardımıyla bir ARP cevabını, istemci makinaya gönderdiği ARP protokolünün bir çeşididir.

**Pruning** Shortest-path tree'yi azaltma eylemi. Bu grup üyesi olmayan interface'leri çalışmaz duruma getirir.

**PSE** Packet switching exchange: Bir switch için X.25 terimi.

**PSN** Packet-switched network: Paket anahtarlama teknolojisi kullanan herhangi bir ağ. Aynı zamanda, packet-switched data network (PSDN) olarak da bilinir. *Ayrıca bakınız: Paket anahtarlama.*

**PSTN** Public switched telephone network: Halk arasında plain old telephone service (POTS) olarak bilinir. Telefon ağlarının çeşidi ve servislerin global uygunluğunu açıklayan bir terimdir.

**PVC** Permanent virtual circuit: Bir Frame Relay veya ATM ağında, kalıcı oluşturulan, yazılımda tanımlı bir mantıksal bağlantı. *Mukayese ediniz: SVC. Ayrıca bakınız: Sanal devre.*

**PVP** Permanent virtual path: PVC'lerden oluşan bir sanal yoldur. *Ayrıca bakınız: PVC.*

**PVP tunneling** Permanent virtual path tunneling: Public ağın transparan olarak, iki özel ağ arasında sanal yoldaki sanal kanalların tüm kolleksiyonlarını trunk yaptığı bir sanal yolu kullanarak iki ATM ağını birleştiren bir teknik.

**QoS** Quality of service: Herhangi bir verilen transmisyon sisteminin servis kullanılabilirliği ve transmisyon kalitesini ölçmek için kullanılan metriklerin ayarı.

**RADIUS** Remote Authentication Dial-In User Service: Uzak erişim cihazı ile bir kimlik doğrulama sunucusu arasında bağlantı kurmak için kullanılan bir protokol. Bazen, RADIUS çalışan bir kimlik doğrulama sunucusu, RADIUS sunucu olarak belirtilecektir.

**RAM** Random-access memory: Bilgilerini depolamak için tüm bilgisayarlar tarafından kullanılır. Cisco router'lar, donanım adres önbelleği ile birlikte paket tampon bellek ve routing tablolarını depolamak için RAM kullanırlar.

**RARP** Reverse Address Resolution Protocol: MAC adresini IP adresine eşleştiren, TCP/IP ailesindeki protokol. *Ayrıca bakınız: ARP.*

**RARP sunucusu** Bir Reverse Address Resolution Protocol sunucusu, bilinen bir MAC adresinden bir IP adresi sağlamak için kullanılmaktadır.

**Rate kuyruğu** Özel bir sanal devrenin, veriyi uzak uca aktaracağı belirtilen, bir veya daha fazla sanal devreye tanımlandığı değer. Her rate kuyruğu, bir ATM linkinde mümkün olan toplam bant genişliğinin bir segmentini belirtir. Tüm rate kuyruklarının ortalaması, toplam uygun bant genişliğini aşmaması gerekir.

**RCP** Remote Copy Protocol: Güvenli veri taşınmasını garantilemek için TCP kullanan, ağda uzak bir sunucuda bulunan bir dosya sistemine veya dosya sisteminden kopyalamak için kullanılan bir protokol.

**Referans modeli** Herhangi bir ağda çalışan uygulamaları oluşturmak için geliştiriciler tarafından kullanılır. En popular referans modeli, Open Systems Interconnection'dır (OSI).

**Referans noktası.** Bir NT1 ve S/T cihazı arasındaki bağlantıyı belirtmek için ISDN ağlarında kullanılır. S/T cihazı, dört-telli ağı, iki-telli ISDN standart ağına dönüştürür.

**Reliability (güvenirlik)** IGRP gibi, varsayılan olarak uzak bir ağa, en iyi yolu belirlemek için, sadece hattın bant genişliğini ve gecikmesini (delay) kullanır. Bununla birlikte EIGRP, uzak bir ağa en iyi yolu bulmak için arayışında bant genişliği, gecikme, yük (load) ve güvenirliğin bir kombinasyonunu kullanabilir. Reliability, tüm uzak ağlara linkin güvenirliğine işaret eder.

**Reliable Transport Protocol (RTP)** EIGRP paketlerinin, tüm komşulara taşınmasının garanti edilmesi ve tanzim edilmesinden sorumlu olan, EIGRP routing protokolünde kullanılan, güvenli taşıma protokolüdür.

**Reload** Cisco router'ların tekrar başlatılmasına sebep olan bir olay veya komut.

**RIF** Routing Information Field: Kaynak-route köprülemede, frame ya da token'ın gidiş yönünü belirten bir başlık alanı. Şayet, Route Information Indicator (RII) biti belirlenmediyse, RIF, hedeften kaynağa okunur, bu nedenle RIF, sağdan sola okunur. Yol bilgisini içeren, kaynağa yönlendirilmiş frame'ler için token ring frame başlığının bir parçası olarak tanımlanır.

**RIP** Routing Information Protocol: İnternette, en yaygın kullanılan interior gateway protokolü. RIP, routing metriği olarak hop sayısını kullanır. *Ayrıca bakınız: Enhanced IGRP, IGP, OSPF ve hop sayısı.*

**Ring** Mantıksal, dairesel bir topolojide bağlı iki veya daha fazla istasyondur. Bu topolojide, Token Ring, FDDI ve CDDI için temeldir, bilginin istasyondan istasyona sıra ile taşınmasıdır.

**Ring topoloji** Tek yönlü aktarım linklerini bağlayarak kapalı döngü oluşturan repeater serisinden oluşan bir ağ topolojisidir. Ağdaki tek istasyonlar, ağa bir repeater ile bağlanırlar. Fiziksel olarak, ring topolojileri, genel olarak, kapalı-döngü star yıldız düzeninde organize edilirler. *Mukayese ediniz: Bus topoloji ve star topoloji.*

**RJ konektör** Registered jack connector: Bakır kabloyu, network interface kartları, switch ve hub'lara bağlamak için sarmal-çift kablolamayla kullanılır.

**Rolled cable** Bir PC'nin COM portunu, bir router ve switch'in konsol portuna bağlamak için kullanılan kablo tipi.

**ROM** Read-only memory: Cihazı boot etmeye yardım etmek için bilgisayarlarda kullanılan çip. Cisco router'lar, power-on self-test (POST) çalıştırırlar ve sonra varsayılan olarak flash bellekteki IOS'u yüklerler. Bootstrap yüklemek için bir ROM çipi kullanılır.

**Route bridge** Ağ kısır döngülerini durdurmak için Spanning Tree Protokolü ile kullanılır. Root bridge, en düşük bridge ID'sine sahip olarak seçilmektedir. Bridge ID, priority (tüm bridge ve switch'lerde varsayılan 32,768dir) ve cihazın donanım adresi ile belirtilir.

**Route flap** Up/down biçiminde duyurulan bir router terimi.

**Route summarize** OSPF, EIGRP ve IS-IS gibi çeşitli routing protokollerinde, dağıtılan altağ adreslerinin birleştirilmesidir. Böylece, özet route, bir are border router'ı aracılığıyla diğer area'lara yayınlanır.

**Route zehirlenmesi** Geniş routing kısır döngülerinin üstesinden gelmek ve bir subnet veya ağ erişilemez olduğunda (güncellemelere dahil etmeyerek ağa erişilemez olduğunu önermek yerine) kesin bilgi sunmak için DV routing protokolleri ile kullanılmaktadır. Tipik olarak, bu, hop sayısını maksimumdan bir fazlaya ayarlayarak başarılmaktadır. *Ayrıca bakınız: Poison reverse güncellemeleri.*

**Router** Ağ trafiğinin aktarımında kullanmak için en iyi yola karar vermekte bir veya daha fazla metrik kullanan, donanımsal veya yazılımsal bir Network katmanı mekanizması. Router'lar tara-

fından ağlar arasında paketler göndermek, Network katmanında sağlanan bilgiye bağlıdır. Geçmişte bu cihaz bazen bir gateway olarak belirtilmiştir.

**Router ID (RID)** Router ID (RID), router'ı tespit etmek için kullanılan bir IP adresidir. Cisco, tüm yapılandırılmış loopback interface'lerinin en yüksek IP adresini kullanarak Router ID seçer. Şayet loopback interface yapılandırılmamışsa, OSPF, tüm aktif fiziksel interface'lerin en yüksek IP adreslisini seçecektir.

**Routing domaini** Aynı yönetimsel kuralların grubu altında çalışan uç ve ara sistemlerin bir koleksiyonu. Her routing domain'i ayrı ayrı verilen ve belirli bir area adresi olan bir veya daha fazla area içerir.

**Routing** Lokal altağlarından en son hedefe doğru mantıksal adreslenmiş paketlerin gönderilmesi işlemi. Geniş ağlarda, hedefine ulaşmadan önce, bir paketin seyahat edebileceği aralardaki çok sayıda hedef, routing'i çok karmaşık yapabilir.

**Routing metriği** Bir route'ın diğerlerinden öncelikli olup olmadığını belirtmek için routing algoritmaları tarafından kullanılan bir değer. Metrikler, bant genişliği, gecikme, hop sayısı, path cost, yük, MTU, reliability ve iletişim costu gibi bilgiler içerir. Link-state veya topolojik veritabanında diğer tüm bilgiler tutulurken, routing tablosunda, sadece en iyi route'lar saklanmaktadır. *Ayrıca bakınız: Cost.*

**Routing protokolü** Router'lar arasında routing tablolarının güncellenmesinde kullanılan algoritmaları tanımlayan protokol. Örnekler IGRP, RIP ve OSPF'tir.

**Routing tablosu** Router'da tutulan bir tablo veya belirli ağ hedeflerine, sadece en iyi route'ları ve bu route'larla ilgili metriklerin kayıtlarını tutan diğer ağlar arası iletişim mekanizması.

**RP** Route processor: Denetleyici işlemci olarak da bilinir; router'da kullanılan hafıza bileşenlerinin çoğunu, sistem yazılımı ve CPU'yu kontrol eden Cisco 7000 serisi router'larda kullanılan bir modüldür.

**RSP** Route/Switch Processor: Cisco 7500 serisi router'larda kullanılan RP ve SP'nin fonksiyonlarının birleştirildiği bir işlemci modülü. *Ayrıca bakınız: RP ve SP.*

**RTS** Request To Send: Bir iletişim hattında, veri aktarımı için bir EIA/TIA kontrol sinyal istek izni.

**S referans noktası** Dört-telli ISDN ağını, ağ sağlayıcısındaki ISDN switch'lerle iletişim için gerekli iki-telli ISDN ağına çevirmek için bir T referans noktasıyla çalışan ISDN referans noktası.

**Sabit yapılandırılmış router** Yeni herhangi bir arayüzle güncellenemeyen bir router.

**Sanal ring** Bir SRB ağında, hem lokal hem de uzak fiziksel ring'ler arasındaki mantıksal bir bağlantı.

**SAP** (1)Service Access Point: Bir adres düzenlemesinin parçası, IEEE 802.2 tarafından belirtilen bir alan. (2) Service Advertising Protocol: Router ve sunucuların kullandığı, ağdaki kaynak ve servislerin uygunluğunu, ağ kullanıcılarına bilgilendirmenin bir yolunu destekleyen Novell NetWare protokolü. *Ayrıca bakınız: IPX.*

**SCR** Sustainable cell rate: Trafik yönetimi için kullanılan bir ATM Forum parametresi. Aktarilabilen VBR bağlantıları için uzun-süreli ortalama hücre hızıdır.

**SDH** Synchronous Digital Hierarchy: Fiber Optics Transmission Systems (FOTS) için geliştirilen standartlardan biri.

**SDLC** Synchronous Data Link Control: SNA Data link katmanı iletişimlerinde kullanılan bir protokol. SDLC, HDLC'yi içeren birçok benzer protokol için temel olan bit-yönelimli, full-duplex seri protokolüdür.

**Seed (kaynak) router** Bir AppleTalk ağında, port açıklayıcısında ağ numarası ve kablo (ağ numarası) aralığıyla düzenlenmiş router. Kaynak router, bu ağ bölümündeki diğer router'lar için ağ numarası ve kablo aralığı belirtir ve bağlı AppleTalk ağındaki kaynak olmayan router'lardan gelen yapılandırma isteklerini cevaplar, yapılandırmalarını doğrulamak ve değiştirmek için bu router'lara izin verir. Her AppleTalk ağı, her ağ segmentine fiziksel bağlı, en az bir kaynak router'a ihtiyaç duyar.

**Senkron aktarım** Duyarlı saat denetimiyle dijital olarak aktarılan sinyaller. Bu sinyaller, aynı frekansa sahiptir ve her karakterin başı ve sonunda belirtilen kontrol bitlerinde (start/stop bitleri olarak belirtilir) enkapsülasyon için özel karakterler içerir. *Ayrıca bakınız: Asenkron aktarım ve senkron aktarım.*

**Sequencing** Sanal devrelerde ve segment'leri numaralamak için bölümlenmesinde kullanılır, böylece doğru sırayla tekrar geri konulabilirler.

**Seri aktarım** Tek bir kanal boyunca, bir defada tek bit'in olduğu, WAN seri konnektörleri, seri aktarım kullanır.

**Session katmanı** OSI referans modelinin 5.katmanıdır, Session katmanının görevleri arasında veri değişimini denetlemek ve uygulamalar arasındaki oturumları oluşturmak, yönetmek ve kapatmak sayılabilir. *Ayrıca bakınız: Application katmanı, Data Link katmanı, Network katmanı, Physical katman, Presentation katmanı ve Transort katmanı.*

**Set-tabanlı** Set-tabanlı router ve switch'ler cihazları yapılandırmak için "set" komutu kullanır. Cisco, set-tabanlı komutlardan uzaklaşıyor ve tüm yeni cihazlarında command-line interface (CLI) kullanmaktadır.

**Setup modu** Router boot edildiğinde, NVRAM'de yapılandırma bulunmazsa, router'ın girdiği moddur. Yöneticilere, bir router'ı adım adım yapılandırmaya izin verir. Komut-satırı interface'i kadar güçlü ve esnek değildir.

**SF** Bir super frame (ayrıca D4 frame olarak da bilinir), her biri 192 bit ve hata kontrolü içeren diğer fonksiyonları sağlayan 193.bit ile 12 frame'den oluşur. SF, sık sık T1 devrelerinde kullanılır. 24 bit kullanan teknolojinin en son versiyonu, Extended Super Frame'dir (ESF). *Ayrıca bakınız: ESF*

**shared tree** Multicast veri göndermenin bir yöntemi. Shared tree'ler, çoklu kaynakların genel bir kesişme noktasını paylaştığı bir mimaride kullanılır.

**Shortest Path First (SPF)** Bir çeşit routing algoritması. Tek gerçek SPF protokolü, Open Shortest Path First'dür (OSPF).

**Sıkıştırma** Tek imli verinin tekrarlayan dizgisinin gösterilmesiyle, bir hat boyunca normalde kabul edilebilenden daha fazla veri göndermek için kullanılan bir teknik.

**Silikon switching** Ayrı bir işlemci (Silicon Switch Processor veya SSP) kullanmaya dayalı, Cisco 7000 serisi router'larda kullanılan bir yüksek-hızda switching çeşidi.

**Simplex** Veri ya da dijital bir sinyalin aktarıldığı mod. Simplex, sadece tek yönde aktarımın bir yoludur. Half duplex, bir defada sadece tek yön olarak iki yönde aktarır. Full duplex, eşzamanlı iki yönde aktarır.

**Sinyalleşme paketi** Diğer benzer mekanizmalarla bağlantı kurmak isteyen, bir ATM-bağlı mekanizma tarafından oluşturulan bilgilendirme paketi. Paket, bağlantılar ve uç noktaların ATM NSAP adresleri için gerekli Qos parametrelerini içerir. Şayet, istenilen QoS desteklenebiliyorsa, uç nokta, bir kabul mesajı ile yanıt verir ve bağlantı kurulur. *Ayrıca bakınız: QoS.*

**SLIP** Serial Line Internet Protocol: Sadece TCP/IP'yi destekleyen, point-to-point bağlantılar için bir endüstri satandardı seri enkapsülasyonudur. SLIP, PPP'nin atasıdır. *Ayrıca bakınız: PPP.*

**sliding window (kayan pencere)** Hem TCP hem de çeşitli Data Link katman protokolleri tarafından kullanılan akış kontrolü yöntemi. Bu yöntem, alınan uygulamalar ve ağ veri akışı arasındaki bir arabelleğe yerleşir. Orası için "window" uygundur. Bu pencere boyutu, uygulama veriyi oradan okudukça büyür ve yeni veri gönderildiğinde küçülür. Alıcı, vericiye kesin pencere boyutunu duyurur ve pencere kesin bir eşiğin üzerinde artıncaya kadar veri kabul etmeyi durdurabilir.

**SMDS** Switched Multimegabit Data Service: Yüksek-hız sağlayan telefon firmaları tarafından tavsiye edilen, packet-switching, datagram-tabanlı bir WAN ağ kurulum teknolojisidir.

**SMTP** Simple Mail Transfer Protocol: Elektronik posta servisi sağlamak için Internette kullanılan bir protokol.

**SNA** System Network Architecture: OSI referans modeline benzer, karmaşık, zengin özellikli bir ağ mimarisidir. Çeşitli varyasyonları vardır; 1970'lerde IBM tarafından oluşturuldu ve yedi katmandan meydana gelmektedir.

**SNAP** Subnetwork Access Protocol: SNAP, Ethernet, Token Ring ve FDDI LAN'larında kullanılan bir frame'dir. Veri aktarımı, bağlantı yönetimi ve QoS seçimi SNAP tarafından çalıştırılan üç ana fonksiyondur.

**snapshot routing** Snapshot routing, dinamik routing tablosunun o anki görüntüsünü alır ve uzak bağlantı koptuğu zaman bile onu devam ettirir. Bu, linkin aktif kalması gerekliliği olmaksızın, dinamik bir routing protokolünün kullanılmasına izin verir, dakika başına kullanım ücreti kullanılabilir.

**SNMP** Simple Network Management Protocol: Bu protokol, istatistiksel ve çevresel veri için SNMP agent ve cihazlarını sorgular. Bu veri cihaz sıcaklığı, ismi, performans istatistikleri ve daha fazla bilgi içerebilir. SNMP, SNMP agentlarda olan MIB objeleri ile çalışır. Bu bilgi sorgulanır, daha sonar SNMP sunucusuna gönderilir.

**SOHO** Small office/home office: Uzak kullanıcılar için güncel bir terimdir.

**Soket** (1) İletişimler için bir hedef noktası gibi bir ağ cihazında çalışan yazılım yapısıdır. (2) AppleTalk ağlarında, bir düğümdeki belirli lokasyondaki bir varlık; AppleTalk soketleri, kavramsal olarak TCP/IP portlarına benzer.

**SONET** Synchronous Optical Network: Bell laboratuvarlarında geliştirilen, fiber-optik ortamda senkron aktarım için bir ANSI standartı. 51.84Mbps bir temel sinyal hızını ve bu hızın, Optical Carrier olarak da bilinen çoklu bir ayarını belirtir. (2.5 Gbps a kadardır.)

**Source tree** Bir multicast veri gönderme yöntemi. Source tree'ler, bir tree gibi, multicast trafiğinin kaynak mimarisini kullanır.

**SP** Switch processor: CiscoBus controller olarak da bilinir. Tüm CxBus aktiviteleri için agent yönetir gibi davranan bir Cisco 7000 serisi işlemci modülüdür.

**Span** İki tesisi bağlayan bir full-duplex dijital aktarım hattı.

**SPAN** Switched Port Analyzer: Ortamdaki mevcut ağ analizörlerinin görüntüleme kabiliyetlerini geliştirerek, anahtarlamalı Ethernet ortamında idare etmek için serbestlik öneren, bir Catalyst 5000 switch özelliği. Bir anahtarlamalı segmentte, SPAN portuna bağlı bir analizörü, diğer Catalyst anahtarlamalı porttan trafik görüntüleyebilirken, SPAN, trafiği önceden saptanan SPAN portuna mirror uygular.

**Spanning explorer paketi** Bazen, limitli-route veya tek-route explorer paketi olarak da bilinir. Bir kaynak-route köprüleme ağında, yollar araştırıldığında, istatistiksel yapılandırılmış bir spanning tree izler.

**Spanning tree** Kısır döngünün olmadığı bir ağ topolojisinin altkümesi. Bridge'ler bir kısır döngüye bağlandıklarında ya da switch, önceden gönderilen bir frame'i tespit edemediğinde, interface'in

defalarca geçireceği bir frame'i silmenin bir mekanizması yoktur. Bu frame'i silmenin bir yöntemi olmaksızın bridge'ler sürekli bunları gönderir, bant genişliğini tüketir ve ağa ek yük getirir. Spanning tree, her paket için sadece bir yol sağlamak için ağı budar. *Ayrıca bakınız: Spanning Tree Protokolü ve spanning-tree algoritması.*

**Spanning Tree Protokolü (STP)** Spanning tree algoritması kullanarak bir spanning tree oluşturarak ağ topolojisinde kısır döngülerden dinamik olarak kaçınmak için bir learning bridge'i mümkün kılan, bridge protokolü (IEEE 802.1D). Bridge Protocol Data Units (BPDU) olarak bilinen spanning tree frame'leri düzenli aralıklarla ağda tüm cihazlar tarafından gönderilir ve alınır. Spanning Tree'nin parçası switch'ler, frame'leri göndermezler; yerine, spanning tree topolojisini belirlemek için işlemden geçerler. Cisco Cataylist serisi switch'ler, bu özelliği çalıştırmak için STP 802.1D kullanırlar. *Ayrıca bakınız: BPDU, learning bridge, MAC adresi, spanning tree ve spanning tree algoritması.*

**Spanning-tree algoritması (STA)** Spanning-tree protokolü (STP) kullanarak bir spanning tree oluşturan algoritmadır. *Ayrıca bakınız: Spanning-tree ve Spanning Tree Protokolü.*

**SPF** Shortest Path First algorithm: En kısa yola karar vermek için kullanılan bir routing algoritmasıdır. Bazen, Dijkstra'nın algoritması olarak bilinir ve düzenli olarak link-state routing algoritmasında kullanılır. *Ayrıca bakınız: Link-state routing algoritması.*

**SPID** Service Profile Identifier: Servis sağlayıcılar veya lokal telefon firmaları tarafından atanan ve yöneticiler tarafından BRI portlarında yapılandırılan bir numara. SPID'ler, ISDN yardımıyla bağlı bir cihazın abonelik servislerini belirtmek için kullanılır. ISDN cihazları, bir servis sağlayıcıya bağlantı başlatan telefon firmasına erişildiğinde, SPID kullanır.

**Split horizon** Routing kısır döngülerini engellemek için kullanışlı, router'lar hakkındaki bilgilerin, bilginin alındığı interface'den gönderilmesinin engellendiği distance-vector routing kuralıdır.

**Spoofing** (1) Dial-on-demand routing'de (DDR) gönderilecek bir trafik olmadığında, kullanım bedeli kaydetmesi durdurulan bir devre-anahtarlamalı linkte, spoofing, bir kullanıcının, çalışıyor ve bir oturumu desteklercesine bir interface gibi davranmasına sebep olan, router'lar tarafından kullanılan bir hiledir. *Ayrıca bakınız: DDR.* (2) Filtre ve access list'ler gibi ağ güvenlik mekanizmalarını aldatmak için, yanlış bir adresle etiketli bir paketi gönderme eylemi.

**Spooler** Bir kuyruktan sıralı bir biçimde çalışma için onaylı isteklerin işlemden geçtiği bir yönetim uygulaması. Print spooler iyi bir örmektir.

**SPX** Sequenced Packet Exchange: Network katmanı (3.katman) tarafından sağlanan datagram servislerini arttıran bir Novell NetWare transport protokolüdür. XNS protokol ailesinin Switch-to-Switch Protokolünden türemiştir.

**SQE** Signal Quality Error: Bir Ethernet ağında, Bir alıcı/vericiden, collision-detection devresi çalışan, bağlı bir makinaya gönderilen mesaj.

**SR/TLB** Source-Route Translational Bridging: İki bridge protokolünü dönüştüren ara bir bridge tarafından yardım edilen transparan bridge istasyonlarıyla iletişim için kaynak-route istasyonlarına izin veren bir köprüleme yöntemi. Token Ring ve Ethernet arasındaki köprülemede kullanılır. *Mukayese ediniz: SRT.*

**SRB** Source-Route Bridging: IBM tarafından oluşturulan, Token Ring ağlarında kullanılan bridging yöntemidir. Kaynak, veri göndermeden önce bir hedefe tüm route'ları belirler ve her paketin routing information fields'deki (RIF) bilgiyi içerir. *Tersidir: transparent.*

**SRT** Source-Route Transparent bridging: Kaynak-route ve transparan bridging'i birleştiren, IBM tarafından geliştirilen bir bridging düzenlemesi. SRT, bir cihazda her iki teknolojiden faydalanır, tüm uç düğümlerin ihtiyaçlarını yerine getirir. Bridging protokolleri arasında dönüşüme gerek yoktur. *Mukayese ediniz: SR/TLB.*

**SSAP** Source Service Access Point: Network katman protokolünü tespit eden paketin kaynak alanında tanımlanan ağ düğümünün SAP'si. *Ayrıca bakınız: DSAP ve SAP.*

**SSE** Silicon Switching Engine: Silicon Switch Processor (SSP) içine güçlü şifreli, Cisco'nun silikon anahtarlama teknolojisinin yazılım bileşeni. Silikon anahtarlama, sadece bir SSP ile Cisco 7000 de mümkündür. Silikon-anahtarlamalı paketler, SSE'deki silikon-anahtarlama belleği ile kıyaslanabilir. SSP, route işlemcisinden anahtarlama işlemi yükleyen tahsis edilmiş bir switch işlemcisidir. Bir hızlı-anahtarlama işlemi sağlar, fakat paketler, SSH'yi almak için hala router'ın omurgasından geçer ve sonra çıkış interface'ine döner.

**Standart access list** Bir ağı filtrelemek için sadece kaynak IP adresleri kullanan IP access listi.

**Standart IPX access list** Bir ağı filtrelemek için sadece kaynak ve hedef IPX adresi kullanan IPX access listi.

**Star topolojisi** Point-to-point linkler kullanan, ortak merkezi bir cihazda (bir hub olarak bilinen) olan ağdaki uç noktalarla bir LAN fiziksel topolojisi. Mantıksal ring toplojisi, point-to-point linklerden ziyade tekyönlü bir kapalı-döngü star kullanan fiziksel bir star topoloji olarak yapılandırılabilir. Bu, hub'taki bağlantıların, dahili bir ring'de düzenlenmesidir. *Ayrıca bakınız: Bus topoloji ve ring topoloji.*

**Startup aralığı** Şayet bir AppleTalk düğümü, son boot edildiğinde kaydedilmiş bir numaraya sahip değise, 65,280'den 65,5342e kadar olan aralıktan seçer.

**Statik route** Bir yönetici tarafından routing tablosuna amaca dönük girilen route bilgisidir ve dinamik routing protokolleri tarafından seçilen route'lar üzerinde öncelik alır.

**Statik VLAN** Port,port elle yapılandırılan bir VLAN. Bu genel olarak ağlarda kullanılan yöntemdir.

**STM-1** Synchronous Transport Module Level 1: Avrupa SDH standartında, ATM cell'leri taşımak için kullanılan 155.52Mbps hızında olacak frame yapısını belirleyen formatlardan birisi.

**store-and-forward paket switching** Switch'in ilk olarak her paketi arabelleğe kopyaladığı ve bir cyclic redundancy check (CRC) çalıştırdığı bir teknik. Şayet paket hatasızsa, o zaman switch filtreleme tablosundaki hedef adresine bakar, uygun çıkış portu belirler ve paketi gönderir.

**STP** (1) Shilded twisted-pair: EMI'yi düşürmek için korumalı izolasyon katmanına sahip, birçok ağ uygulamasında kullanılan kablolama düzenlemesi. (2) Spaning Tree Protocol.

**straight-through (düz) kablo** Bir kullanıcıyı switch'e, kullanıcıyı hub'a veya router'ı switch veya hub'a bağlayan Ethernet kablo türü.

**Stub area** Bir default route, intra-area route ve inter-area route taşıyan OSPF area'sıdır, harici route taşımaz. Sanal linklerin yapılandırması, bir stub area üzerinde yapılamaz ve stub area'ların bir ASBR içermesine izin verilmez. *Ayrıca bakınız: Non-stub area, ASBR ve OSPF.*

**Stub network** Router'a sadece bir bağlantıyla bağlı ağ.

**STUN** Serial Tunnel: Bir HDLC linkini bir SDLC linkine bağlamak için kullanılan bir teknoloji.

**Subarea** Bir subarea düğümü, onun bağlı linkler ve kenardaki düğümlerden oluşan SNA ağının bir bölümü.

**Subarea düğümü** Bir SNA iletişiminde, tüm ağ adreslerini kullanan bir kullanıcı veya denetleyici.

**Subchannel (altkanal)** Ayrı bir broadband iletişim kanalı oluşturan frekans-tabanlı altbölüm.

**Subinterface** Tek bir fiziksel interface'de kullanılabilir birçok sanal interface.

**Subnet adresi** Alt ağ gibi, özellikle subnet maskesi ile tanımlanan bir IP adresinin bölümü. *Ayrıca bakınız: IP adresi, altağı, subnet maskesi.*

**Subnet** Bakınız: Subnetwork.

**Subnet maskesi** Basitçe maske olarak da bilinen, subnet adresleri için kullanılan bir IP adresinin bitlerini tanımlamak için kullanılan 32-bit bir adres maskesi. Bir maske kullanılmasıyla, router'ın tüm 32 bit'i dikkate almasına gerek kalmaz, sadece maske tarafından gösterilenler kullanılır. *Ayrıca bakınız: Adres maskesi ve IP adresi.*

**Subnetleme**

**Subnetwork (Altağ)** (1) Geniş bir IP ağının parçası olan bir ağdır ve bir subnet adresi tarafından tanımlanmaktadır. Bir ağ yöneticisi, hiyerarşik, çok seviyeli bir routing yapısı sağlamak ve aynı zamanda bağlı ağların adresleme karmaşıklığından alt ağı korumak için bir ağı segmentlerine ayırır. Ayrıca bir subnet olarak da bilinir. *Ayrıca bakınız: IP adresi, subnet maskesi ve subnet adresi.* (2) OSI ağlarında, özellikle, sadece bir yönetimsel domain tarafından kontrol edilen ES'ler ve IS'lerin bir kolleksiyonunu işaret eden terimdir. Tek bir ağ bağlantı protokolü kullanır.

**Summarization (özetleme)** Çoklu routing tablosu girişlerini tek girişe özetlenmesi işlemi için kullanılan terimdir.

**Sunucu** Kullanıcılara ağ servisleri sağlayan donanım ve yazılımdır.

**Supernetting** Bakınız. Summarization.

**SVC** Switched virtual circuit: İstendiğinde oluşturulan, aktarım biter bitmez durdurulan ve devreye artık ihtiyaç duyulmayan, dinamik olarak kurulan bir sanal devredir. ATM terminolojisinde, bir anahtarlamalı sanal bağlantıya işaret eder.

**Switch** (1) Ağ kurulumunda, frame filtreleme, yayma ve gönderme gibi çoklu fonksiyonlardan sorumlu bir cihazdır. Özel frame'lerin hedefini kullanarak çalışır. Switch'ler OSI modelinin Data Link katmanında çalışır. (2) Yaygın olarak, ihtiyaç duyulduğunda kurulan ve artık ihtiyaç olmadığında sonlandırılan bağlantılara izin verilen elektronik/mekanik bir cihaz.

**Switch bloğu** 2.katman ve 3.katman switch'lerin bir kombinasyonu. 2.katman switch'ler, kablolama odasındaki kullanıcı uçlarını access katmanına bağlar ve 10 veya 100Mbps tanımlı bağlantılar sağlar. 1900/2820 ve 2900 Catalyst switch'ler, switch bloğunda kullanılmaktadır.

**Switch fabric** Birçok switch ile 2.katman anahtarlamalı ağ topluluğunu tanımlamak için kullanılan terimdir.

**Syslog** Bir uzak cihaz tarafından sistem log mesajlarını görüntülemek için kullanılan bir protokol.

**Şifreleme** Yetkisiz girişleri engellemek için bilginin farklılaştırılmış bir forma dönüştürülerek etkili bir şekilde saklanması. Şifreleme işlemi için algoritmalar kullanılır. Deşifre olarak bilinen bir işlem ise algoritmanın ters yönde işletilmesiyle şifrelemenin çözümlenmesidir.

**T referans noktası** Bir 4-telli ISDN ağını, iki-telli ISDN ağına dönüştürmek için bir S referans noktası ile kullanılır.

**T1** 1.536 Mbps bant genişliği oluşturmak için her bir 64Kbps 24 DS0'lar kullanan dijital WAN, eksi saat denetimi ek yükü, 1.544Mbps kullanılabilir bant genişliği sağlar.

**T3** 44.763Mbps bant genişliği sağlayabilen dijital WAN.

**TACACS+** Terminal Access Controller Access Control System Plus: TACACS'ın gelişmiş bir versiyonudur. Bu protokol RADIUS'a benzer. *Ayrıca bakınız: RADIUS.*

**TCP** Transmission Control Protocol: OSI referans modelinin transport katmanında tanımlı bir bağlantı-tabanlı protokol. Güvenli veri taşıma sağlar.

**TCP/IP** Transmission Control Protocol/Internet Protocol: İnternetin temelini oluşturan protokol ailesi. TCP ve IP bu ailede en yaygın bilinen protokollerdir. *Ayrıca bakınız: IP and TCP.*

**TDM** Time Division Multiplexing: Çeşitli kanallardan veriye, öntanımlı time slotlarına bağlı, tek telde bant genişliği tahsis etmek için bir teknik. Bant genişliği, istasyonun veri göndermek için bir istasyonun amacı ne olursa olsun her kanala tahsis edilir. *Ayrıca bakınız: ATDM, FDM ve multiplexing.*

**TE** Terminal equipment: Bir telefon veya bilgisayar gibi, ağda bağlı ve ISDN-uyumlu ikincil cihazı. TE1'ler, ISDN'e hazır ve ISDN sinyalleşme tekniklerinden anlayan cihazlardır. TE2'ler ISDN'e hazır olmayan ve ISDN sinyalleşme tekniklerinden anlamayan tekniktir. TE2 ile bir terminal adaptör kullanılmak zorundadır.

**TE1** Terminal Equipment Type 1: Dört-telli bir cihaz. Sarmal-ikili dijital interface'leri terminal equipment type 1 olarak belirtilir. En modern ISDN cihazları bu tip olanlardır.

**TE2** Terminal Equipment Type 2: Terminal equipment type 2 olarak bilinen cihazlar, ISDN sinyalleşme tekniklerinden anlamazlar ve bir terminal adaptör, sinyalleşmeyi çevirmek için kullanılmak zorundadır.

**Telco** Telefon Company için yaygın bir kısaltma.

**Telnet** TCP/IP protokol ailesinde standart terminal emülasyon protokolü. Uzak terminal bağlantı yöntemi, kullanıcıların uzak ağlarda login olmalarını mümkün kılar ve lokal olarak bağlanır gibi, bu kaynakları kullanırlar. Telnet, RFC 854 de tanımlanmıştır.

**Temel Yönetim Ayarlaması** Setup modundayken Cisco router'larla kullanılır. Sadece, birinin router'a telnet yapıp onu yapılandıracağı şekilde bir router'ı çalıştırmak için yeterli yönetim ve yapılandırma sağlar.

**Terminal adapter (TA)** Doğal ISDN interface'i olmayan bilgisayar ile bir ISDN hattı arasındaki donanımsal interface. Gerçekte, standart asenkron interface'i doğal-olmayan ISDN cihazına bağlamak için bir cihazdır.

**Terminal emülasyonu** Belirli bir anabilgisayara direkt bağlı "dumb" bir terminal gibi çalışması için PC'ye izin veren bir PC veya LAN'da kurulu olan yazılımın kullanımı.

**TFTP kullanıcı/sunucu** Trivial File Transfer Protokolünün, Network katmanında IP, Transport katmanında UDP (onu güvensiz yapar) kullanarak dosyaları göndermek için kullanıldığı bir host ya da sunucudur.

**TFTP** Trivial File Transfer Protocol: Kavramsal olarak, FTP'nin kısıtlı versiyonudur. Ne istediğinizi ve nerede bulunacağını tam olarak bilip bilmeme seçeneğinin protokolüdür. TFTP, FTP'nin sahip olduğu fonksiyon zenginliğini içermez. Özellikle, directory arama özelliği yoktur, dosyaları alıp göndermek dışında hiçbir şey yapamaz.

**Thicknet** Ayrıca 10Base5 olarak da bilinir. Bus tipi ağlar, kalın bir koaksiyel kablo kullanır ve 500 metreye kadar Ethernet çalışırlar.

**Thinnet** Ayrıca 10Base2 olarak da bilinir. Bus tipi ağlar, ince bir koaksiyel kablo kullanır ve 185 metreye kadar Ethernet çalışırlar.

**three-way handshake** Bir sanal devrenin nasıl kurulduğunu tanımlamak için bir TCP oturumunda kullanılan terim. Üç veri segmenti kullanıldığından, "üç-yönlü" anlaşma olarak isimlendirilmektedir.

**Token bus** IEEE 802.4 düzenlemesi için temel olan ve bir bus topolojisi üzerinde token-passing erişimi çalıştıran LAN mimarisi. *Ayrıca bakınız: IEEE.*

**Token passing** Bir token olarak bilinen küçük bir frame'in sahipliği tabanlı sistematik bir yolda fiziksel ortam araçlarına erişmek için ağ cihazları tarafından kullanılan bir yöntem.

**Token Ring** IBM'in token-passing LAN teknolojisi. Bir ring topolojisinde 4Mbps veya 16Mbps'da çalışır. IEEE 802.5 ile tanımlanmıştır. *Ayrıca bakınız: Ring topoloji ve token passing.*

**Token** Sadece kontrol bilgisi içeren bir frame. Bu kontrol bilgisine sahip olmak, ağ üzerinde veri aktarmak için bir ağ cihazına izin verir.

**Toll network** Paketleri göndermek için the public switched telephone network (PSTN) kullanan WAN ağı.

**Toplam interface gecikmesi** Hattın gecikmesi ile ilgili bir Cisco terimidir. IGRP ve EIGRP'deki birleşik metrik, varsayılan, hattın bant genişliği ve gecikmesi kullanılarak hesaplanmaktadır.

**Topoloji veritabanı** Bir topoloji veritabanı (topoloji tablosu olarak da tanımlanır), neighbor router'lar tarafından yayınlanan tüm hedefleri içerir. Her giriş, hedef adresi ile ilgilidir ve hedefleri yayınlayan bir neighbor listesidir.

**Traceroute** Traceroute, bir ağ topluluğu boyunca yollanan bir paketin yolunu izlemek için kullanılan IP komutudur.

**Transparent bridging** Ethernet ve IEEE 802.3 ağlarında kullanılan bridging düzenlemesi. Bir defada bir hop boyunca frame'leri geçirir. Uç-düğüm MAC adreslerini, bridge portları ile ilişkilendiren tablolarda saklanan bridging bilgisini kullanır. Bu tür bridging transparan sayılır, çünkü kaynak düğümü bridging yapıldığını bilmez. Hedef frame'leri direkt olarak uç düğümlere gönderilir.

**Transport Katmanı** Ağdaki, uç düğümler arasında güvenli iletişim için kullanılan, OSI referans modelinin 4.katmanı. Transport katmanı, sanal devreleri kurmak, devam ettirmek ve sonlandırmak görevlerini yerine getirir. Ayrıca transport hata tespiti ve çözümü ile bilgi akış kontrolü için kullanılan mekanizmalar içerir. *Data link Katmanı, Network katmanı, Physical katman, Presentation katmanı, Session katmanı ve Application katmanı.*

**Trap** SNMP mesajlarını, SNMP yöneticilerine göndermek için kullanılmaktadır.

**TRIP** Token Ring Interface Processor: Cisco 7000 serisi router'larda kullanılan, yüksek-hızlı interface işlemcisi. TRIP, her birine bağımsız ayarlı 4Mbps veya 16Mbps hızda portlarla IBM ve IEEE 802.5 ile bağlantı için iki veya dört port sağlar.

**Trunk link** Switch'ler arasında ve bazı sunuculardan switch'lere kullanılan linkler.

**TTL** Time to live: Bir paketin geçerli olduğu zamanı belirten, IP başlığındaki bir alan.

**TUD** Trunk Up-Down: Trunk'ları görüntülemek için ATM ağlarında kullanılan bir protokol. Bir trunk, trunk hattı kalitesinden emin olmak için ATM switch'ler tarafından gönderilen belirli bir sayıda test mesajını kaybettiğinde, TUD, trunk'ın down olduğunu duyurur. Bir trunk, durumunu değiştirir ve tekrar up olduğu zaman, TUD, trunkın up olduğunu farkeder ve trunk hizmet verir hale gelir.

**Tünelleme**

**U referans noktası** Bir TE1 ve ISDN ağı arasındaki referans noktası. U referans noktası, ISDN sinyalleşme tekniklerinden anlar ve 2-telli bağlantı kullanır.

**Uçnoktalar** Bakınınz: BGP neighbors.

**Uçtan-uca VLAN ler** Uçtan uca switch yapısı boyunca yayılan VLAN'ler; uçtan uca VLAN'lerdeki tüm switch'ler, yapılandırılmış tüm VLAN'lerden anlarlar. Uçtan-uca VLAN'ler, fonksiyon, proje, bölüm ve bu gibi özelliklerine bağlı üyelikleri kabul etmek için yapılandırılırlar.

**UDP** User Datagram Protocol: Basit olarak, onay veya taşıma garantisi olmaksızın datagramların değiş tokkuş edilmesine izin veren, TCP/IP protokol yığınındaki, bir connectionless transport katman protokolüdür,. UDP, RFC 768'de tanımlıdır.

**Unicast** Direk kullanıcıdan-kullanıcıya iletişim için kullanılır. İletişim, sadece bir hedefe yönlendirilir ve bir kaynaktan çıkartılır.

**unidirectional shared tree** Shared tree multicast routing'in bir yöntemi. Bu yöntem, sadece, RP'den, gönderilecek verinin multicast edilmesine izin vermektedir.

**unnumbered frames** Link başlatma ve kapatma veya mod belirtme gibi kontrol-yönetim amaçları için kullanılan HDLC frame'leri.

**User (kullanıcı) modu** Az sayıda komutla çalışmak için bir yöneticiye izin veren, Cisco IOS EXEC modu. Kullanıcı modunda, sadece istatistikleri doğrulayabilirsiniz, router ya da switch yapılandırmalarını göremez ve değiştiremezsiniz.

**UTP** Unshielded twisted-pair: Kullanıcı cihazlarını switch ya da hub'a bağlamak için küçükten-genişe ağlarda kullanılan, bakır kablolalaması. Ayrıca switch'i-switch'e veya hub'ı hub'a bağlamakta da kullanılır.

**VBR** Variable bit rate: Real time (RT) klas ve non-real-time (NRT) klaslara bölünenen ATM ağlarında kullanmak için ATM Forumu tarafından tanımlı bir QoS klasıdır. RT, bağlantılar, örnekler arasında sabit bir zaman ilişkisine sahip olduklarında, çalıştırılırlar. Tersine, NRT, bağlantılar, sabit olmayan zaman ilişkisine sahip olduklarında çalıştırılırlar, fakat hala garanti edilmiş bir QoS'a ihtiyaç duymaktadır.

**VCC** Virtual channel connection: VCL'ler (virtual channel links) tarafından oluşturulan mantıksal bir devre. VCC'ler, bir ATM ağında iki uç nokta arasında veri taşır. Bazen, sanal devre bağlantısı olarak belirtilir.

**Veri enkapsülasyonu** Bir protokoldeki bilginin, başka protokoldeki veri bölümünde sarmalandığı veya taşındığı bir işleyiştir. OSI referans modelinde, her bir katman, hemen üzerindeki katmana enkapsülasyon yapar, böylece veri protokol yığının aşağısına doğru akar.

**Veri frame'i** OSI referans modelinin Data Link katmanındaki Protokol Data Ünitesi (PDU) enkapsülasyonu. Ağ katmanından paketleri enkapsüle eder ve bir ağ ortamındaki aktarımlar için veriyi hazırlar.

**Veri sıkıştırma** Bakınız: Sıkıştırma

**VIP** Versatile Interface Processor: Çok katmanlı anahtarlama sağlayan ve Cisco IOS yazılımı çalıştıran, Cisco 7000 ve 7500 serisi router'lar için bir interface kartı. VIP'nin en güncel versiyonu, VIP2'dir. (2) Virtual IP: Switch portu üzerinde Virtual Networking Servisleri çalıştırmak için mantıksal ayrılmış anahtarlamalı IP çalışma gruplarına, bunu mümkün yapan bir fonksiyondur.

**virtual circuit (VC)** Ağdaki iki cihaz arasında güvenli iletişim sağlamak için tasarlanan mantıksal bir devredir. VPI/VCI çifti tarafından tanımlıdır. Sanal bir devre, kalıcı (PVC) ya da anahtarlamalı (SVC) olabilir. Sanal devreler, Frame Relay ve X.25'de kullanılmaktadır. ATM'de sanal kanal (virtual channel) olarak bilinmektedir. *Ayrıca bakınız: PVC ve SVC.*

**VLAN ID** Bazen VLAN rengi olarak da belirtilen VLAN ID, frame'in VLAN üyeliğini bir switch'e belirtmek için yapılan etiketlemedir.

**VLAN** Virtual LAN: Bir veya daha fazla mantıksal segmentlere ayrılmış LAN'da (yönetim yazılımı kullanılırak yapılandırılan) cihazlar grubu. Cihazların, gerçekten çok farklı LAN segmentlerinde bulunduklarında, aynı fiziksel ortam aracına bağlıymış gibi iletişimde olmalarını mümkün kılar. VLAN'ler, fiziksel bağlantılar yerine mantıksal tabanlıdırlar ve bundan dolayı çok esnektirler.

**VLSM** Variable Length Subnet Mask: Uygun adres aralığı optimize etmeye yardımcı olur ve çeşitli subnet'lerde aynı ağ adresi için farklı subnet maskesi belirtir. Yaygın olarak, bir subnet'i subnet'leme olarak da belirtilir.

**VMPS** VLAN Management Policy Server: VLAN'leri bir switch portuna dinamik olarak atamak için kullanılmaktadır.

**VPN** Virtual private network: Internet gibi public bir ağ boyunca point-to-point mantıksal bağlantıları şifrelemenin bir yöntemi. Bu, public bir ağ boyunca güvenli bağlantılara izin verir.

**VTP tranparent modu** VLAN Trunk Protokol VLAN bilgisini alıp aktaran fakat bilgiyi okumayan switch modudur.

**VTP** VLAN Trunking Protocol: Bir VTP sunucu tarafından yapılandırılan VLAN'ler hakkında bir switch yapısındaki switch'leri güncellemek için kullanılır. VTP cihazları, VTP server, celient veya transparent olabilir. Sever modu, client modunu günceller. Transparent cihazlar sadece lokal cihazlardır ve VTP kullanıcıları ile bilgi paylaşmazlar. VTP cihazları, VLAN bilgisini sadece trunk olan linklerden öğrenir.

**WAN** Wide area network: Bir DCE (data communications equipment) ağı üzerinden LAN'ları bağlamak için kullanılan bir gösterimdir. Tipik olarak, bir WAN, PSTN ağı üzerindeki bir kiralık-hat veya dial-up bağlantısıdır. Frame Relay, PPP, ISDN ve HDLC, WAN protokol örnekleridir.

**WINS** Windows Internet Name Service: NetBIOS isimlerinden TCP/IP adreslerine, isim çözümleme veritabanı.

**Wildcard** Access listeleri ve OSPF yapılandırması için kullanılır. Wildcard'lar, bir subnet aralığı belirtmek için kullanılan gösterim yöntemidir.

**Windowing – pencereleme** OSI modelinin Transport katmanında TCP ile kullanılan, akış-kontrol yöntemidir.

**WinSock** Windows Socket Interface: Bir uygulama çeşidinin kullanılmasını mümkün kılan ve internet bağlantısını paylaşan bir yazılım arayüzüdür. WinSock yazılımı, bağlantı başlatan bir dialer (çevirici) programı gibi destek programlarıyla dynamic link library (DLL) içeren bir yazılımdır.

**Workgroup katmanı** Distribution katmanı, bazen workgroup katmanı olarak belirtilir ve access ile core katmanları arasındaki iletişim noktasıdır. Distribution katmanının ana fonksiyonu, routing, filtreleme ile WAN erişimi sağlamak ve ihtiyaç olduğunda, paketlerin, core'a nasıl erişebileceğini belirtmektir.

**Workgroup switching** Hem Ethernet ağları arasında yüksek-hız (100Mbps) trasparent bridging hem de Ethernet ile CDDI veya FDDI arasında yüksek-hızlı translational bridging sağlayan bir switching yöntemidir.

**X Window** X terminalleri ve Unix iş istasyonları arasındaki iletişim için orjinal olarak, MIT tarafından geliştirilen bir dağıtık çok görevli pencereleme ve grafik sistemidir.

**X.25** DTE ve DCE ağ cihazları arasındaki iletişimi belirtmek için bir ITU-T paket-switching standartıdır. X.25, adı LAPB olan güvenli bir Data link katman protokolü kullanır. X.25, aynı zamanda, Network katmanında PLP kullanır. X.25 yerine, çoğunlukla Frame Relay kullanılır.

**Yayılma gecikmesi** Verinin, kaynağından hedefine bir ağı geçmek için geçen süredir.

**Yazılım adresi** Bir mantıksal adres olarak da bilinir. Bu tipik olarak bir IP adresidir, fakat bir IPX adresi de olabilir.

**Yedekleme** Ağlararası iletişimde, birincil bağlantı, cihaz veya servisin arızalanması durumunda, yedek (backup) olarak kullanılabilen, bağlantı, cihaz veya servislerin kopyalanması.

**Yetki bölgesi** Bir isim sunucusunun yetkili olması için DNS ile ilişkili domain hiyerarşisinin bir bölümü. *Ayrıca bakınız: DNS.*

**Yönlendirilebilir protokol** Yönlendirilebilir protokoller (IP ve IPX gibi), bir ağ topluluğu boyunca kullanıcı verisinin aktarılmasında kullanılmaktadır. Tersine, routing protokolleri (RIP, IGRP ve OSPF gibi), router'lar arasında routing tablolarının güncellenmesinde kullanılmaktadır.

**ZIP storm** Uygulama sırasında, uygun bir zone ismi olmadığı için, AppleTalk çalışan bir router'ın bir route'u ürettiği veya aktardığı zaman oluşan bir broadcast fırtınası. Ondan sonra route diğer downstream router'lar tarafından gönderilir, bu durum bir ZIP fırtınasına neden olur. *Ayrıca bakınız: Broadcast fırtınası ve ZIP.*

**ZIP** Zone Information Protocol: Ağ numaralarını, zone isimlerine eşleştirmek için AppleTalk tarafından kullanılan bir Session katmanı protokolüdür. NBP, bir zone'a ait düğümleri içeren ağların belirlenmesinde ZIP kullanır.

**Zone** AppleTalk'ta ağ cihazlarının mantıksal bir gruplaması. Aynı zamanda DNS'de de kullanılır. *Ayrıca bakınız: ZIP.*