Modern Datacenter Overlay Network Technologies

Ali Aydemir

First Switch in the History?

Emma Nutt (1860–1915) became the world's first Switch
on 1 September 1878 when she started working for
the Edwin Holmes Telephone Despatch Company in Boston, Massachusetts, USA.
Emma was hired by Alexander Graham Bell.
She was paid a salary of $10 per month for a 54 hour week.

A few hours after Emma started working, her sister, Stella Nutt,
became the world's second Switch, also making the pair the first two Switches in history.

NOT: To be an operator, a woman had to be unmarried [clarification needed] and between the ages of seventeen and twenty-six. She had to look prim and proper, and have arms long enough to reach the top of the tall telephone switchboard. Like many other American businesses at the turn of the century, telephone companies discriminated against people from certain ethnic groups and races. For instance, African-American and Jewish women were not allowed to become operators.

A large Bell System international switchboard in 1943

Switched Network Design

A Study of Non-Blocking Switching Networks

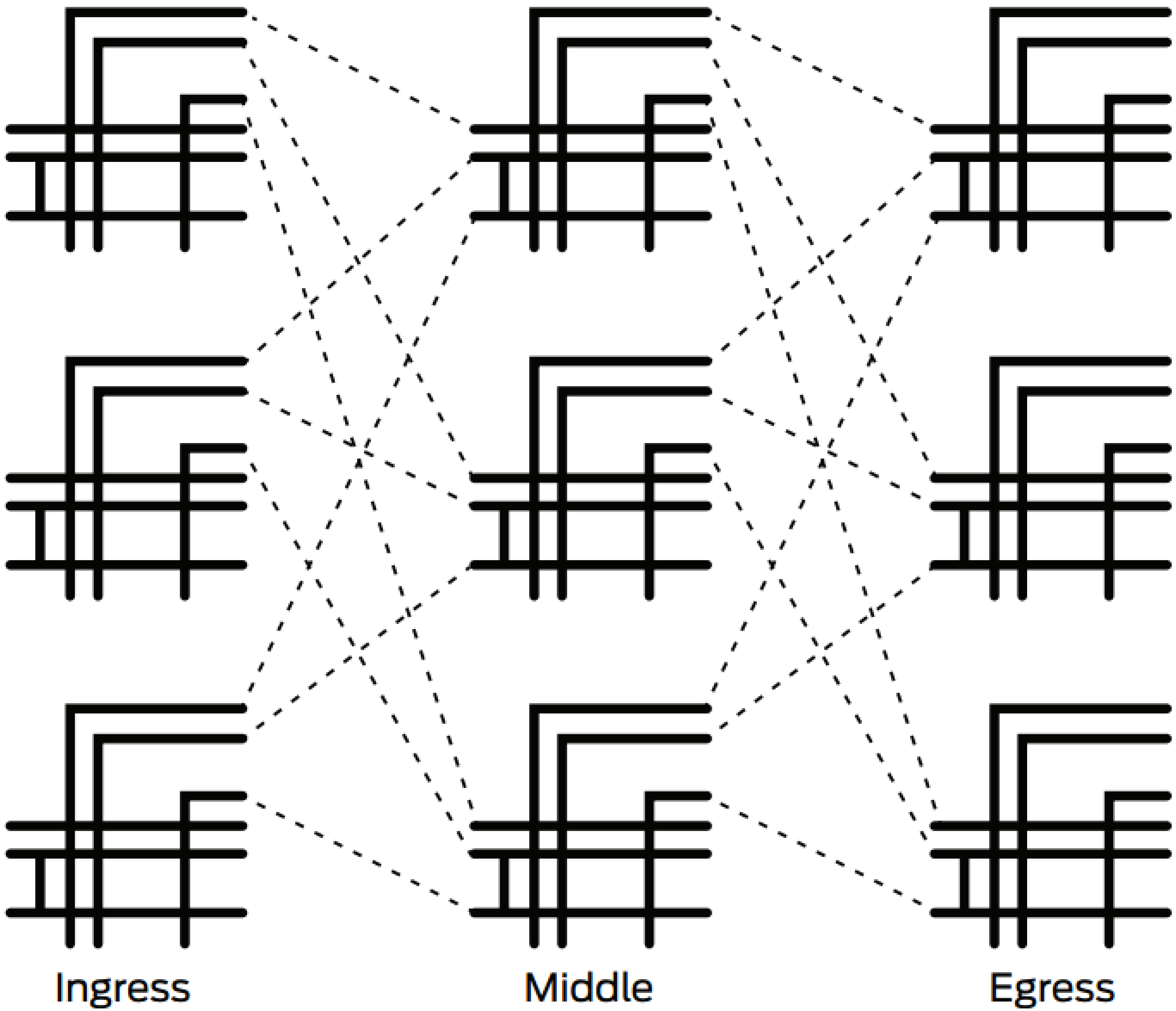
By CHARLES CLOS

(Manuscript received October 30, 1952)

This paper describes a method of designing arrays of crosspoints for use in telephone switching systems in which it will always be possible to establish a connection from an idle inlet to an idle outlet regardless of the number of calls served by the system.

http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6770468

Bell Labs researcher **Charles Clos**, who proposed the model in 1952 as a way to overcome the performance- and cost-related challenges of electromechanical switches then used in telephone networks. Clos used mathematical theory to prove that achieving **fully non-blocking** performance in a "switching array" (now known as a *fabric*) was possible if the switches were organized in a hierarchy.



Fat-Trees: Universal Networks for Hardware-Efficient Supercomputing

CHARLES E. LEISERSON, MEMBER, IEEE

Abstract — This paper presents a new class of universal routing networks called *fat-trees*, which might be used to interconnect the processors of a general-purpose parallel supercomputer. A fat-tree routing network is parameterized not only in the number of processors, but also in the amount of simultaneous communication it can support. Since communication can be scaled independently from number of processors, substantial hardware can be saved over, for example, hypercube-based networks, for such parallel processing applications as finite-element analysis, but without resorting to a special-purpose architecture.

Of greater interest from a theoretical standpoint, however, is a proof that a fat-tree of a given size is nearly the best routing network of that size. This *universality theorem* is proved using a three-dimensional VLSI model that incorporates wiring as a direct cost. In this model, hardware size is measured as physical volume. We prove that for any given amount of communications hardware, a fat-tree built from that amount of hardware can simulate every other network built from the same amount of hardware, using only slightly more time (a polylogarithmic factor greater). The basic assumption we make of competing networks is the following. In unit time, at most $O(a)$ bits can enter or leave a closed three-dimensional region with surface area a. (This paper proves the universality result for *off-line* simulations only.)

Index Terms — Fat-trees, interconnection networks, parallel supercomputing, routing networks, universality, VLSI theory.

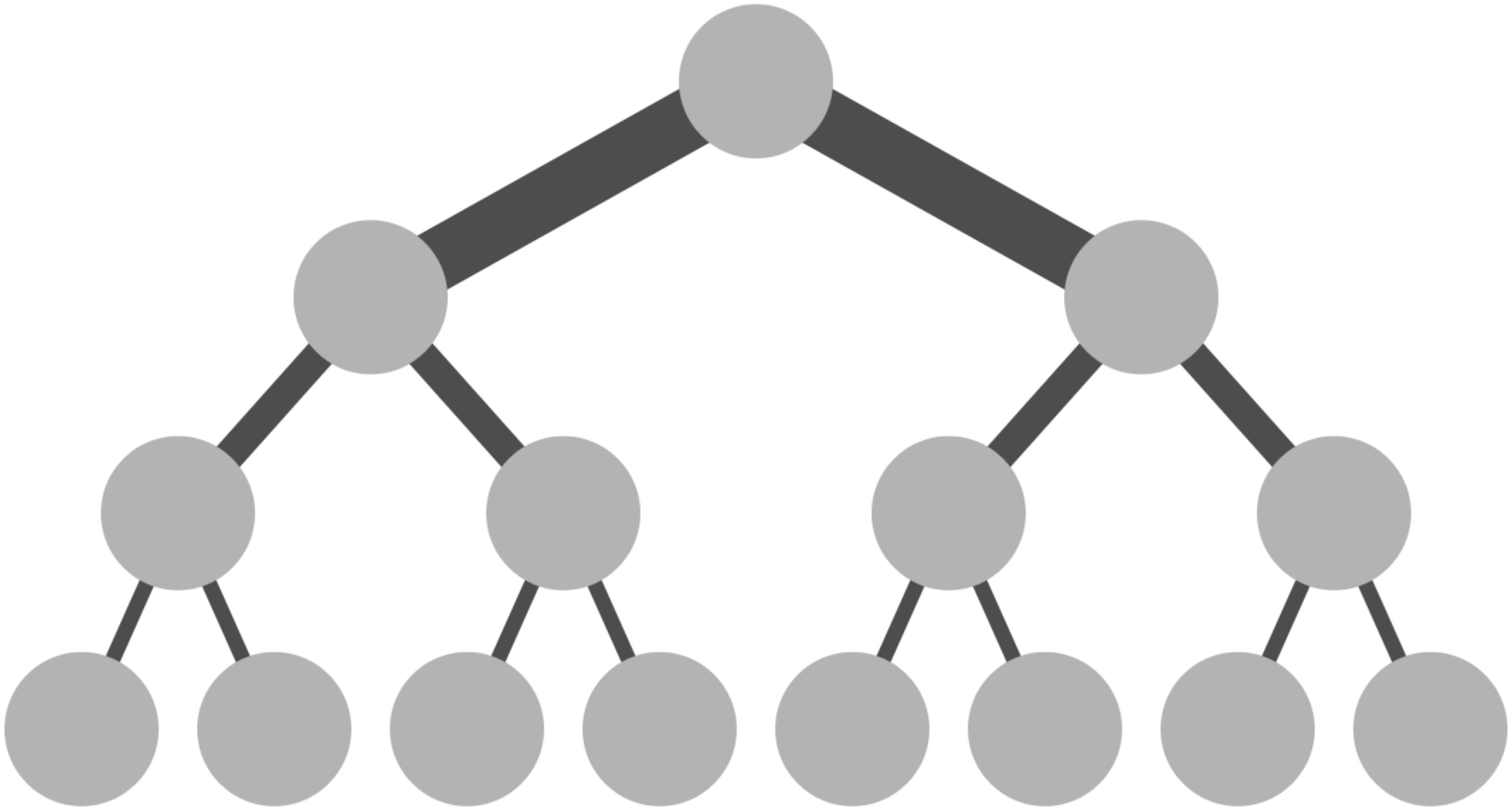
further from the leaves. In physical structure, a fat-tree resembles, and is based on, the *tree of meshes* graph due to Leighton [12], [14]. The processors of a fat-tree are located at the leaves of a complete binary tree, and the internal nodes are switches. Going up the fat-tree, the number of wires connecting a node with its father increases, and hence the communication bandwidth increases. The rate of growth influences the size and cost of the hardware as well.

Most networks that have been proposed for parallel processing are based on the Boolean hypercube, but these networks suffer from wirability and packaging problems and require nearly order $n^{3/2}$ physical volume to interconnect n processors. In his influential paper on "ultracomputers" [27], Schwartz demonstrates that many problems can be solved efficiently on a supercomputer-based on a shuffle network [28]. But afterwards, Schwartz comments, "The most problematic aspect of the ultracomputer architecture suggested in the preceding section would appear to be the very large number of intercabinet wires which it implies." Schwartz then goes on to consider a "layered" architecture, which seems easier to build, but which may not have all the nice properties of the original architecture.

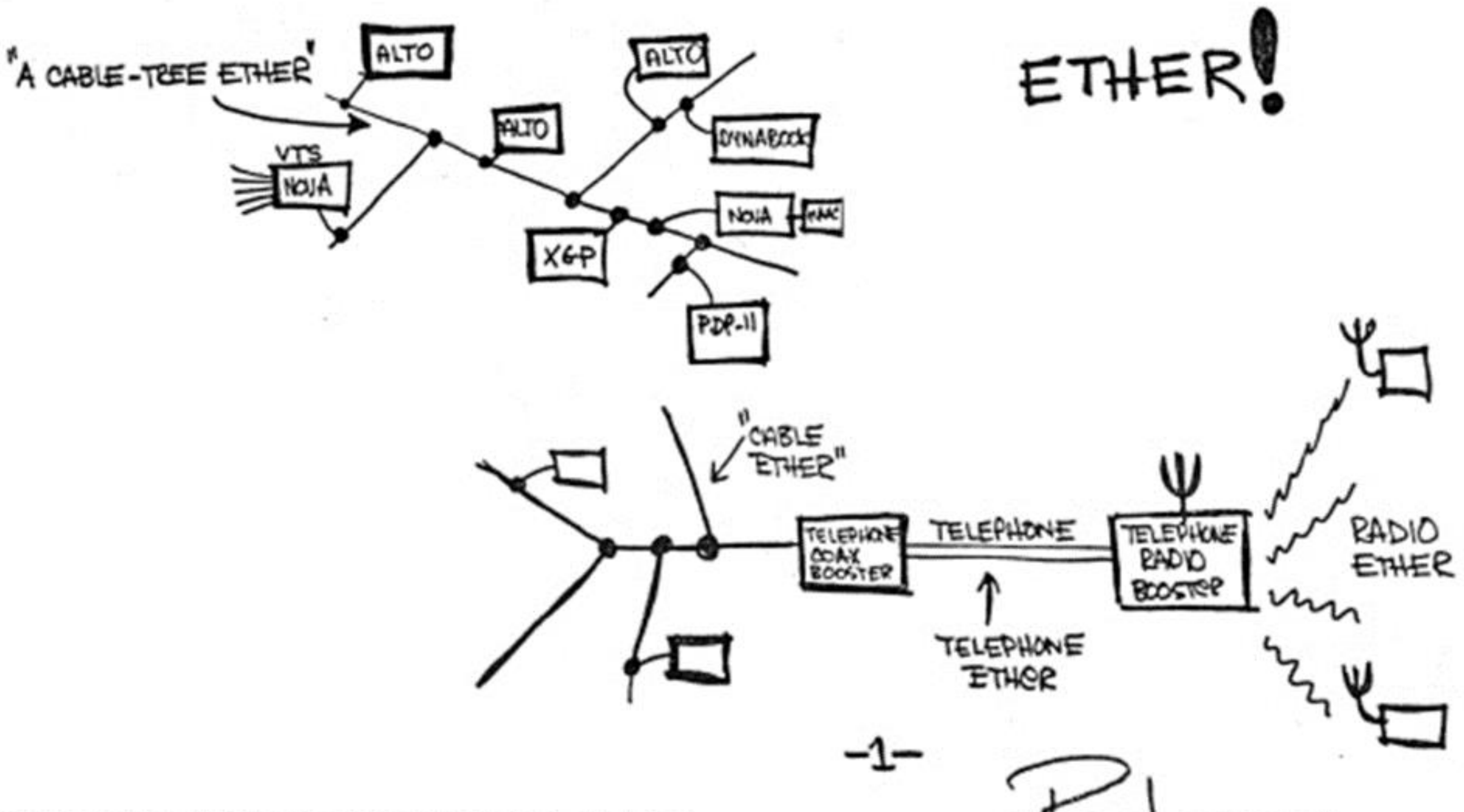
http://courses.csail.mit.edu/6.896/spring04/handouts/papers/fat_trees.pdf

The **Fat Tree** network is a universal network for provably efficient communication. It was invented by **Charles E. Leiserson** of the Massachusetts Institute of Technology.

This topology is actually a special instance of a Clos network, rather than a fat-tree as described above. That is because the edges near the root are emulated by many links to separate parents instead of a single high-capacity link to a single parent. However, many authors continue to use the term in this way.



The Ether-Net and Paradigm Change



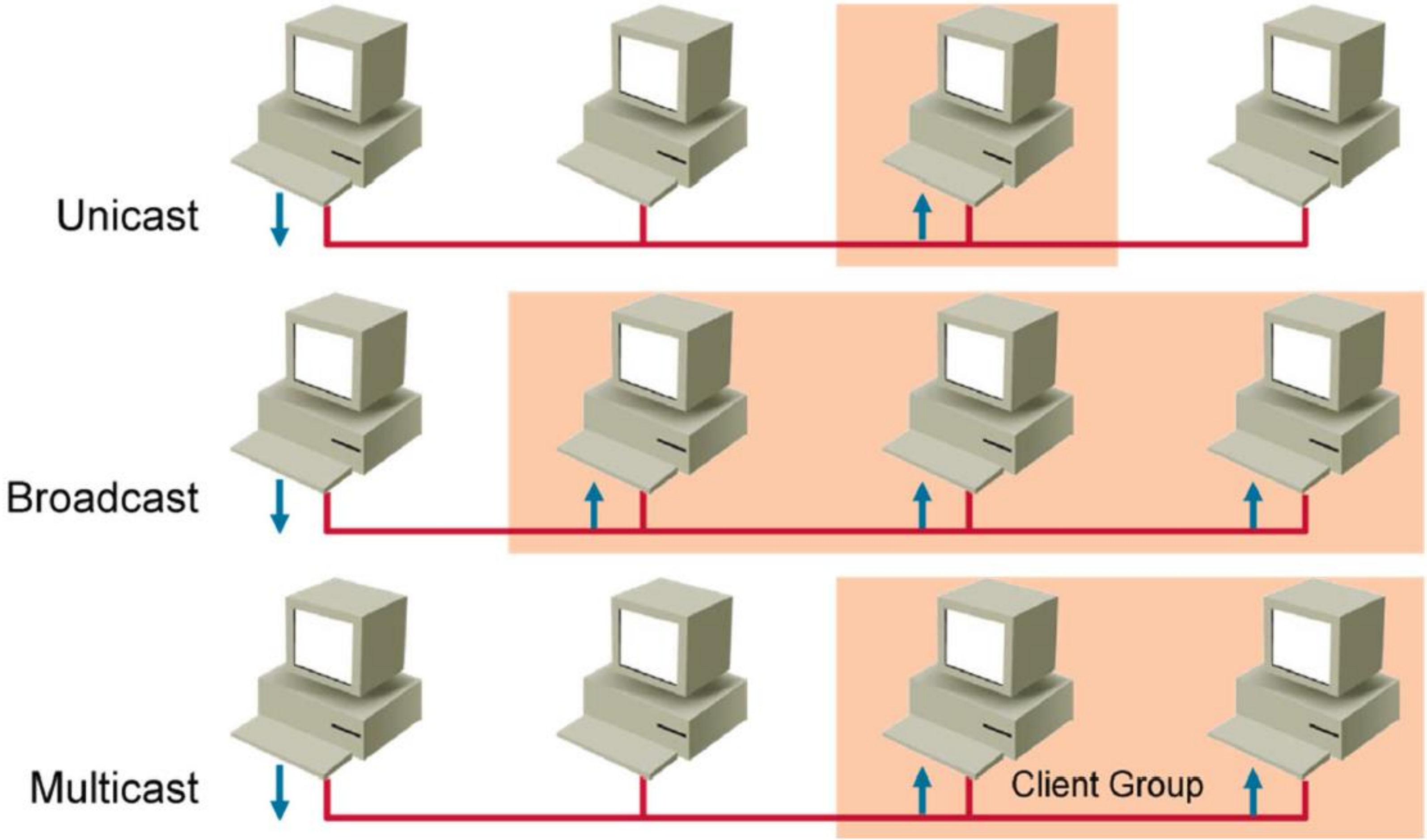
Bob Metcalfe's 1972 sketch of his original "ethernet" vision
Image provided courtesy of Palo Alto Research Center Inc., a Xerox Company

Ethernet Frame Structure

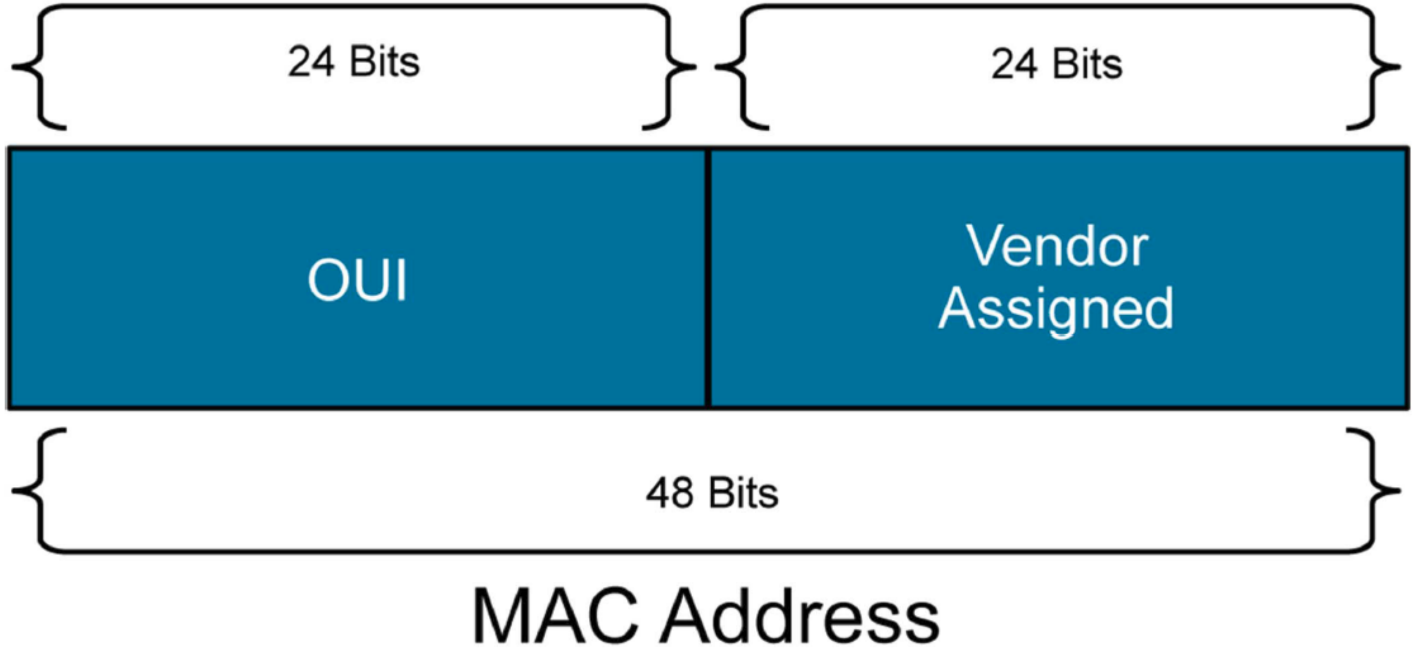
Typical Ethernet Frame					
8 bytes	6	6	2	46-1500	4
Preamble	Destination Address	Source Address	Type	Data	FCS

- FCS = frame check sequence
- Field length is stated in bytes.

MAC Addresses



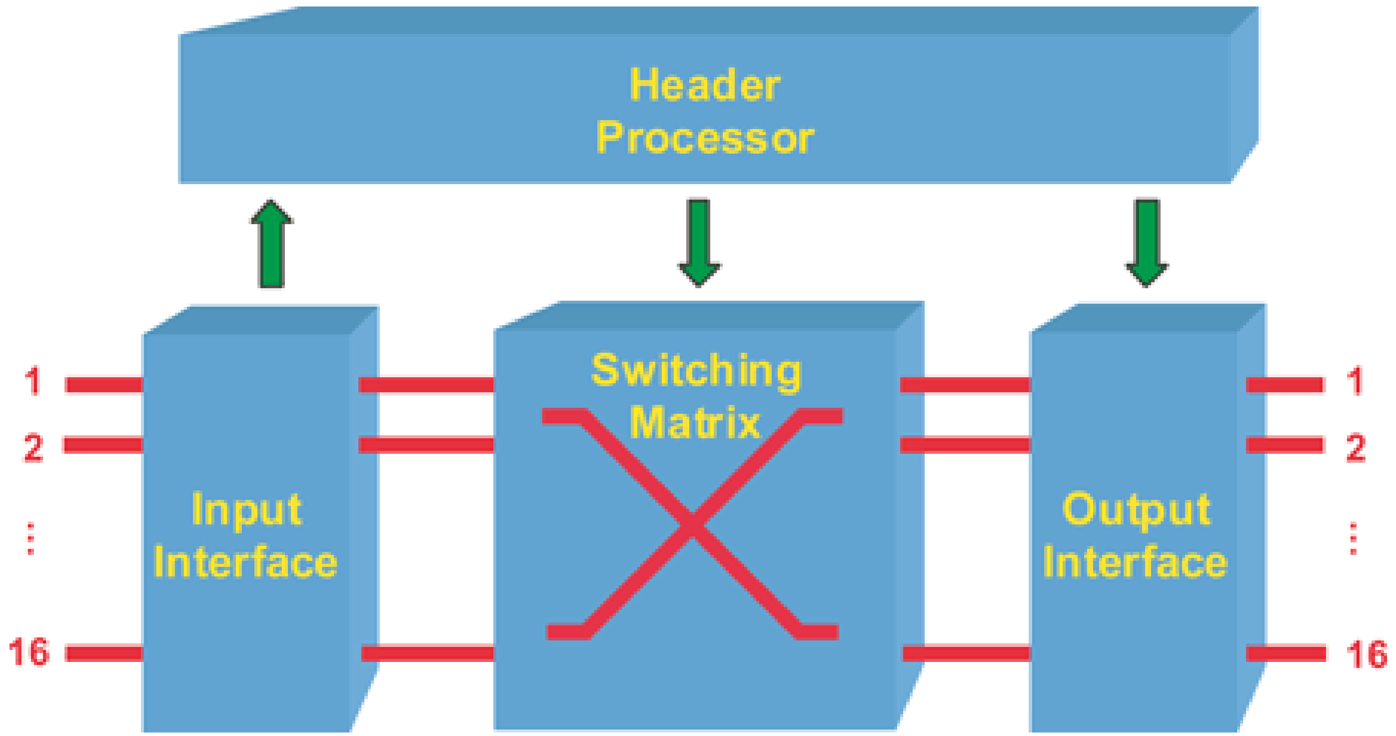
MAC Addresses (Cont.)



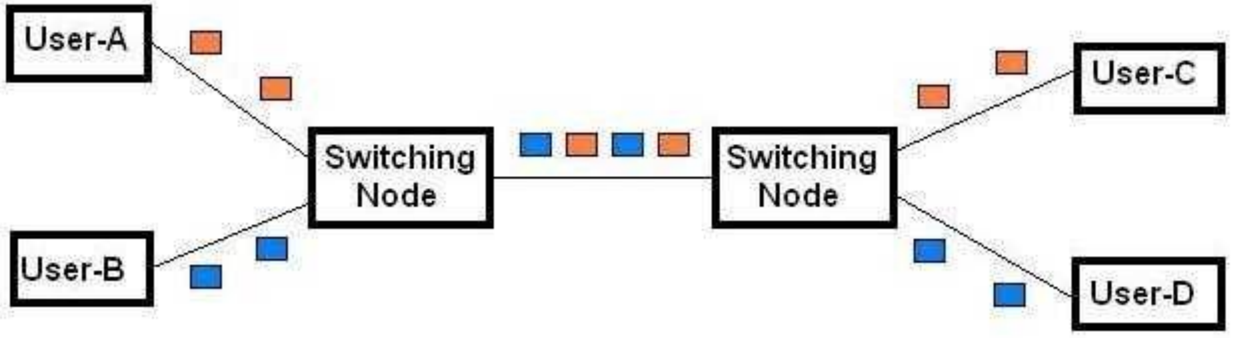
Different display formats:

- 0000.0c43.2e08
- 00:00:0c:43:2e:08
- 00-00-0C-43-2E-08

Packet Switching



Header
Processor
1
2
16
Input
Interface
Switching
Matrix
Output
Interface
1
2
16



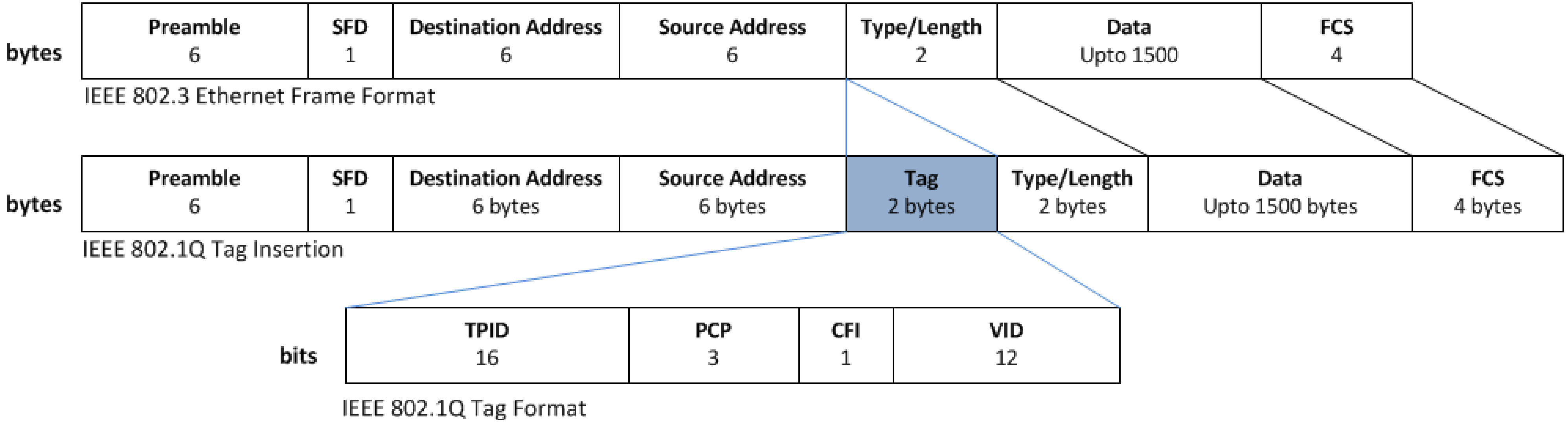
User-A
User-B
Switching
Node
Switching
Node
User-C
User-D

The Ether-Net Problems

L2 networks **did not** scale ➔

1. The MAC address
   - L2 addressing = MAC address
     The MAC address is a flat address with no summarization or hierarchy possible
2. No Scalable Control Plane
   - With no addressing hierarchy possible it was not possible to have a Link State Protocol for L2 networks which could scale
3. No L2 OAM tools
4. Limited Virtualization
   - Only 802.1Q VLAN tagging
5. LOOP never Stop
   - There is no TTL in Ethernet Header

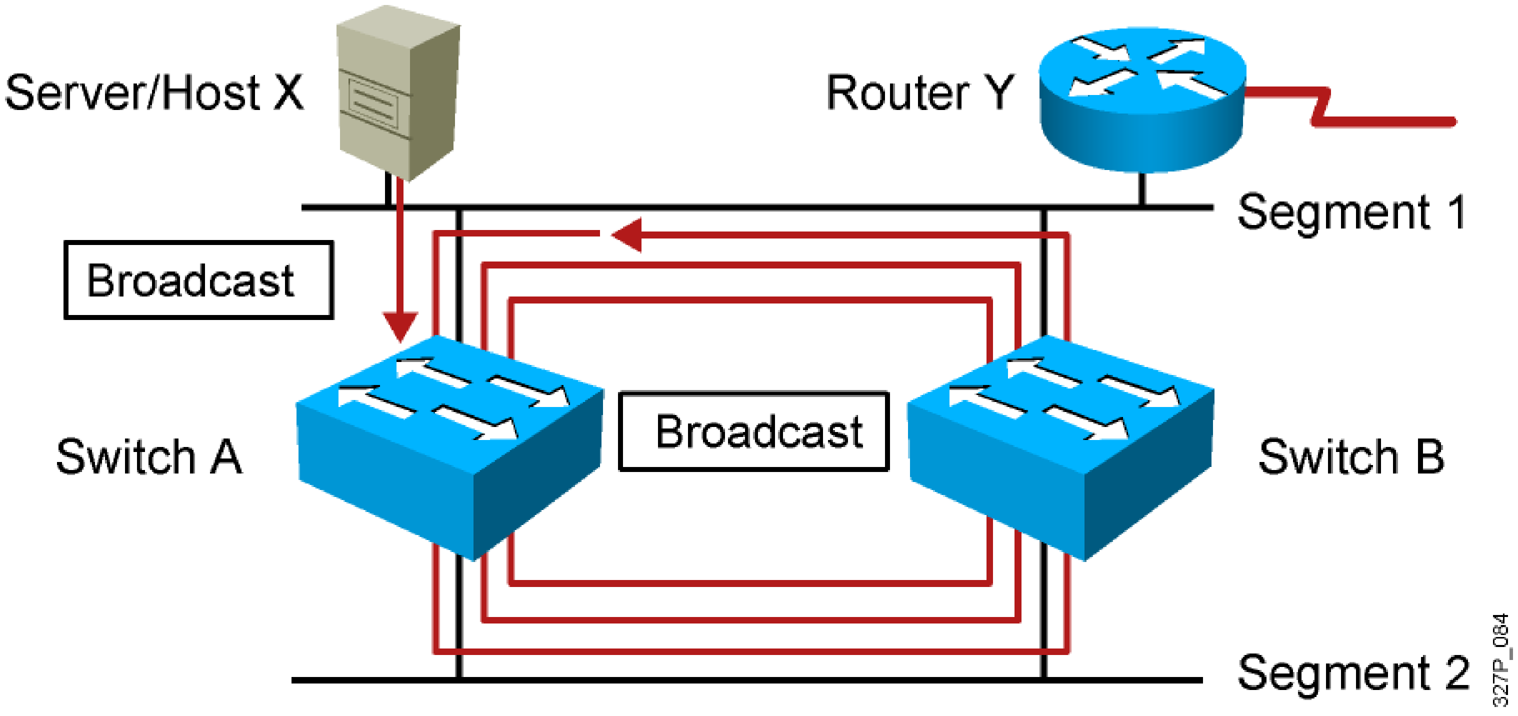
Layer 2 Virtualization



TPID = Tag Protocol Identifier
PCP = Priority Code Point
CFI = Canonical Format Indicator
VID = VLAN Identifies (VLAN ID)

Layer 2 LOOP

The **loop** creates broadcast storms as broadcasts and multicasts are forwarded by switches out every port, the switch or switches will repeatedly rebroadcast the broadcast messages flooding the network. Since the Layer 2 header does not support a time to live (**TTL**) value, if a frame is sent into a looped topology, it can loop forever.



Typical Ethernet Frame					
8 bytes	6	6	2	46-1500	4
Preamble	Destination Address	Source Address	Type	Data	FCS

- Host X sends a broadcast.
- Switches continue to propagate broadcast traffic over and over.

What about STP?

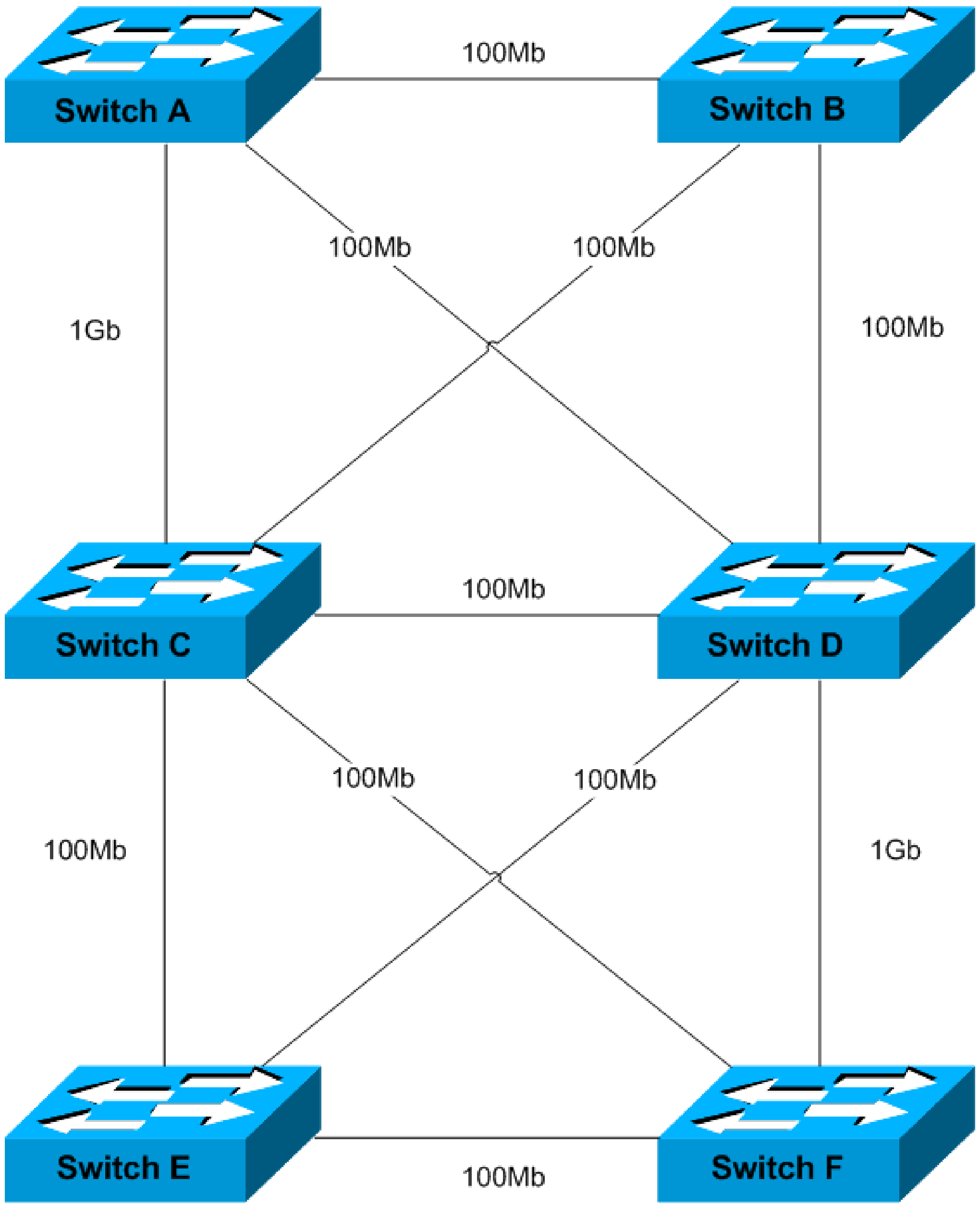
Radia Joy Perlman (born January 1, 1951) is a software designer and network engineer. She is most famous for her invention of the spanning-tree protocol (STP). She is currently employed by EMC Corporation.

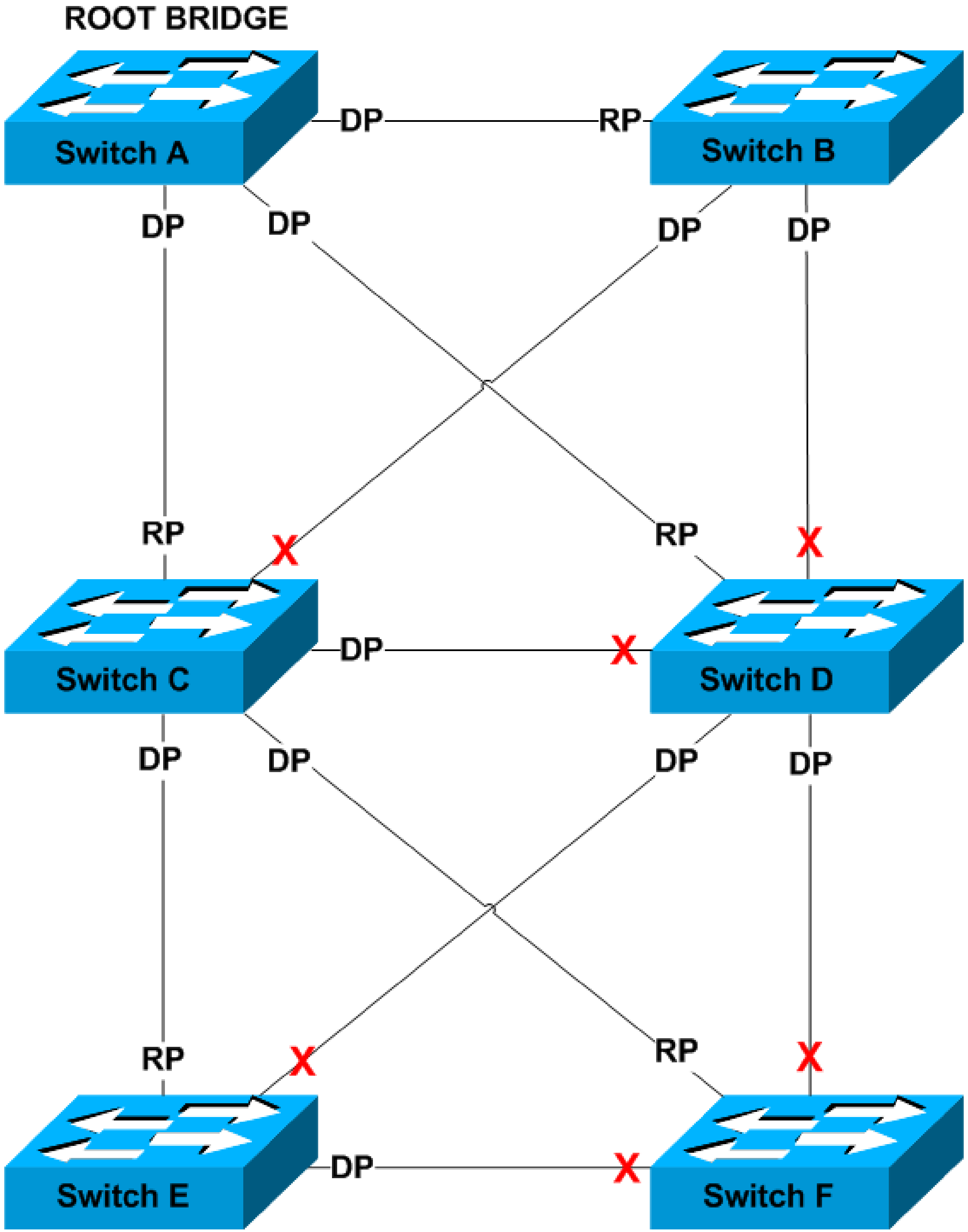
The **Spanning Tree Protocol (STP)** is a network protocol that builds a logical loop-free topology for Ethernet networks.

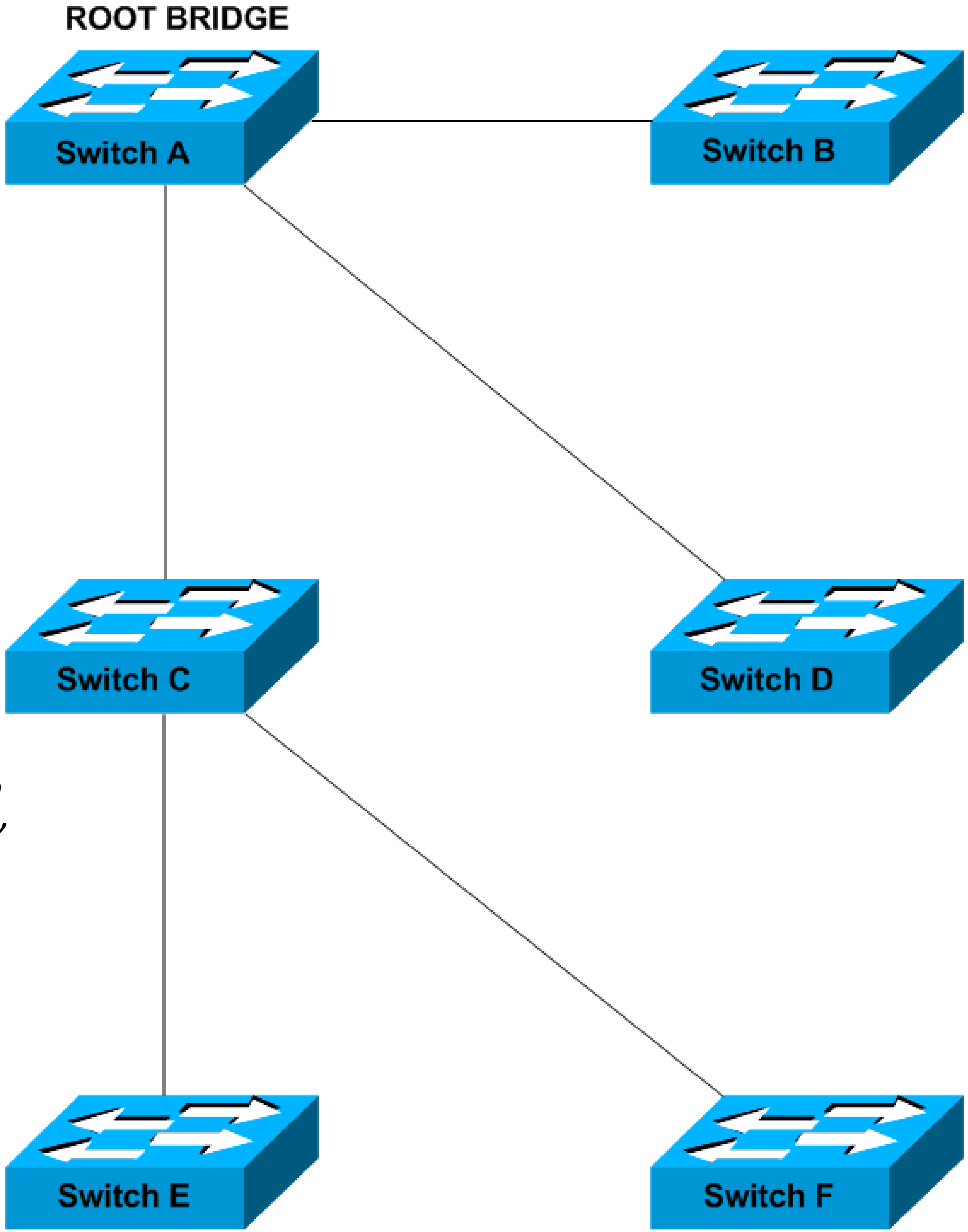
The basic function of STP is to prevent bridge loops and the broadcast radiation that results from them.

Spanning tree also allows a network design to include spare (redundant) links to provide automatic backup paths if an active link fails, without the danger of bridge loops, or the need for manual enabling or disabling of these backup links.

STP just changes **ring** topology to **linear** topology




ROOT BRIDGE
Switch A
Switch B
Switch C
Switch D
Switch E
Switch F
DP
RP
X

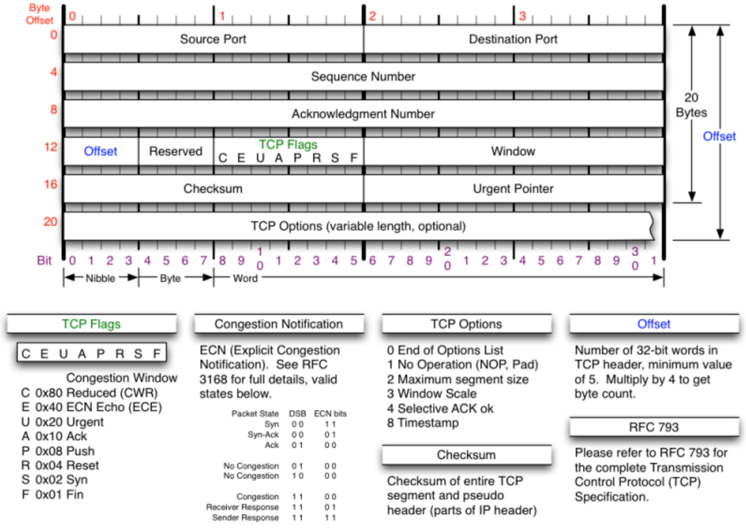


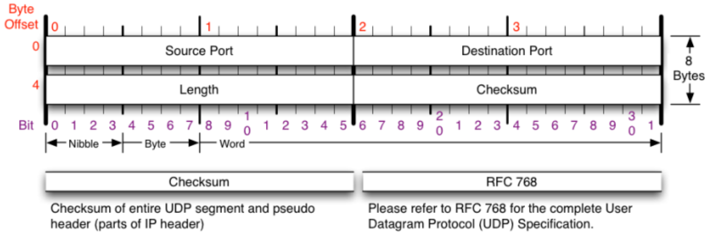
"I think that I shall never see
a graph more lovely than a tree.

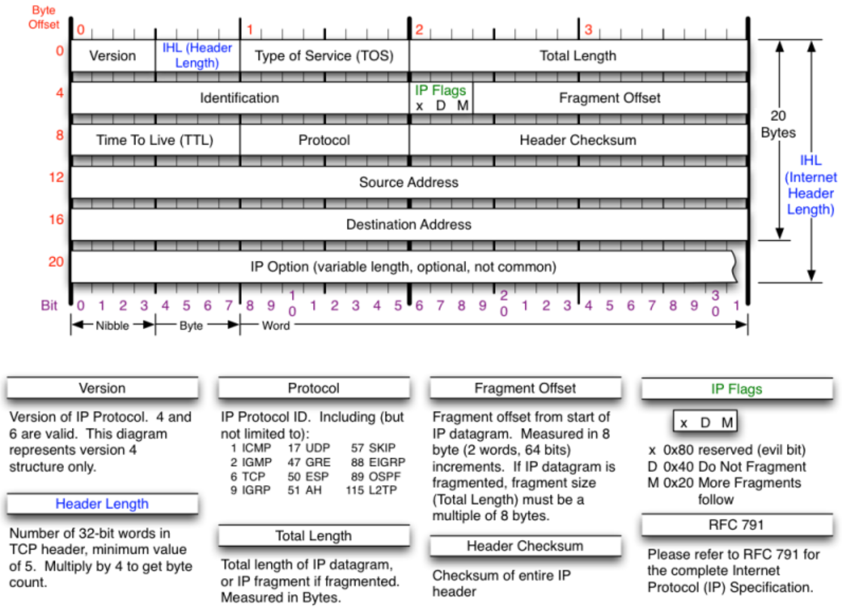
A tree whose crucial property
is loop-free connectivity.

A tree that must be sure to span
so packets can reach every LAN"

Is STP enough for resolving Ether-Net problems?



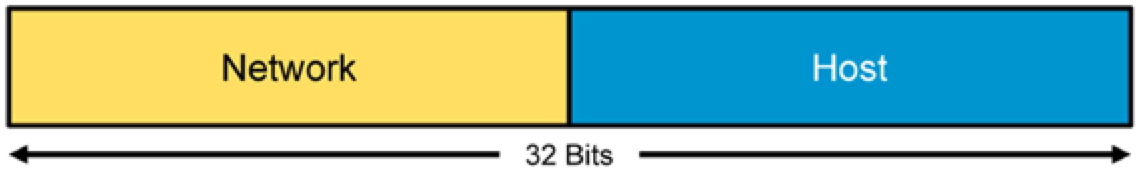




Vint Cerf the "fathers of the Internet“
TCP/IP inventor

IPv4 Header Address Fields

Ver.	IHL	Service Type	Total Length
Identification		Flag	Fragment Offset
Time to Live		Protocol	Header Checksum
Source Address			
Destination Address			
Options			Padding



So Layer 3 (IP) routing had to be used in the Core?

1. The IP address structure
   Can be summarized into networks using a netmask
   Core nodes do not need to know every single IP address on the network (they have no ARP cache)

2. Scalable Control Plane
   Availability of Link State Protocols such as: IS-IS & OSPF

3. IP OAM Tools
   ping, traceroute

4. IP Virtualization possible
   But requires BGP & MPLS for scalability

5. Loop will STOP
   There was a TTL in IP Header

L2 networks **did not** scale ➔ But Solved?

1. The MAC address
   - L2 addressing = MAC address
     The MAC address is a flat address with no summarization or hierarchy possible
2. No Scalable Control Plane
   - With no addressing hierarchy possible it was not possible to have a Link State Protocol for L2 networks which could scale
3. No L2 OAM tools

Solved by IP

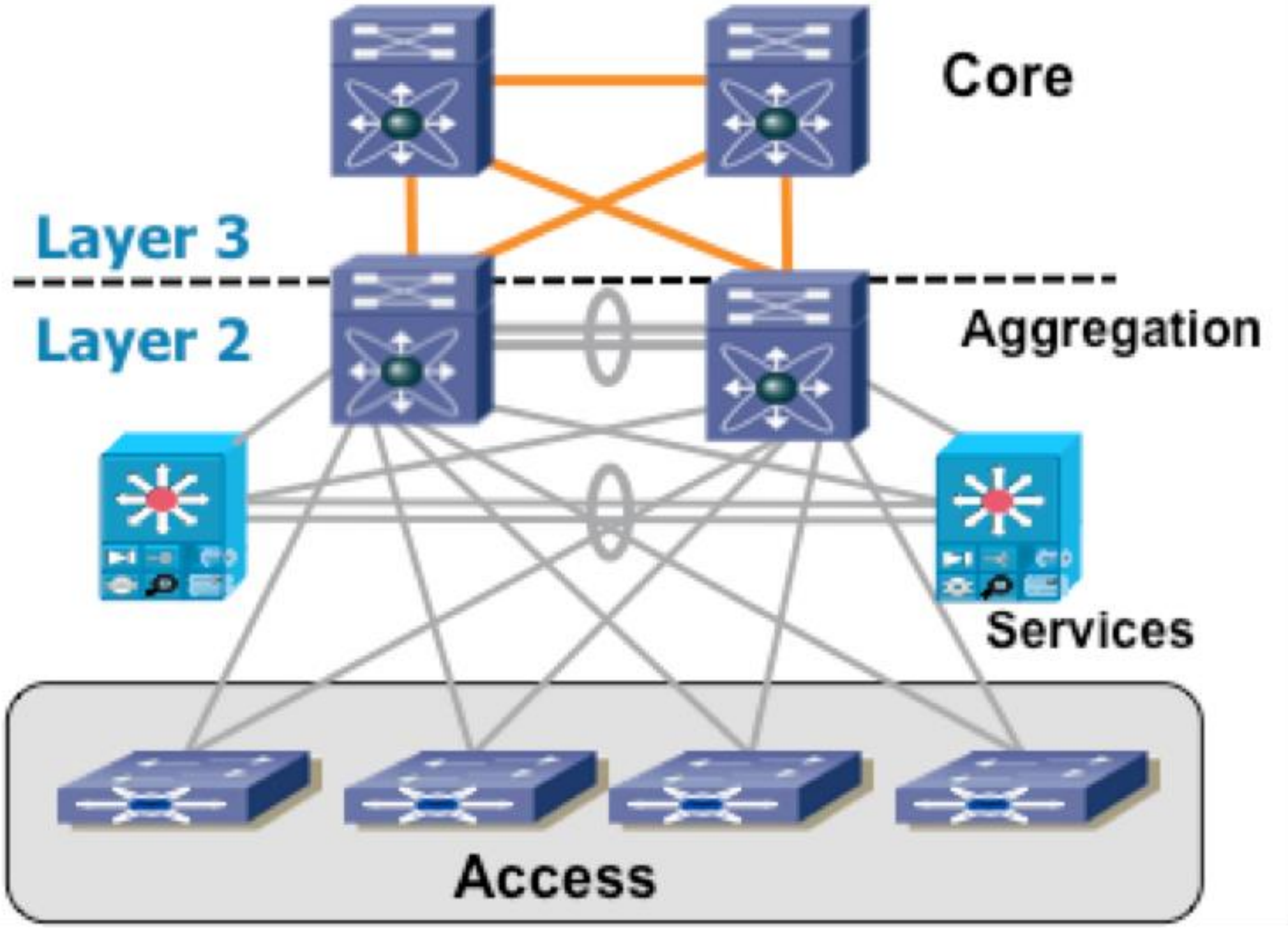
4. Limited Virtualization
   - Only 802.1Q VLAN tagging

Solved by 802.1q

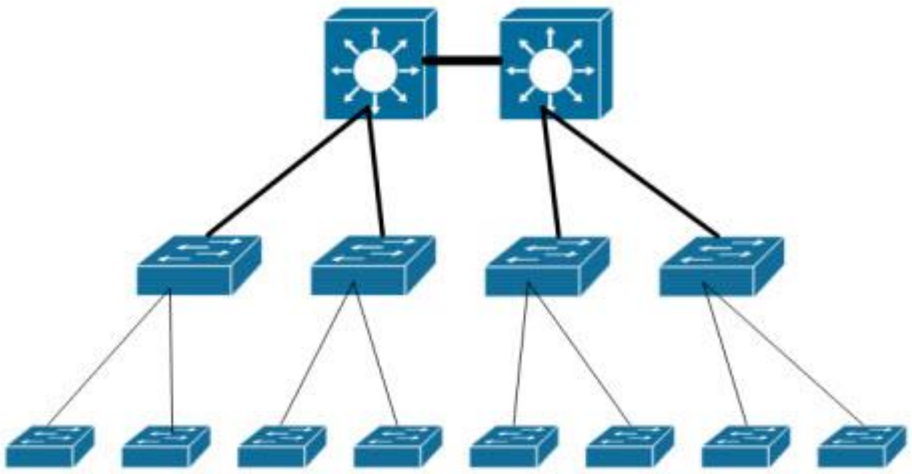
5. LOOP never Stop
   - There is no TTL in Ethernet Header

Solved by 802.1d STP

- IP Network **Over** Ethernet Network
- 802.1q Virtualization
- STP



Fat Tree Network



Over the years, networks started to use the **"fat tree"** model of connectivity using the core - distribution - access architecture. In order to prevent oversubscription, the link speeds got progressively higher as you reached the core.

The problem with traditional networks built using the spanning-tree protocol or layer-3 routed core networks is that a single **"best path"** is chosen from a set of alternative paths. All data traffic takes that **"best path"** until the point that it gets congested then packets are dropped. The alternative paths are not utilized because they topology algorithm deemed them to be less desirable or removed to prevent loops from forming. There is a desire to migrate away from using spanning-tree while still maintaining a loop-free topology yet utilizing all the multiple redundant links. If we could use a method of Equal-Cost Multi-Path (**ECMP**) routing, then performance could increase and the network would have better resiliency in the event of a link failure or a single switch failure.

New Problems Rising !

A new generation of applications:

- Search
- Big Data
- Clouds
- Other Web 2.0 Applications

Traffic Pattern

- Between servers (East-West) instead of client-server (North-South)

Scale

- 10s of thousands to 100s of thousands of endpoints

Agility

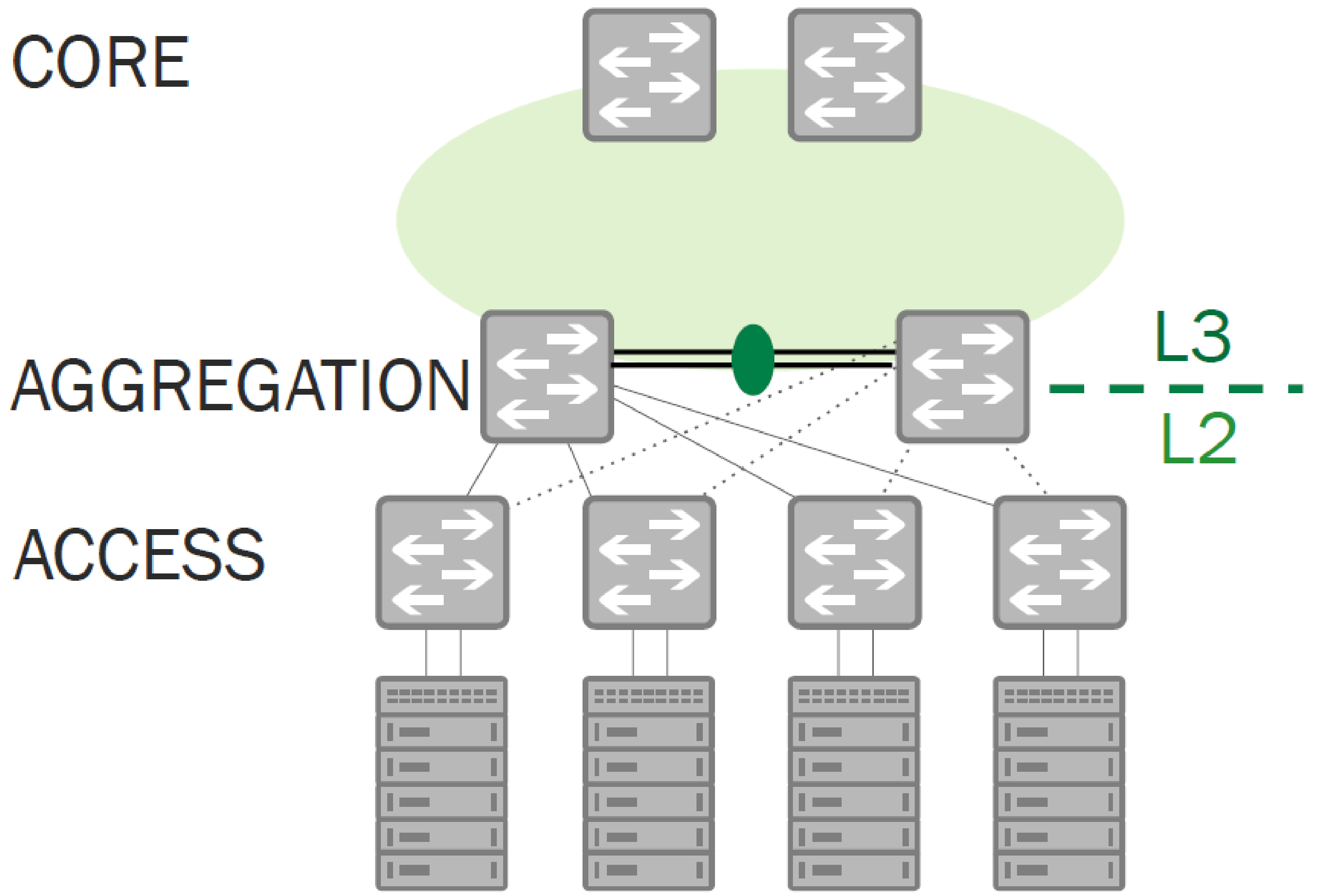
- New endpoints and racks powered up in hours instead of weeks
- New networks spun up in seconds instead of weeks

Flexibility

- Ability to reuse same infrastructure for different applications

Resilience

- Fine grained failure domain



Not suited for E-W traffic

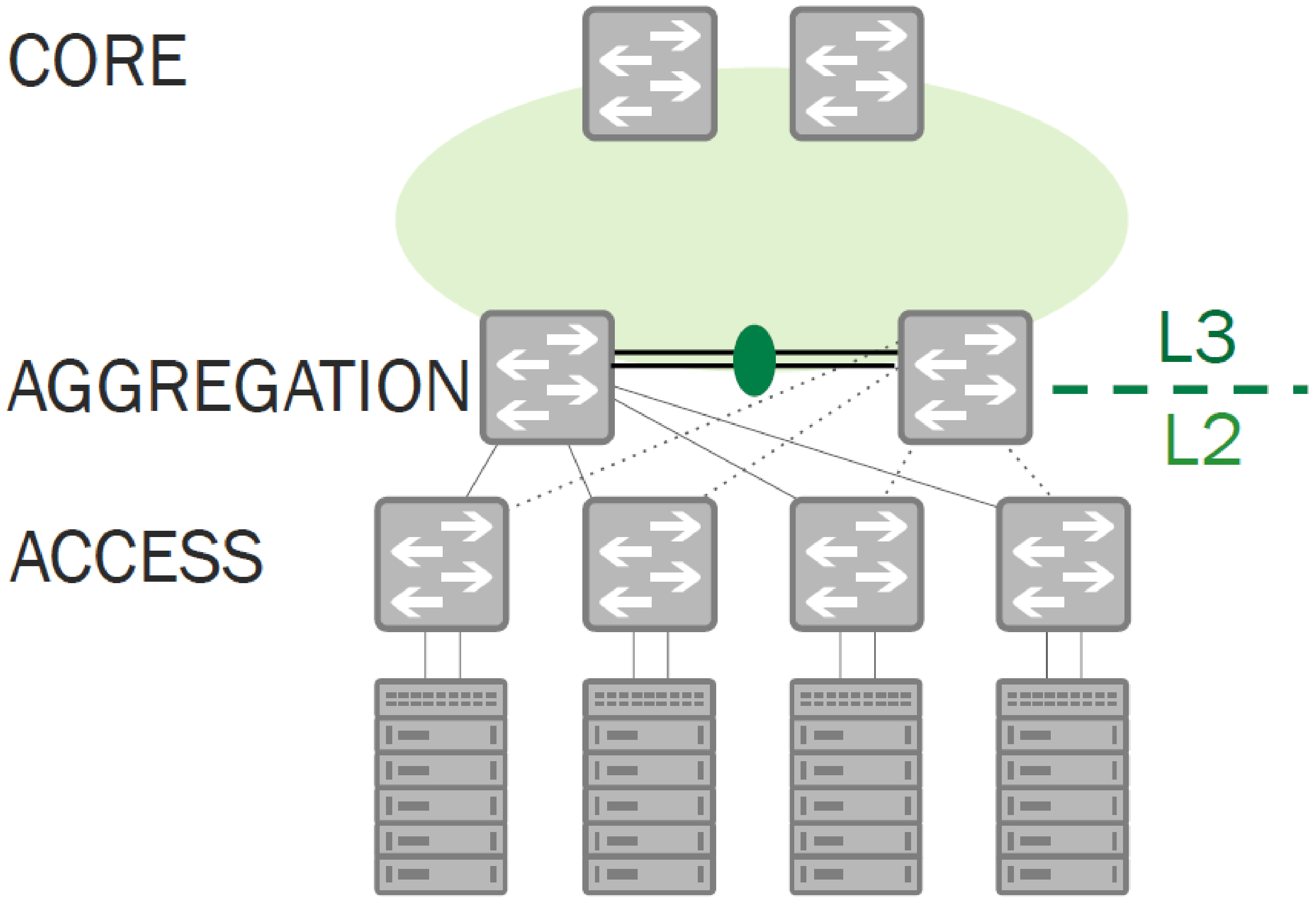
Heavy-core, lean edge design is not scalable

Not Agile

Inflexible Design

Coarse-grained failure domain

Unpredictable Latency



Too many protocols

- Many proprietary (MLAG, vPC, for example)
- STP and its variants, its myriad knobs, UDLD, Bridge Assurance, LACP, FHRP (VRRP, HSRP, GLBP), VTP, MVRP, etc. etc.

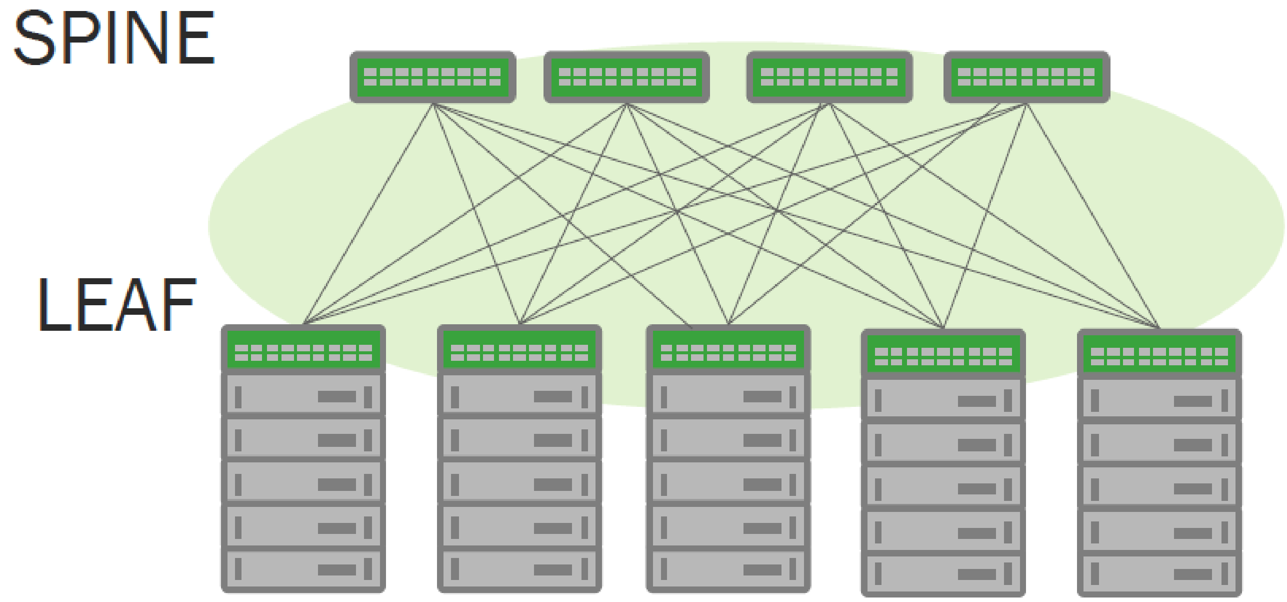
Dual redundancy only adds to the complexity mess

- Dual control planes
- HA
- ISSU etc.

Complex Failure Modes

is it possible to come back?

CLOS Network



Well matched for E-W traffic pattern

Scalable network topology

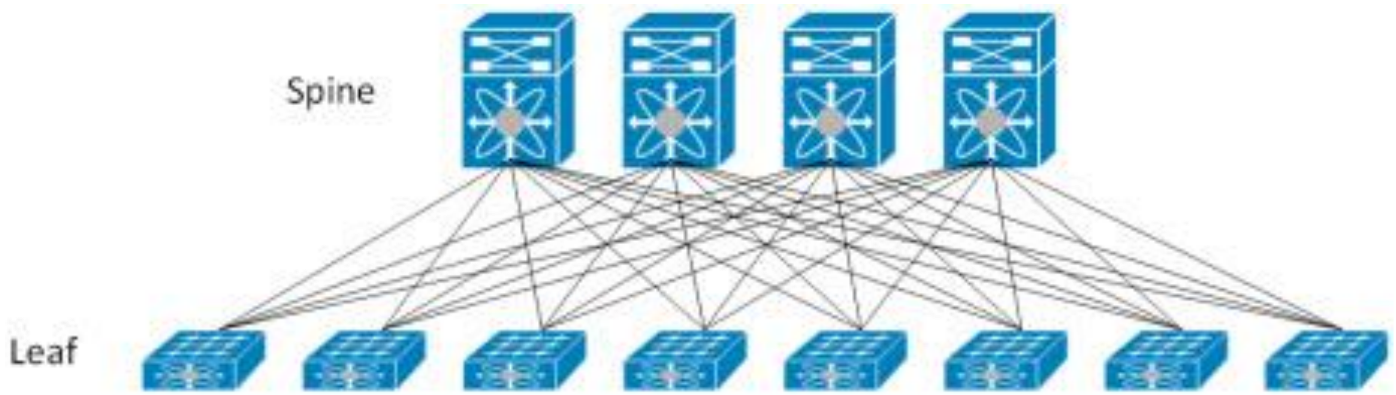
Reliance on ECMP leads to simple IP-based fabrics

Fine grained failure domain

Predictable latency

Coupled with network virtualization, serves as a basis for agility and flexibility

CLOS Network



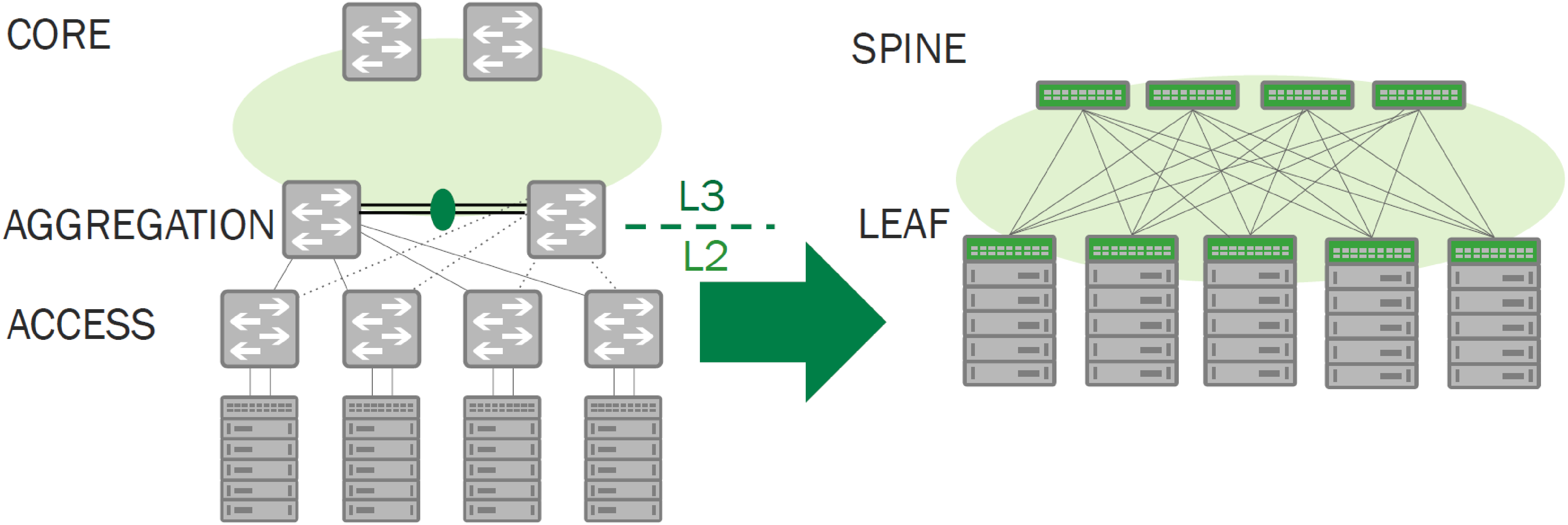
Clos networks have now made their second reappearance in modern data center switching topologies. However, this time, rather than being a ***fabric*** within a single device, the Clos network now manifests itself in the way that the switches are interconnected. Now data center networks are comprised of top-of-rack switches and core switches. The top of rack (**ToR**) switches are the *leaf* switches and they are attached to the core switches which represent the *spine*.

The leaf switches are not connected to each other and spine switches only connect to the leaf switches (or an upstream core device). In this Spine-Leaf architecture, the number of uplinks from the leaf switch equals the number of spine switches. Similarly, the number of downlinks from the spike equal the number of leaf switches. The total number of connections is the number of leaf switches multiplied by the number of spine switches.
In this diagram 8 X 4 = 32 links.

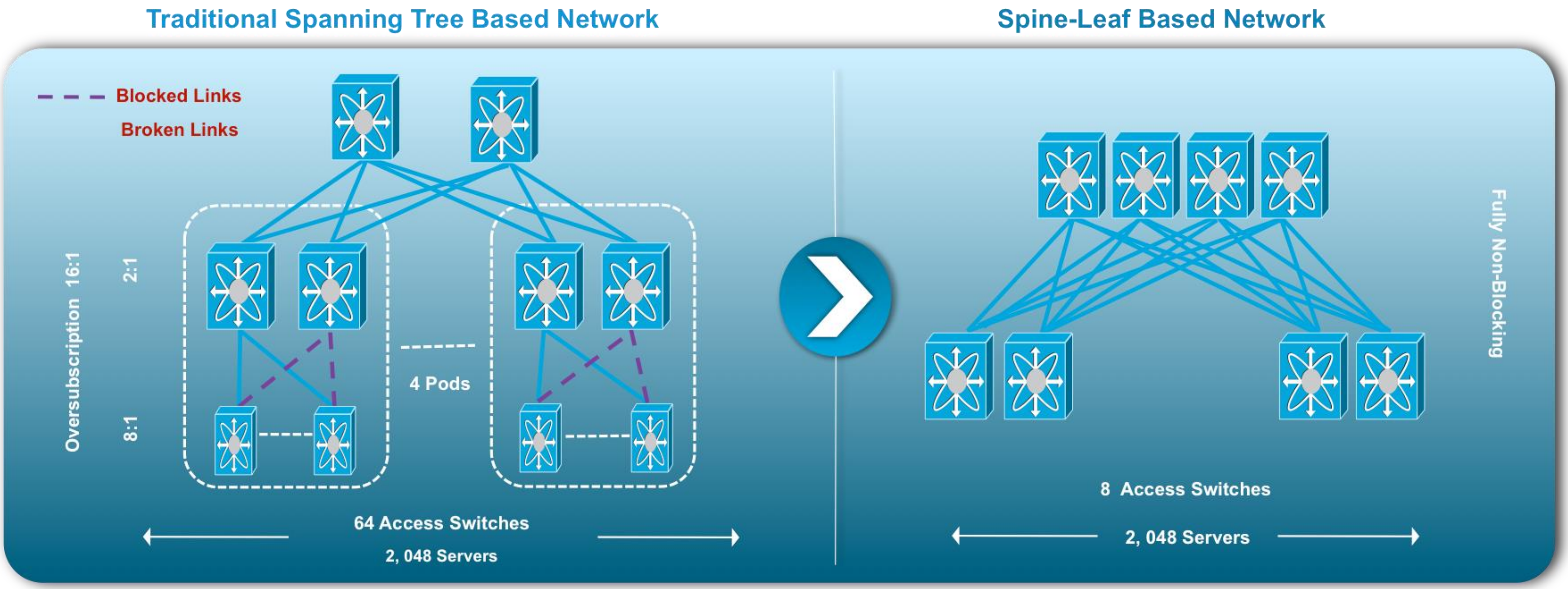
In this Clos topology, every lower-tier switch is connected to each of the top-tier switches in a full-mesh topology. If there isn't any oversubscription taking place between the lower-tier switches and their uplinks, then a non-blocking architecture can be achieved.

Reduced number of protocols

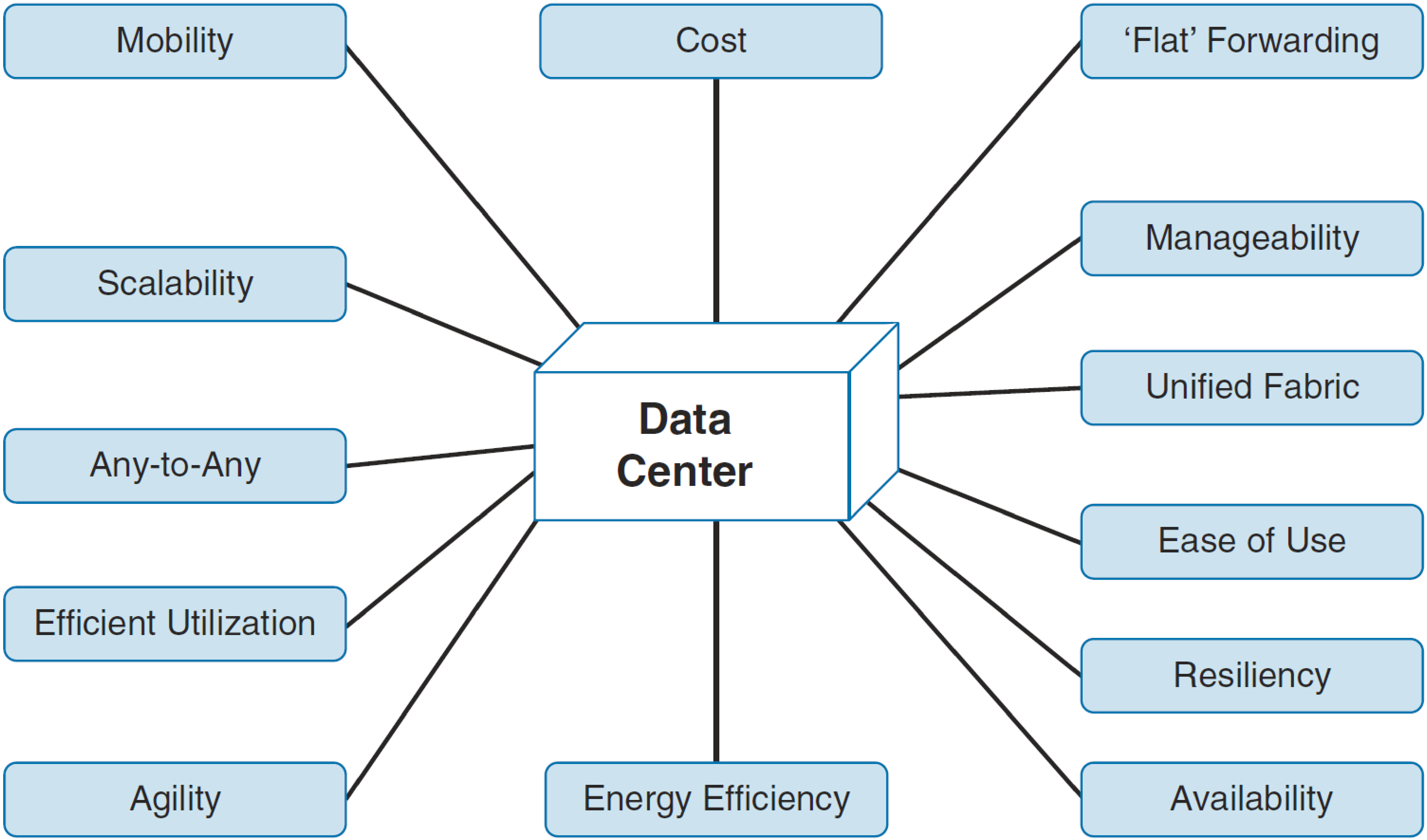
- Single IP protocol is sufficient
- No FHRP, STP, the myriad L2 variants


CORE
AGGREGATION
ACCESS
L3
L2
SPINE
LEAF

How?



Traditional Spanning Tree Based Network
Spine-Leaf Based Network
Blocked Links
Broken Links
Oversubscription 16:1
2:1
8:1
4 Pods
64 Access Switches
2, 048 Servers
Fully Non-Blocking
8 Access Switches
2, 048 Servers


Mobility
Cost
'Flat' Forwarding
Scalability
Manageability
Unified Fabric
Data Center
Any-to-Any
Ease of Use
Efficient Utilization
Resiliency
Agility
Energy Efficiency
Availability

Solution: Overlay Network Technologies

Benefits of Network Overlays

- **Optimized Device Functions**
- **Fabric Scalability and Flexibility**
- **Arbitrary Layer 2 Connectivity without Layer 2 Underlay**
- **Overlapping Addressing**
- **Separation of Roles and Responsibilities**

Overlay Network Use Cases

● **Simplified management:** Use a single point of management to provide network resources for multitenant clouds without the need to change the physical network.

● **Multitenancy at scale:** Provide scalable Layer 2 networks for a multitenant cloud that extends beyond 4000 VLANs. This capability is very important for private and public cloud hosted environments.

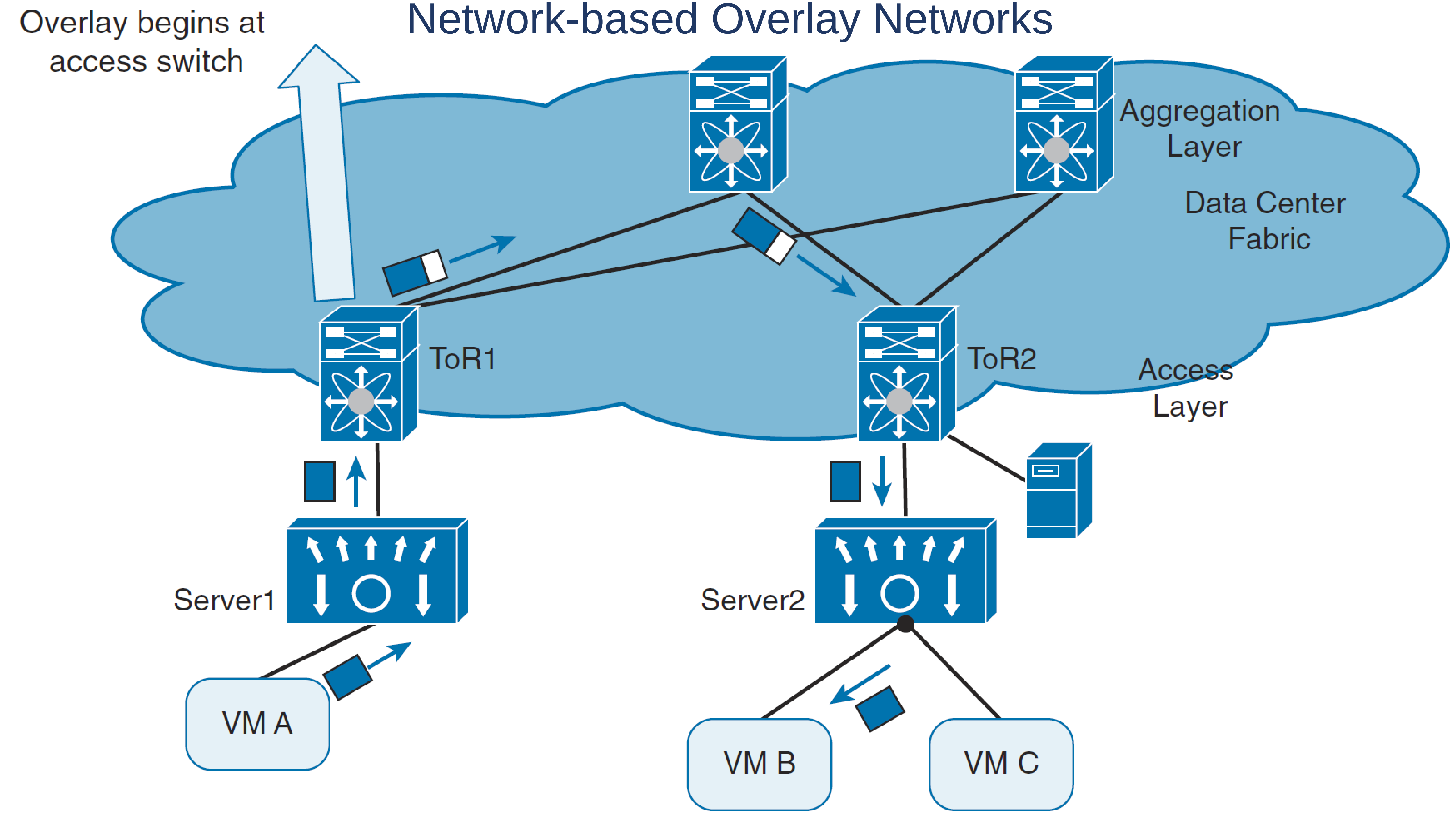
● **Workload-anywhere capability (mobility and reachability):** Optimally use server resources by placing the workload anywhere and moving the workload anywhere in the server farm as needed.

● **Forwarding-topology flexibility:** Add arbitrary forwarding topologies on top of a fixed routed underlay topology.

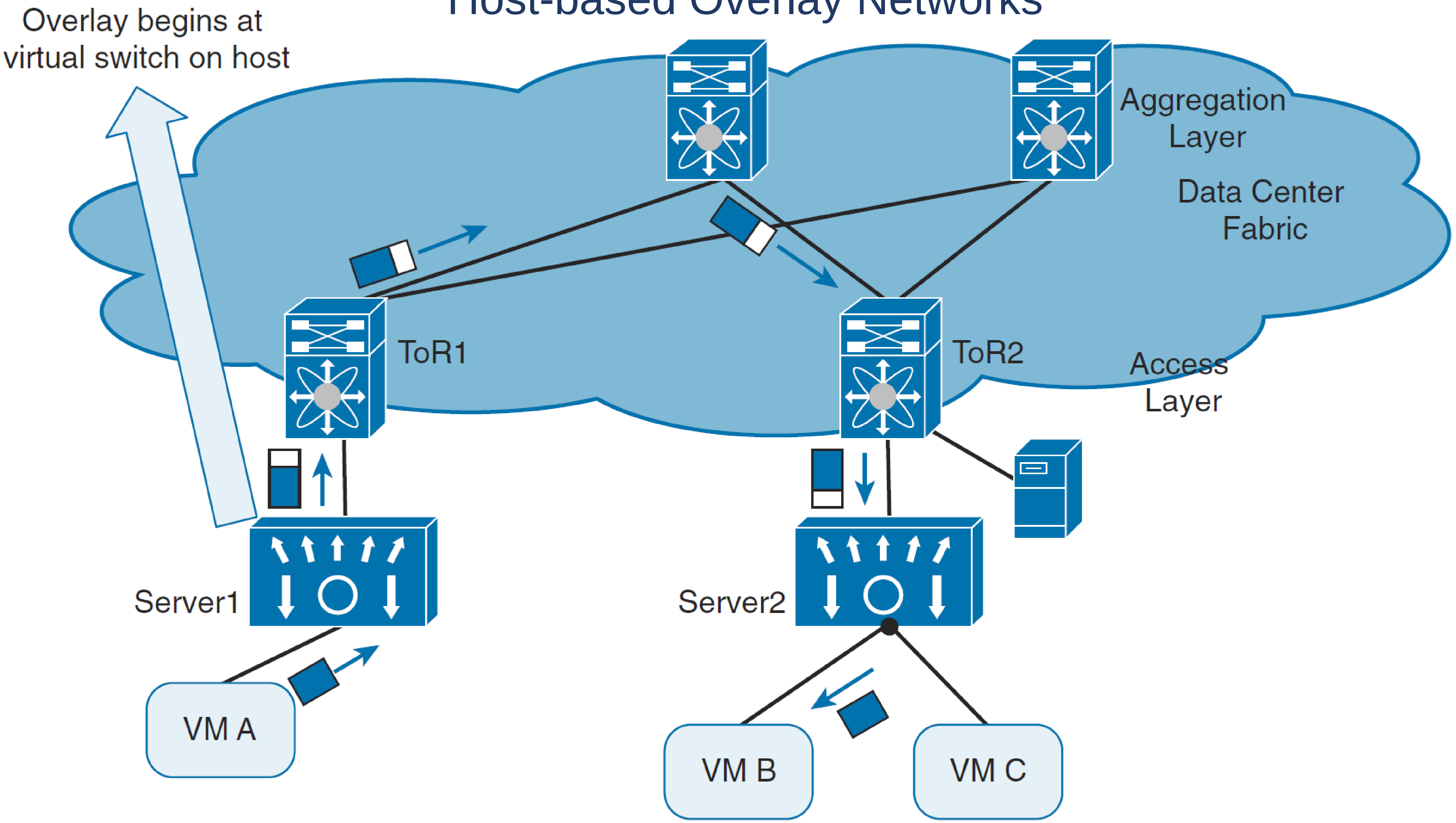
Overlay Networks Classification

- **Network-based** overlay networks
- **Host-based** overlay networks

Network-based Overlay Networks



Host-based Overlay Networks

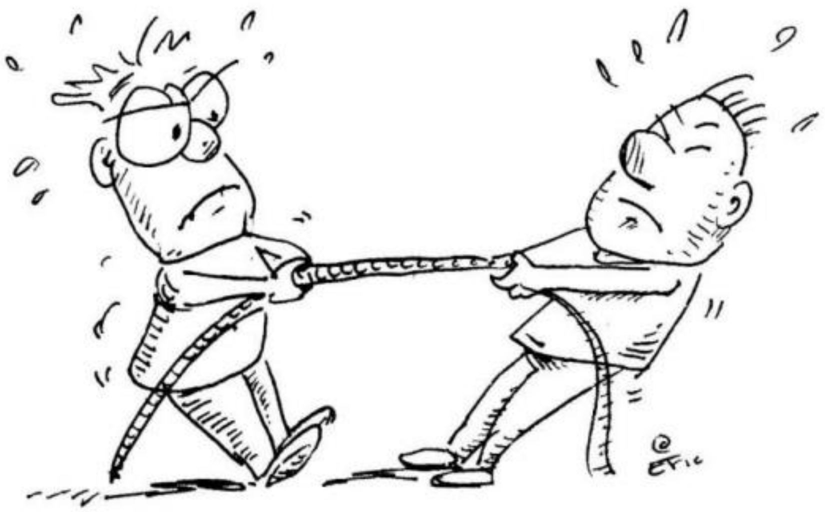


Network-based Overlay Networks

- IEEE 802.1ad Provider Bridge (**Q-in-Q**)
- IEEE 802.1ah Provider Backbone Bridge (**PBB / MAC-in-MAC**)
- IEEE 802.1aq Shortest-Path Bridging (**SPB**)
- IETF Transparent Interconnection of Lots of Links (**TRILL**)
- IETF Multiprotocol Label Switching (**MPLS**) *especially* (**VPLS**)
- IETF BGP MPLS-based Ethernet VPN (**EVPN**)
- Juniper® **QFabric** System
- Brocade® Virtual Cluster Switching (**VCS**)
- Cisco® **FabricPath**
- Cisco® Overlay Transport Virtualization (**OTV**)
- Cisco® Location/Identifier Separation Protocol (**LISP**)

Host-based Overlay Networks

- Network Virtualization Using Generic Routing Encapsulation (**NVGRE**)
- IETF Stateless Transport Tunneling (**STT**)
- IETF Virtual Extensible LAN (**VxLAN**)

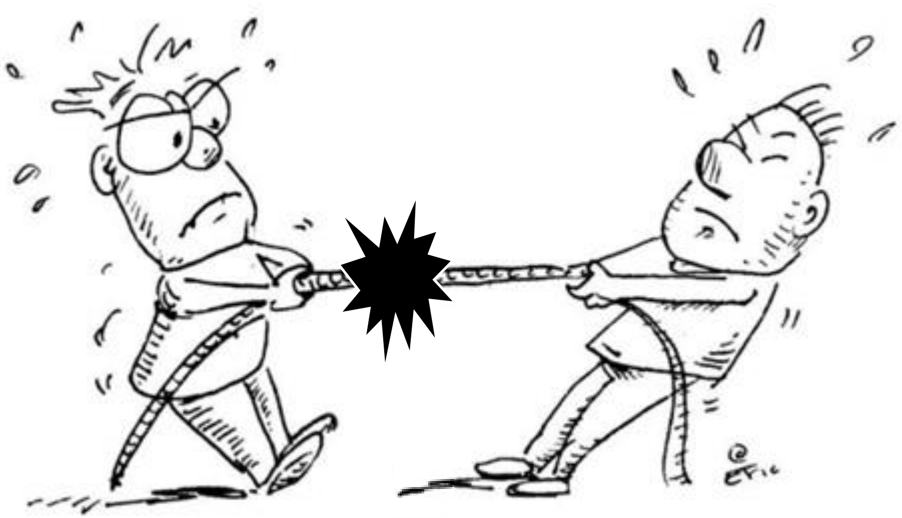


Network-based Overlay Networks

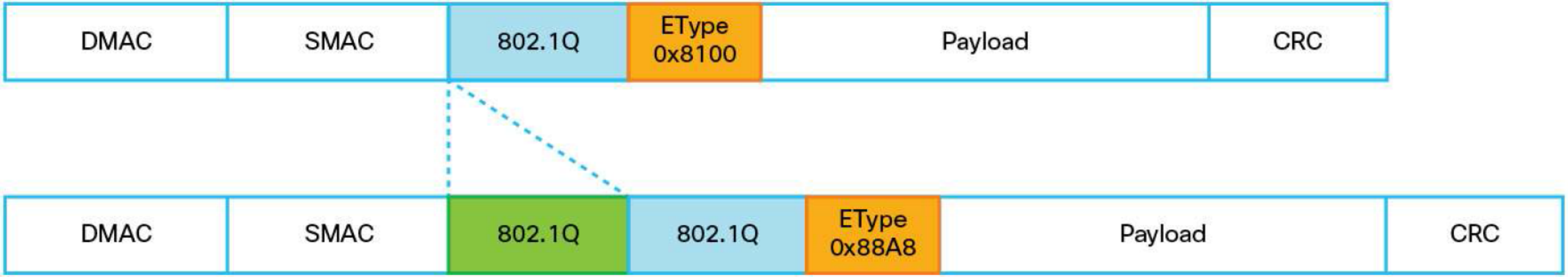
- IEEE 802.1ad Provider Bridge (Q-in-Q)
- IEEE 802.1ah Provider Backbone Bridge (PBB / MAC-in-MAC)
- IEEE 802.1aq Shortest-Path Bridging (SPB)
- IETF Transparent Interconnection of Lots of Links (TRILL)
- IETF Multiprotocol Label Switching (MPLS) especially (VPLS)
- IETF BGP MPLS-based Ethernet VPN (EVPN)
- Juniper® QFabric System
- Brocade® Virtual Cluster Switching (VCS)
- Cisco® FabricPath
- Cisco® Overlay Transport Virtualization (OTV)
- Cisco® Location/Identifier Separation Protocol (LISP)

Host-based Overlay Networks

- Network Virtualization Using Generic Routing Encapsulation (NVGRE)
- IETF Stateless Transport Tunneling (STT)
- IETF Virtual Extensible LAN (VxLAN)

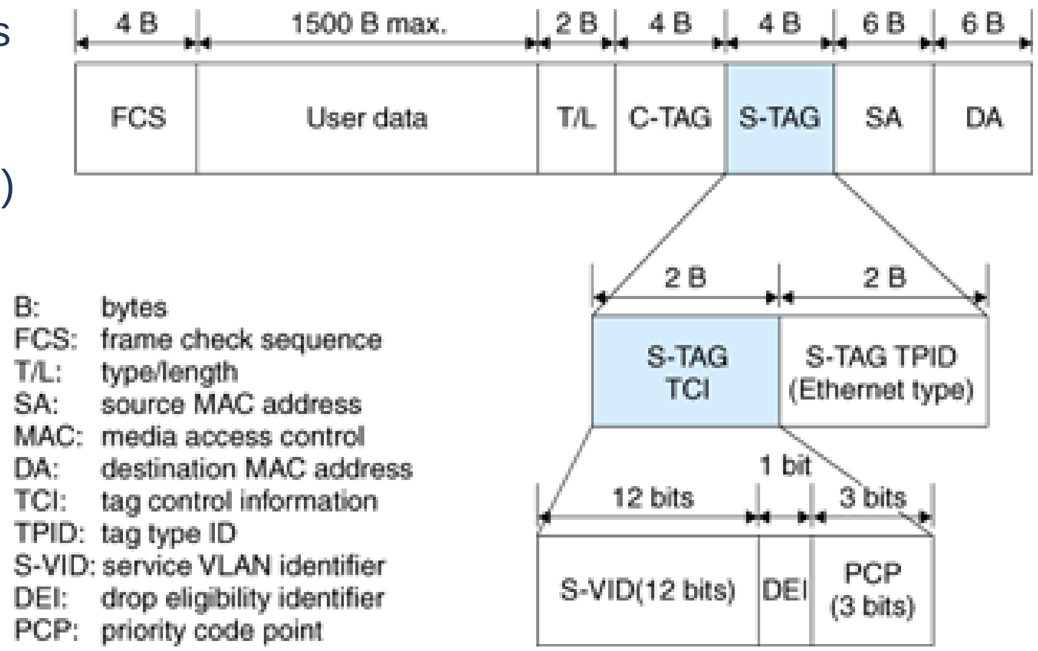


➢ IEEE 802.1ad Provider Bridge (Q-in-Q)

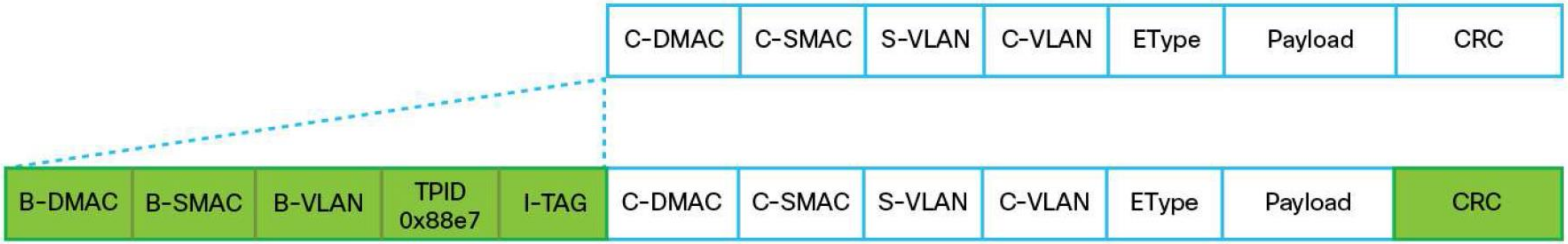


Provider bridging is a tunneling specification that allows multiple VLAN headers to be inserted into a single frame initially used for Metro Ethernet networks.

Stacking the 4-byte VLAN tags (for which 12 bits are allocated for the VLAN ID) allows customers to administer their own VLANs (**C-TAG**)
within a service provider's allocated VLAN (**S-TAG**),
potentially allowing over **16 million** segments with two tags.

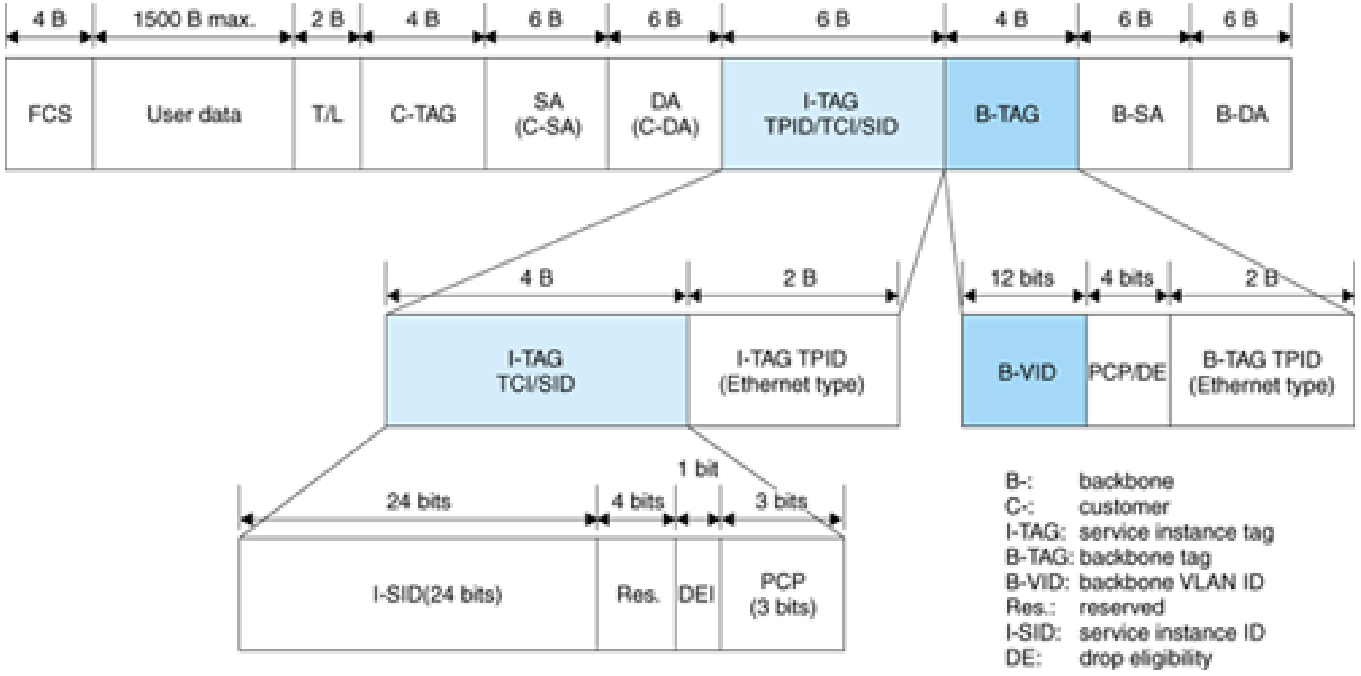


➢ IEEE 802.1ah Provider Backbone Bridge (**PBB / MAC-in-MAC**)

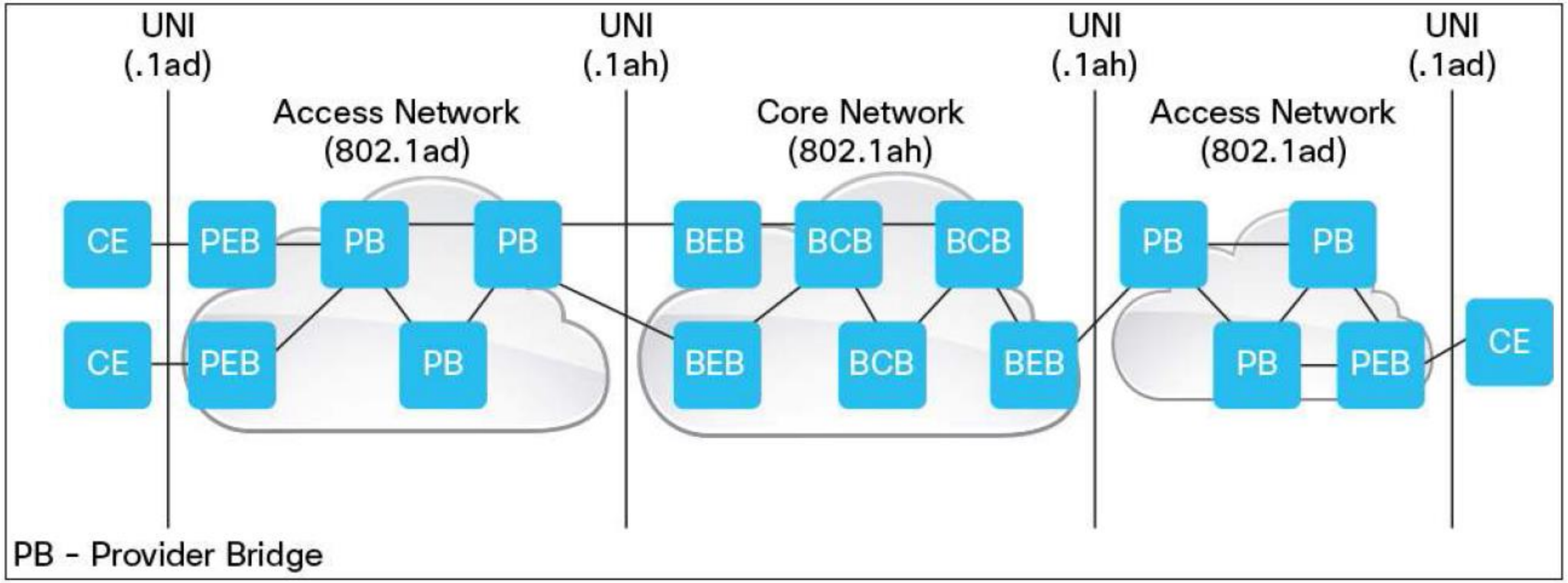


IEEE 802.1ah, or provider backbone bridge (**PBB**), encapsulates end-user or customer traffic in the provider's MAC address header, allowing the backbone edge bridge (**BEB**) to support large numbers of service instances, and at the same time allowing customer MAC addresses to be hidden from the backbone core bridge (**BCB**).

The PBB employs MAC address tunneling encapsulation to tunnel customer Ethernet frames across the PBB network, a backbone VLAN ID (**B-VLAN**) to segregate the backbone into broadcast domains, and a new **24-bit** backbone service instance identifier (**I-SID**) is used to associate a given customer's MAC address frame to the provider's service instance



➢ IEEE 802.1ah Provider Backbone Bridge (PBB / MAC-in-MAC)



In addition to capabilities specified in IEEE 802.1ad, **PBB** can hide customer MAC addresses from the provider network through the additional **MAC-in-MAC** encapsulation; however, it faces challenges with features that many provider networks want such as multipathing, traffic engineering, and carrier-class resiliency because it still relies on *Spanning Tree Protocols* for loop avoidance.

➢ IEEE 802.1aq Shortest-Path Bridging (SPB)

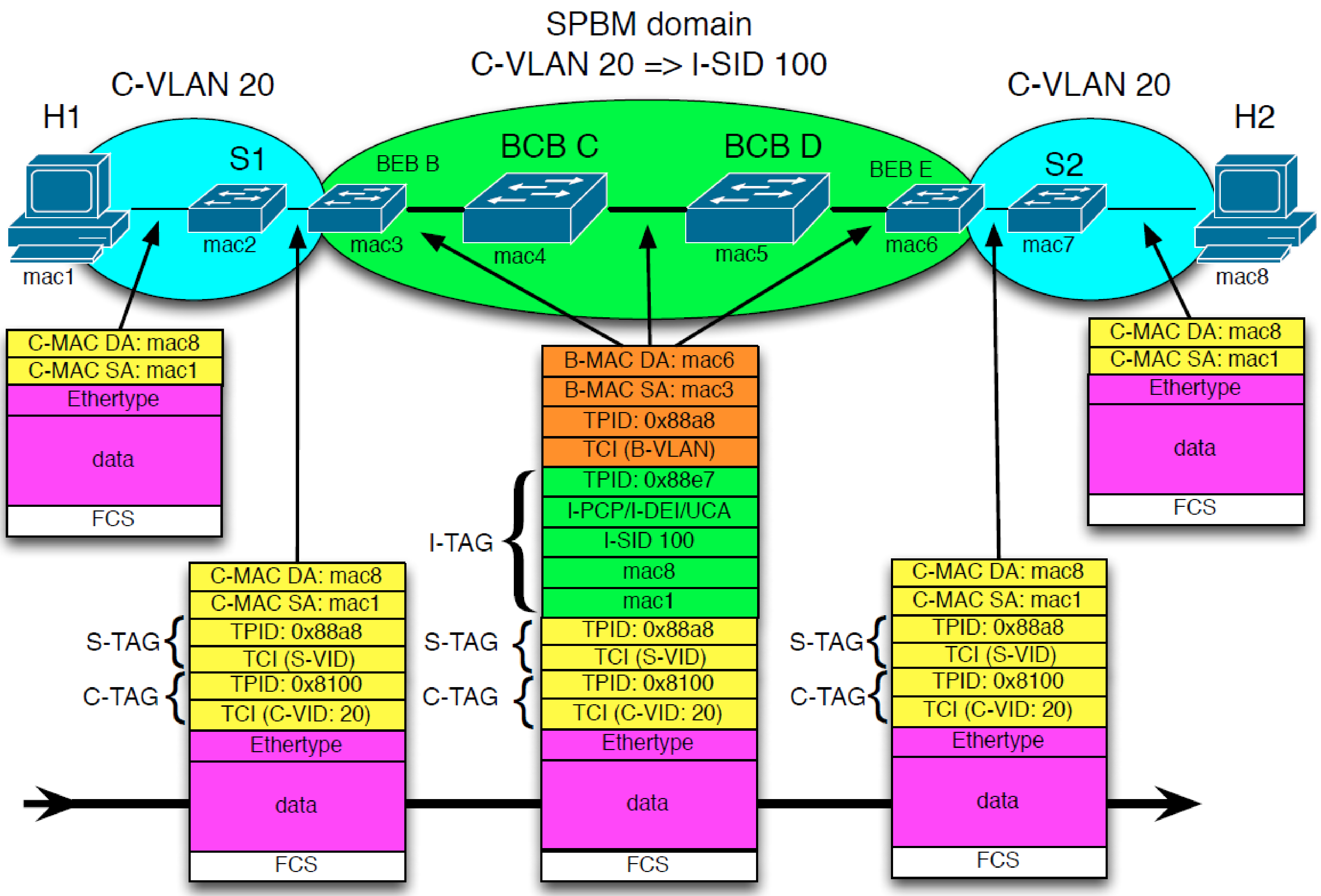
Shortest-Path Bridging (**SPB**) is defined in **IEEE 802.1aq** and is targeted as a replacement for Spanning Tree Protocol, which blocks traffic on all but one alternative path. It is a Layer 2 multipathing technology that allows all paths to be active with multiple equal-cost paths, providing fast convergence times, and it can support larger segment spaces to accommodate scalable virtual networks. **SPB** uses extensions to **IS-IS** as a link-state routing protocol to calculate the shortest-path tree (**SPT**) and discover the topology of the network.

SPBM specifically uses **IEEE 802.1ah** provider backbone bridge frame formats for data-plane encapsulation. Unlike SPBV, SPBM uses **I-SIDs** (**I-TAG**) for service delineation, but for load balancing VLANs can also be used. For forwarding, SPBM uses a combination of one or more **B-VIDs**, known as backbone-MAC (**B-MAC**) addresses that have been advertised in IS-IS. Additionally, in SPBM edge MAC addresses are never learned or looked up in the core of a IEEE 802.1aq network**; B-MAC** addresses are distributed through the control plane through IS-IS, thus eliminating B-MAC address learning in PBB.

IEEE L2 Ethernet evolution

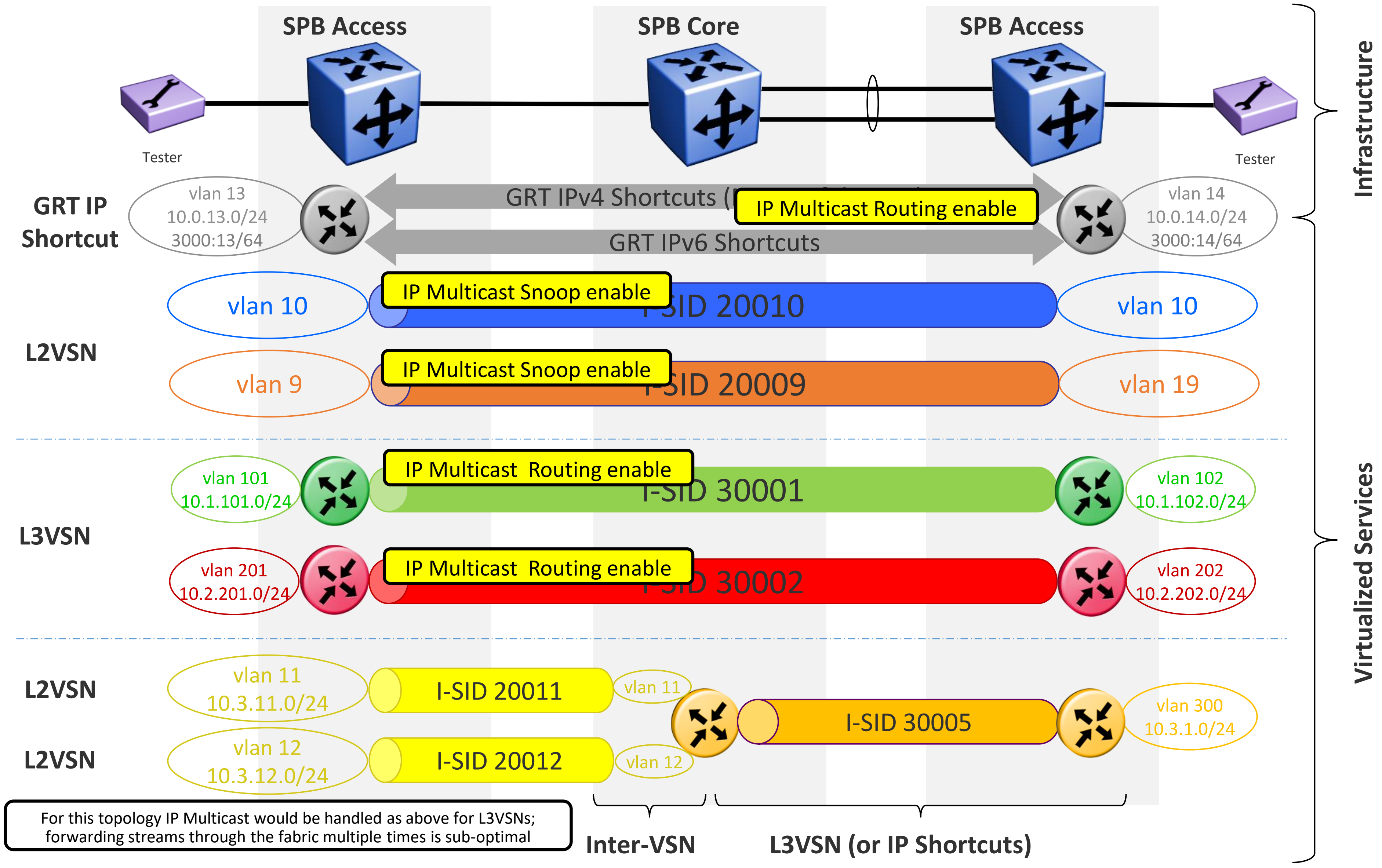
Standard	Year	Name	Loopfree topology by:	Service ID's	Provisioning	Virtualization of
IEEE 802.1Q	1998	Virtual Lans (VLAN Tagging)	Spanning Tree SMLT	4096	Edge and Core	Layer 2
IEEE 802.1ad	2005	Provider Bridging (QinQ)	Spanning Tree SMLT	4096x4096	Edge and Core	Layer 2
IEEE 802.1ah	2008	Provider Backbone Bridging (MacInMac)	Spanning Tree SMLT	16 Mil.	Edge and Core	Layer 2
IEEE 802.1aq	2011	Shortest Path Bridging (SPB)	Link-State-Protocol (IS-IS)	16 Mil.	Only Service Access Points	IEEE: Layer 2 IETF draft: Layer 3 Unicast & Multicast

- IEEE 802.1aq Shortest-Path Bridging (SPB)

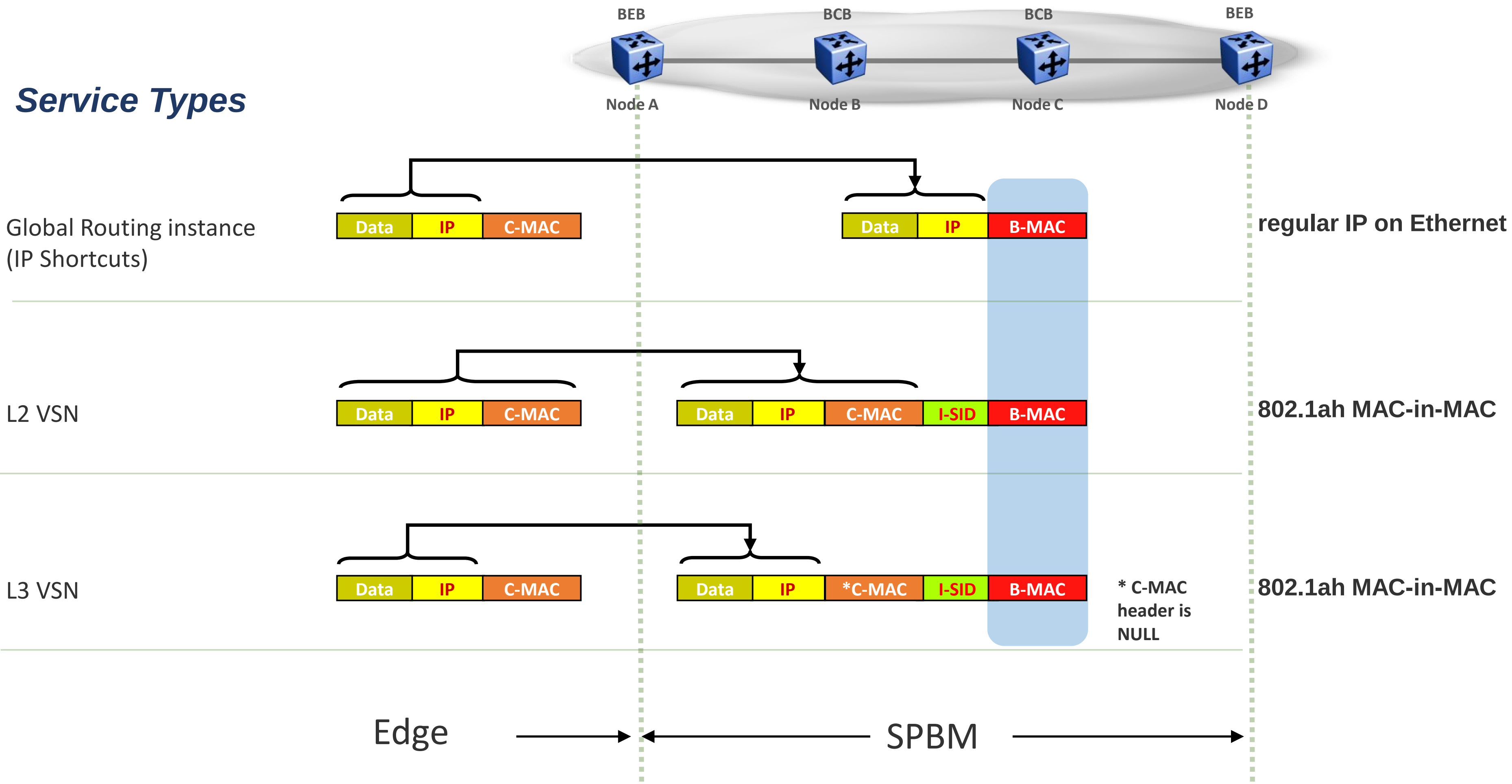


➢ IEEE 802.1aq Shortest-Path Bridging (SPB)

Summary of SPB Services



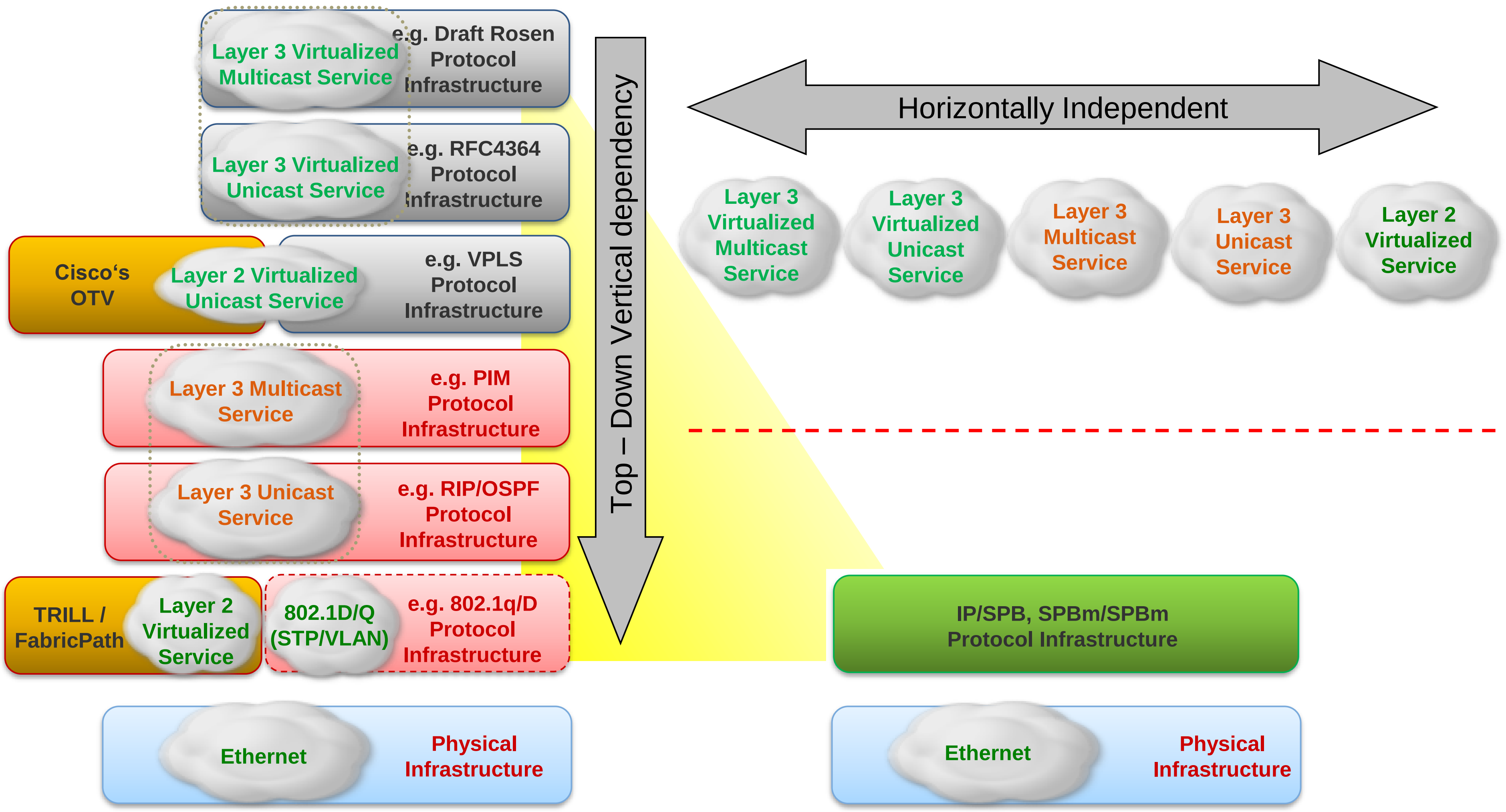
SPB Service Type Encapsulations



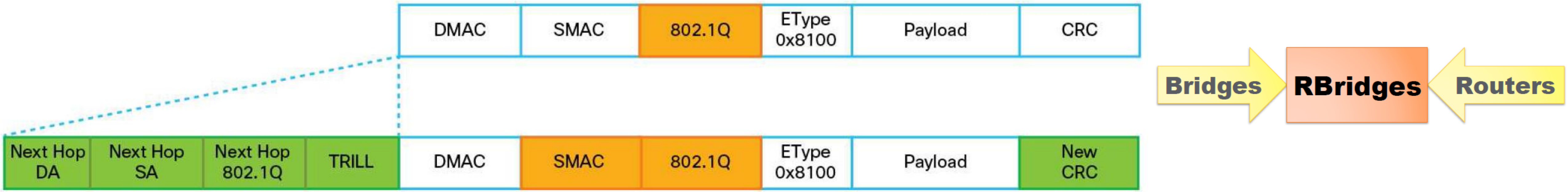
SPB vs Others

➢ Traditional Protocol Stack

➢ SPB's simplicity



IETF Transparent Interconnection of Lots of Links (TRILL)



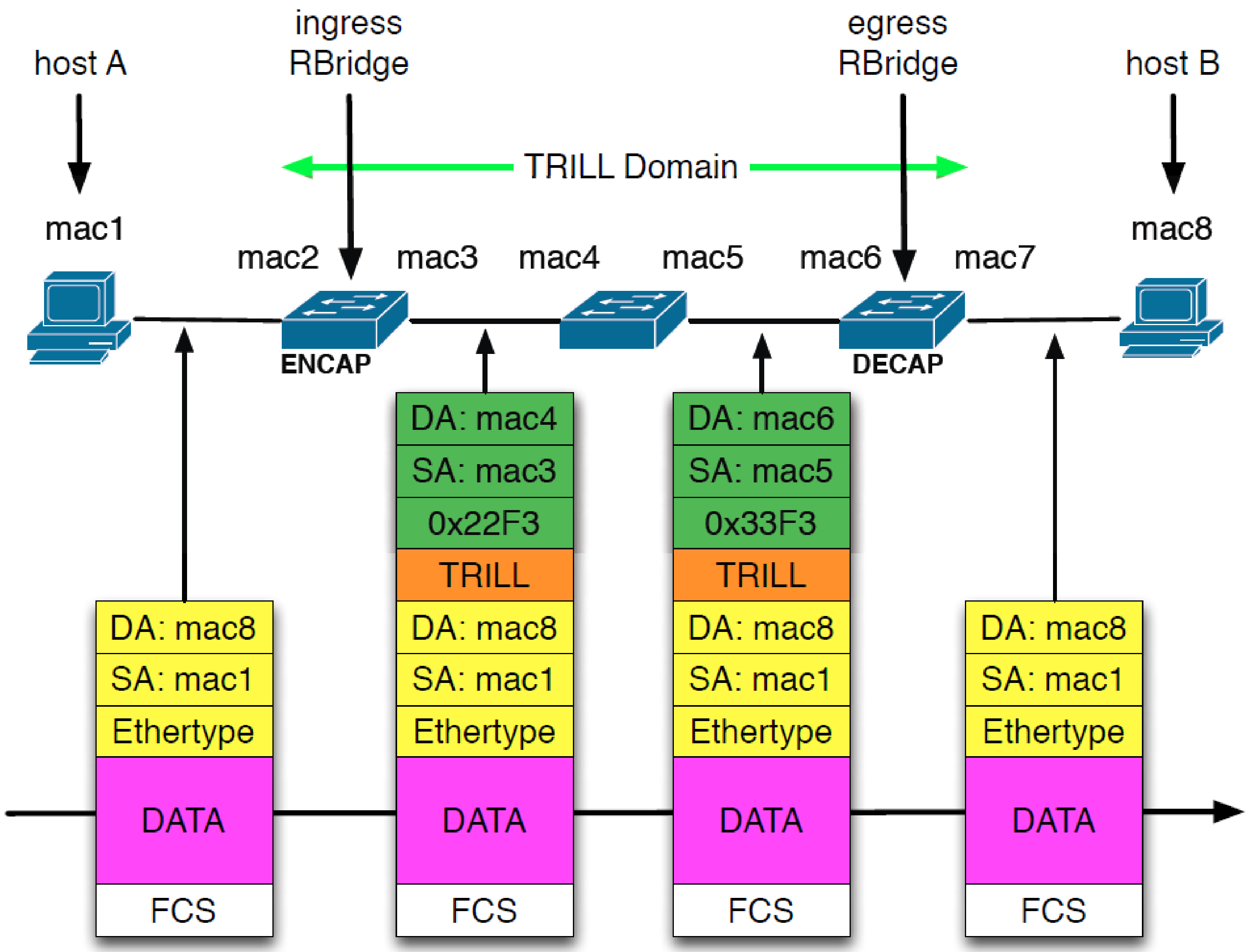
IETF Transparent Interconnection of Lots of Links, or **TRILL**, is also a Layer 2 multipathing technology. It is implemented by devices called routing bridges (**RBridges**) and adds a new encapsulation to the frame. However, this encapsulation is implemented in such a way that it is compatible and can incrementally replace existing IEEE 802.3 Ethernet bridges. With the encapsulation of a new Ethernet MAC address header, the original MAC address header is left unmodified and hence can pass through intermediate Ethernet bridges.

RBridges are similar to routers in that when a TRILL frame requires forwarding by an intermediate RBridge, the outer Layer 2 header is replaced at each RBridge hop with an appropriate Layer 2 header for the next hop, and a **hop count** in the TRILL header is decremented. Despite this, the original encapsulated frame is preserved, including any VLAN tags.

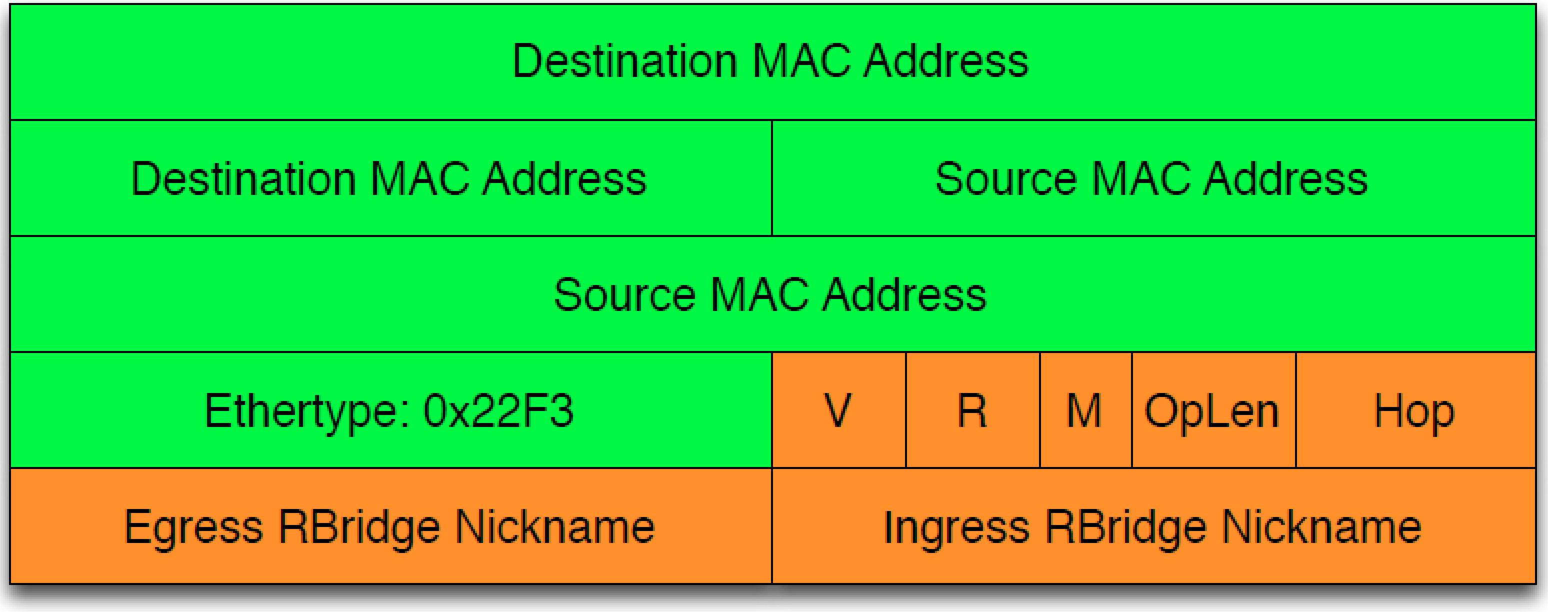
Similar to SPB, TRILL uses extensions to IS-IS as its routing protocol. The link-state protocol provides enough information between the RBridges so that they can compute pair-wise optimal paths for unicast traffic and calculate distribution trees for multidestination frames.

As with Cisco FabricPath, TRILL currently *has no provision* for extending the segment space beyond **4000** segments.

- IETF Transparent Interconnection of Lots of Links (TRILL)

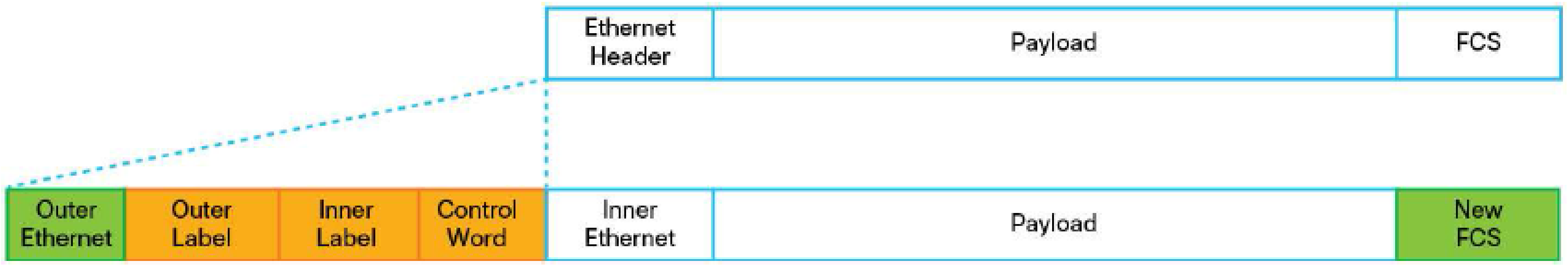


- IETF Transparent Interconnection of Lots of Links (TRILL)



V	Version	2 bit
R	Reserved	2 bit
M	Multi-Destination	1 bit
OpLen	Option Length	5 bit
Hop	Hop Count	6 bit
Nickname	Nickname	16 bit

➢ IETF Multiprotocol Label Switching (MPLS) *especially* (VPLS)



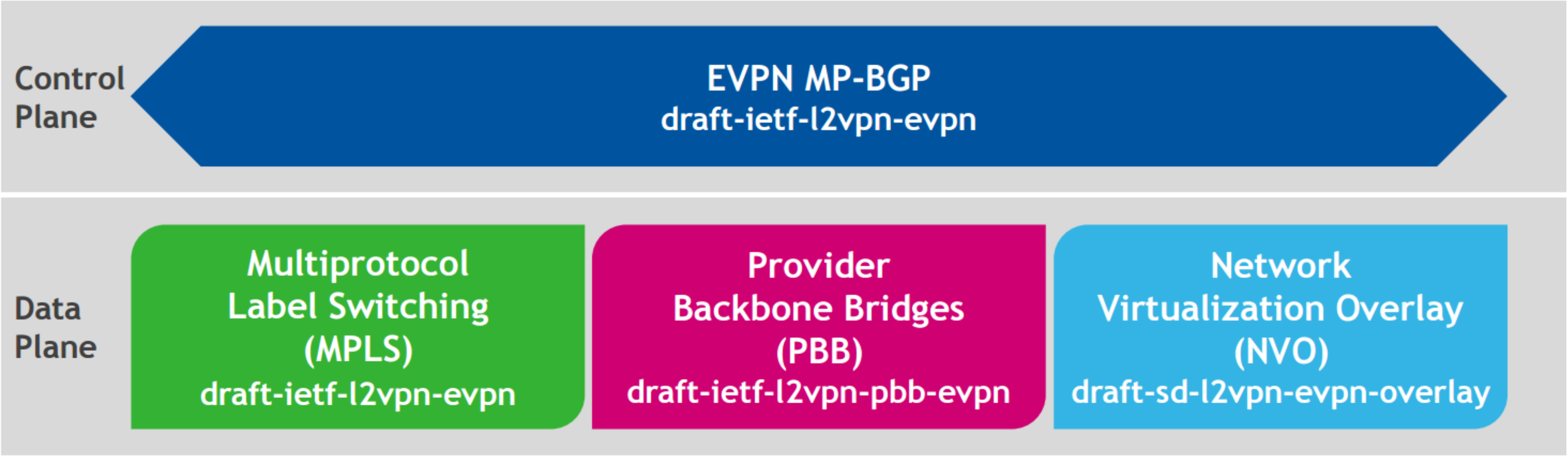
Multiprotocol Label Switching (**MPLS**) has been used extensively in service provider environments and even certain enterprise environments. **VPLS**, as defined in RFC 4761 and RFC 4762, allows the creation of pseudowires that emulate LAN segments (for an Ethernet switch) for a given set of users, and that are fully capable of learning and forwarding Ethernet MAC addresses that are closed to that set of users. **VPLS** allows any-to-any (multipoint) connectivity and is typically deployed in a provider network to *emulate* a switch or a bridge to connect customer LAN segments to create a single bridged LAN.

For label distribution, discovery, and signaling, two control-plane methods have been widely adopted throughout the industry. One of the use of the Border Gateway Protocol (**BGP**) as defined in RFC 4761 (Kireeti Kompella and Yakov Rekhter), and the other is the use of the Label Distribution Protocol (**LDP**) as defined in RFC 4762 (Vach Kompella and Marc Lasserre).

MPLS *changed* routing to be a single route lookup at the edge and Label Switched Path (**LSP**) through the core. But Forwarding performed by *swapping* a MPLS label and the MAC header at each hop. Requires more processing and intelligence causing duplicate info in multiple protocol tables. Very heavy overlay model that requires a complex cocktail mix of protocols to function. The end result is an environment that is very complex to provision, maintain and troubleshoot.

➢ IETF BGP MPLS-based Ethernet VPN (EVPN)

Ethernet VPN introduces the concept of **BGP MAC routing**. It uses MP-BGP for learning MAC addresses between provider edges. Learning between the PE and the CE is still done in the data plane. The BGP control plane has the advantage of scalability and flexibility for MAC routing, just as it does for IP routing. EVPN provides separation between the data plane and the control plane, which allows it to use different encapsulation mechanisms in the data plane while maintaining the same control plane.



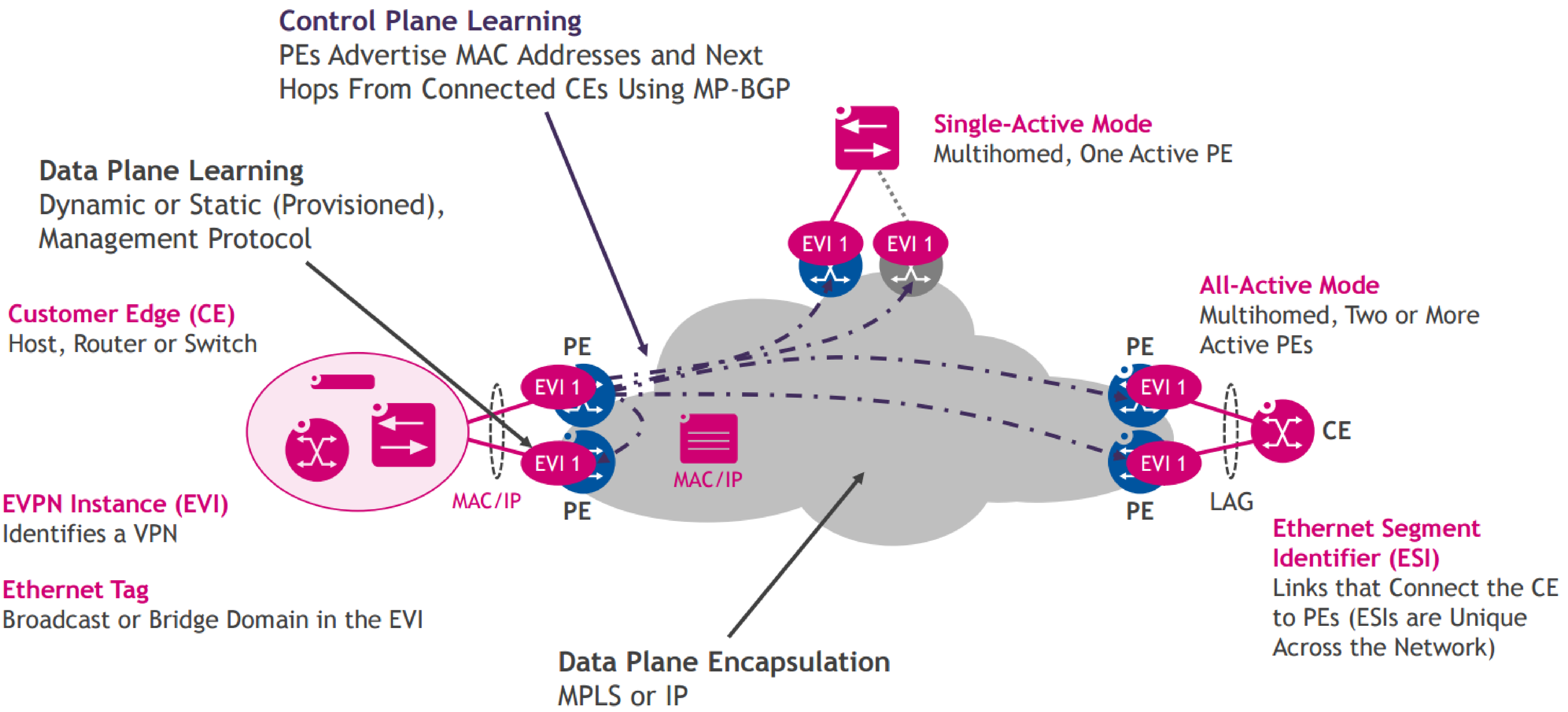
- EVPN over MPLS for E-LAN services
- All-active multihoming for VPWS
- RSVP-TE or LDP MPLS protocols

- EVPN with PBB PE functionality for scaling very large networks over MPLS
- All-active multihoming for PBB-VPLS

- EVPN over NVO tunnels (VXLAN, NVGRE, MPLSoGRE) for data center fabric encapsulations
- Provides Layer 2 and Layer 3 DCI and overlays over simple IP networks

➢ IETF BGP MPLS-based Ethernet VPN (**EVPN**)

EVPN CONCEPTS OVERVIEW



➢ IETF BGP MPLS-based Ethernet VPN (**EVPN**)

EVPN CONTROL PLANE LEARNING WITH MP-BGP

• Brings proven and inherent BGP control plane scalability to MAC routes
- Consistent signaled FDB in any size network instead of flooding
- Even more scalability and hierarchy with route reflectors

• BGP advertises MACs and IPs for next hop resolution with EVPN NLRI
- AFI = 25 (L2VPN) and SAFI = 70 (EVPN)
- Fully supports IPv4 and IPv6 in the control and data plane

• Offers greater control over MAC learning
- What is signaled, from where and to whom
- Ability to apply MAC learning policies

• Maintains virtualization and isolation of EVPN instances

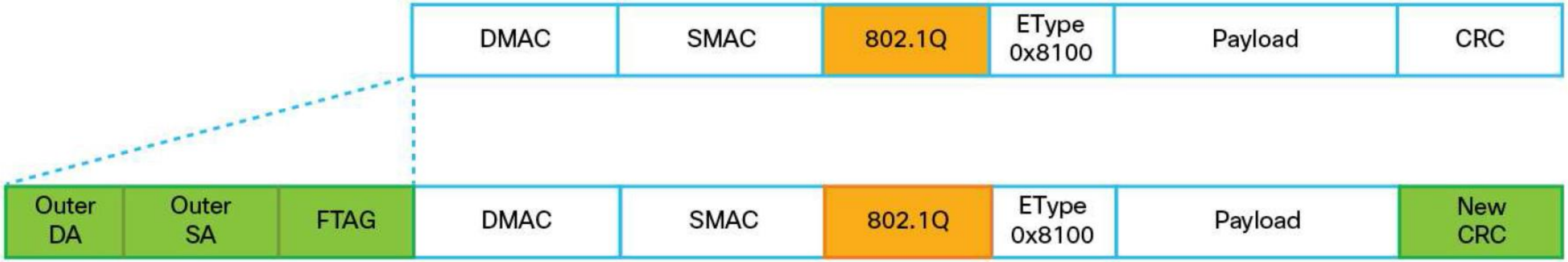
• Enables traffic load balancing for multihomed CEs with ECMP MAC routes

Route Distinguisher (8 octets)
Ethernet Segment Identifier (10 octets)
Ethernet Tag ID (4 octets)
MAC Address Length (1 octet)
MAC Address (6 octets)
IP Address Length (1 octet)
IP Address (0 or 4 or 16 octets)
MPLS Label1 (3 octets)
MPLS Label2 (0 or 3 octets)

MAC Advertisement Route
(Light Blue Fields are Not Used in all Data Planes)

EVPN is technically just another address family in Multi Protocol (MP) BGP. This new address family allows MAC addresses to be treated as routes in the BGP table. The entry can contain just a MAC address or an IP address + MAC address (ARP entry). This can all be combined with or without a VLAN tag as well.

➢ Cisco® FabricPath

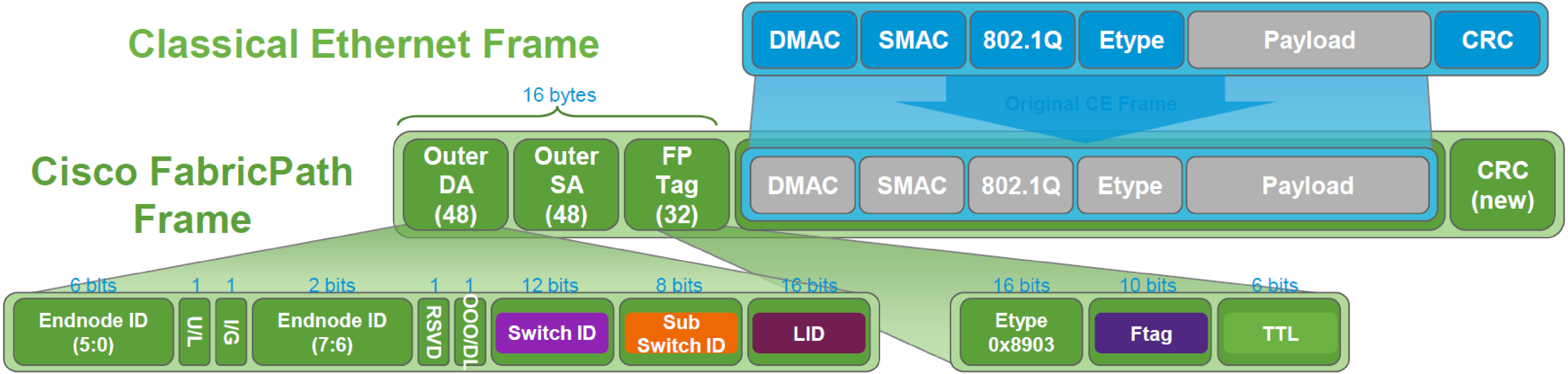


Cisco **FabricPath** switching allows multipath networking at Layer 2 and encapsulates the entire Layer 2 frame with a new Cisco FabricPath header. Cisco FabricPath links are point to point, and devices encapsulate frames at the ingress edge port of the Cisco FabricPath network and de-encapsulate frames on the egress edge port of the Cisco FabricPath network. This new encapsulation allows the core of the Cisco FabricPath network to be hidden (through overlay technology) from the host state information, reducing the scaling requirements of Cisco FabricPath core devices.

All nodes on the Cisco FabricPath network need to support Cisco FabricPath to look up and forward the frame throughout the rest of the network.

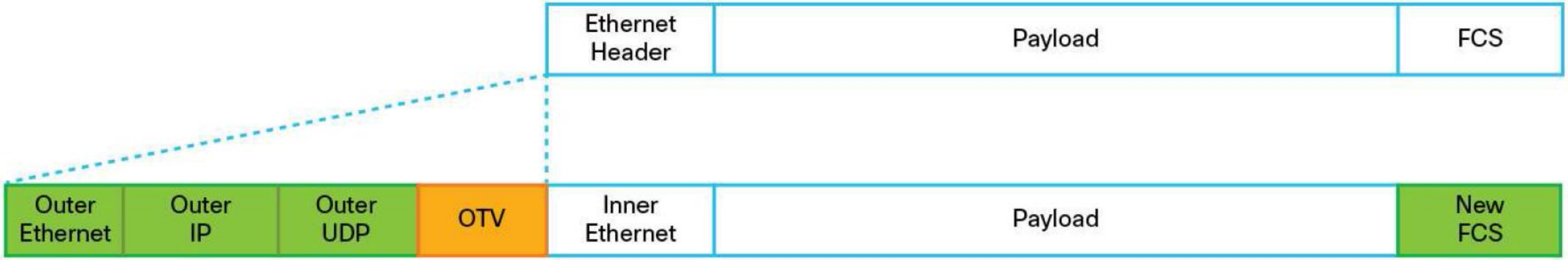
Cisco FabricPath also introduces an additional tag called the forwarding tag (**FTAG**), which can be used to describe and segment multiple forwarding topologies, by mapping Ethernet VLANs to a given topology at the Cisco FabricPath edge. The frame is encapsulated with the appropriate FTAG as it is forwarded throughout the Cisco FabricPath network, where forwarding is constrained to a given topology. Although the Cisco FabricPath does not support extension of the segment space beyond 4000 VLANs.

Cisco® FabricPath



- **Switch ID** – Unique number identifying each FabricPath switch
- **Sub-Switch ID** – Identifies devices/hosts connected via VPC+
- **LID** – Local ID, identifies the destination or source interface
- **Ftag** (Forwarding tag) – Unique number identifying topology and/or distribution tree
- **TTL** – Decremented at each switch hop to prevent frames looping infinitely

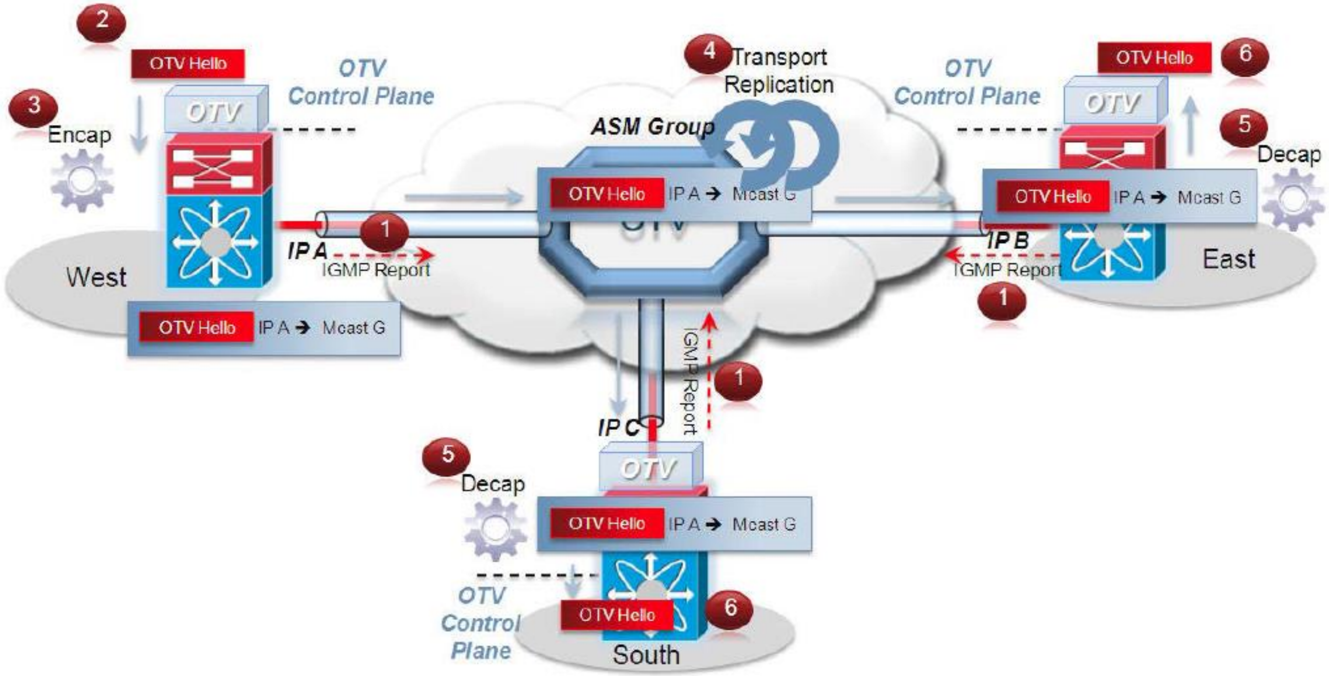
➢ Cisco® Overlay Transport Virtualization (OTV)



Cisco Overlay Transport Virtualization (**OTV**) is a **Layer 2-over-Layer 3** encapsulation "**MAC-in-IP**" technology that is designed to extend the reach of Layer 2 domains across data center pods, domains, and sites. It uses stateless tunnels to encapsulate Layer 2 frames in the IP header and does not require the creation or maintenance of fixed stateful tunnels. OTV encapsulates the entire Ethernet frame in an IP and User Datagram Protocol (IP/UDP) header, so that the provider or core network is transparent to the services offered by OTV.

OTV introduces the concept of "MAC routing," which means a control plane protocol is used to exchange MAC reachability information between network devices providing LAN extension functionality.

➢ Cisco® Overlay Transport Virtualization (OTV)

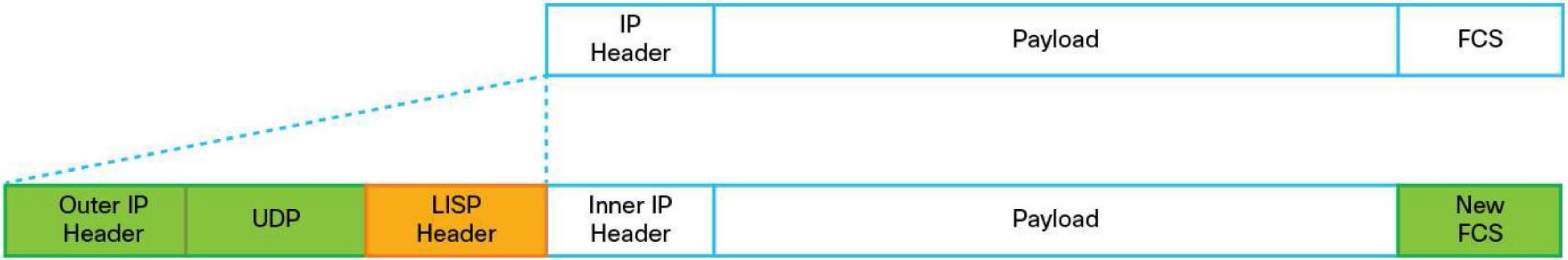


OTV claims to be better than VPLS, but this could be argued. To begin with, VPLS is positioned as provider edge technology and OTV is customer-edge technology. Next, the following list captures similarities and differences between the two technologies:

•The same logical full-mesh of signaling is used in the core. IS-IS it outlined in the patent document, but any other protocol could be obviously used here, e.g. LDP or BGP. Even the patent document mentions that. What was the reason to re-inventing the wheel? The answer could be “SPB” as we see in the following section.

•OTV runs over native IP, and does not require underlying MPLS. Like we said before, it was possible to simple change VPLS transport to any IP tunneling technique instead of coming with a new technology. By missing MPLS, OTV loses the important ability to signal optimal path selection in provider networks at the PE edge.

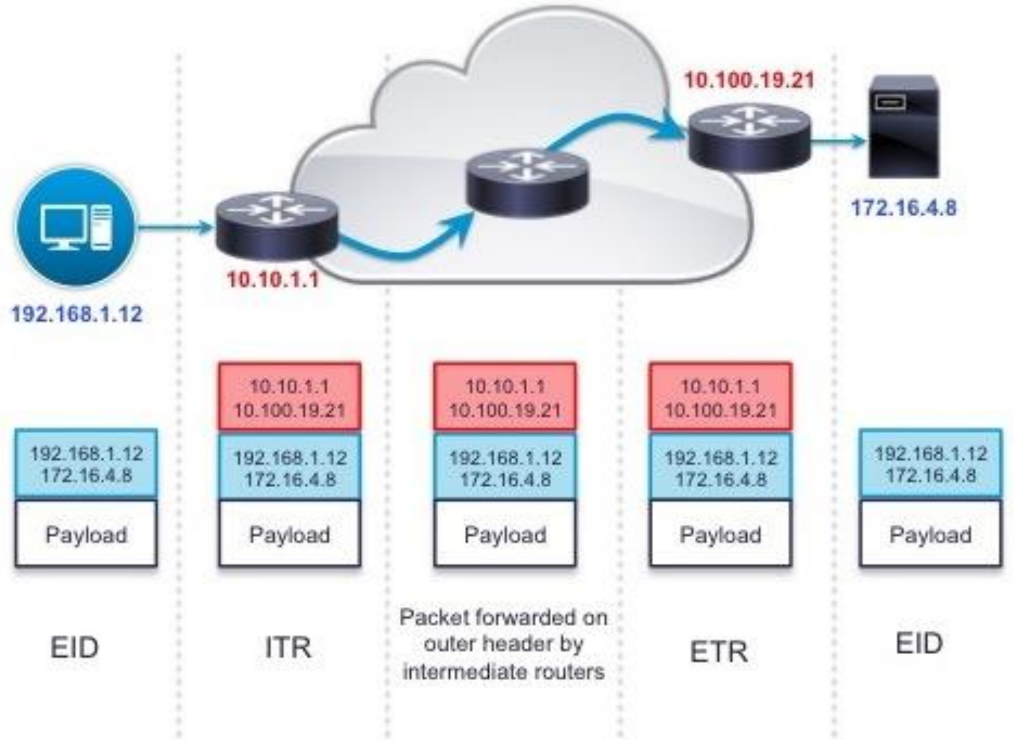
➢ Cisco® Location/Identifier Separation Protocol (LISP)



The Cisco Location/Identifier Separation Protocol, or **LISP**, is designed to address the challenges of using a single address field for both device identification and topology location. This challenge is evident in modern data centers, where the mobility of endpoints should not result in a change in the end-host addressing, but simply the location of the end host.

LISP addresses the problem by uniquely identifying two different number sets: routing locators (RLOCs), which describe the topology and location of attachment points and hence are used to forward traffic, and endpoint identifiers (EIDs), which are used to address end hosts separate from the topology of the network

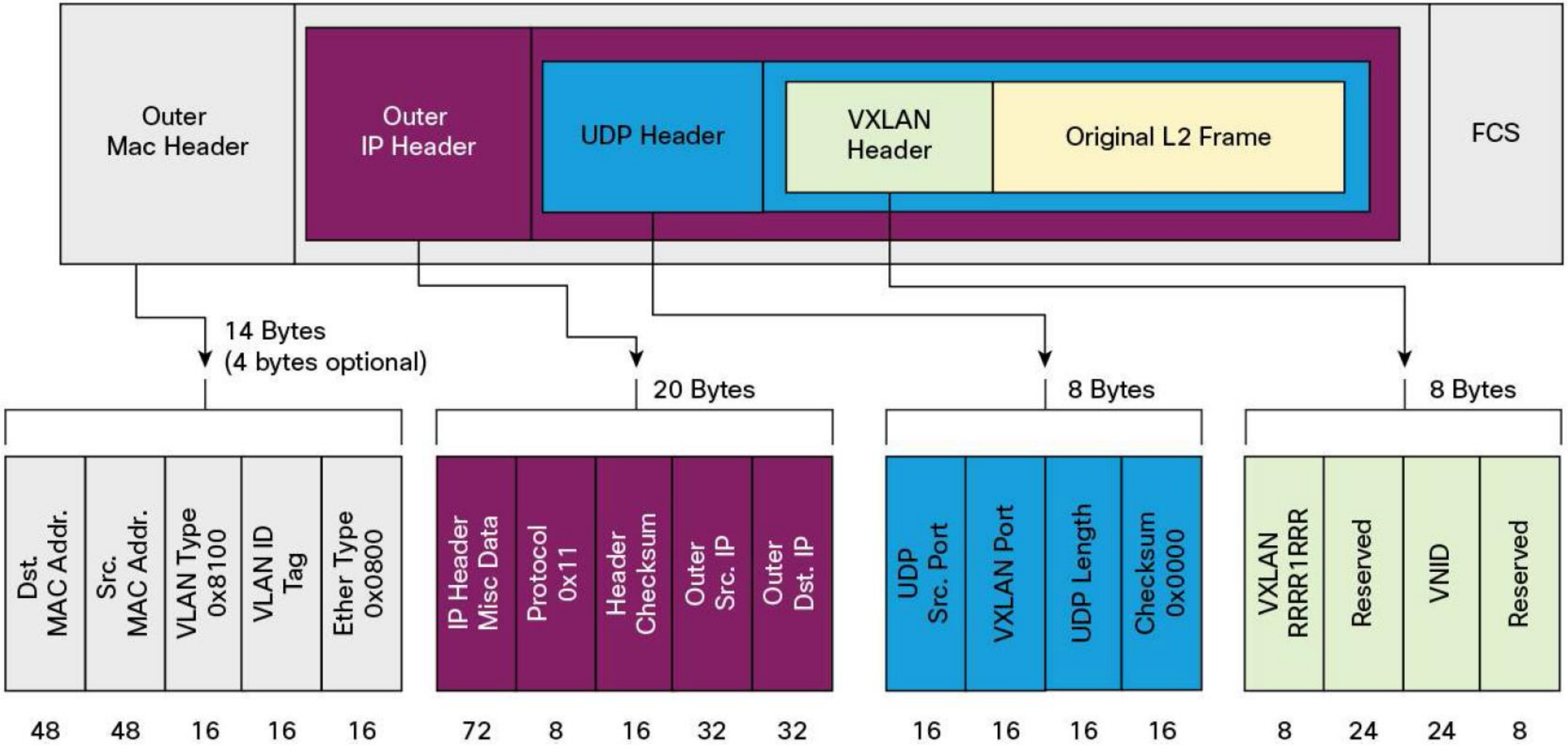
➢ Cisco® Location/Identifier Separation Protocol (LISP)



LISP defines the capabilities and functions of routers and switches to exchange information to map EIDs to RLOCs, as well as a mechanism that allows LISP routers to encapsulate IP-based EIDs for forwarding across an IP fabric or the Internet using RLOC addresses. The devices performing the encapsulation and de-encapsulation of LISP headers are called ingress tunnel routers (ITRs) and egress tunnel routers (ETRs), respectively. LISP is currently defined as a Layer 3 overlay scheme over a Layer 3 network, and it encompasses IPv4 and IPv6 for both the underlay and the overlay.

Similar to other encapsulation schemes described previously, LISP provides a mechanism to help ensure virtual segment isolation through the addition of a 24-bit instance ID field in the LISP header, allowing more than 16 million virtual segments to be instantiated; this mechanism is set by the ITR.

➢ IETF Virtual Extensible LAN (VxLAN)

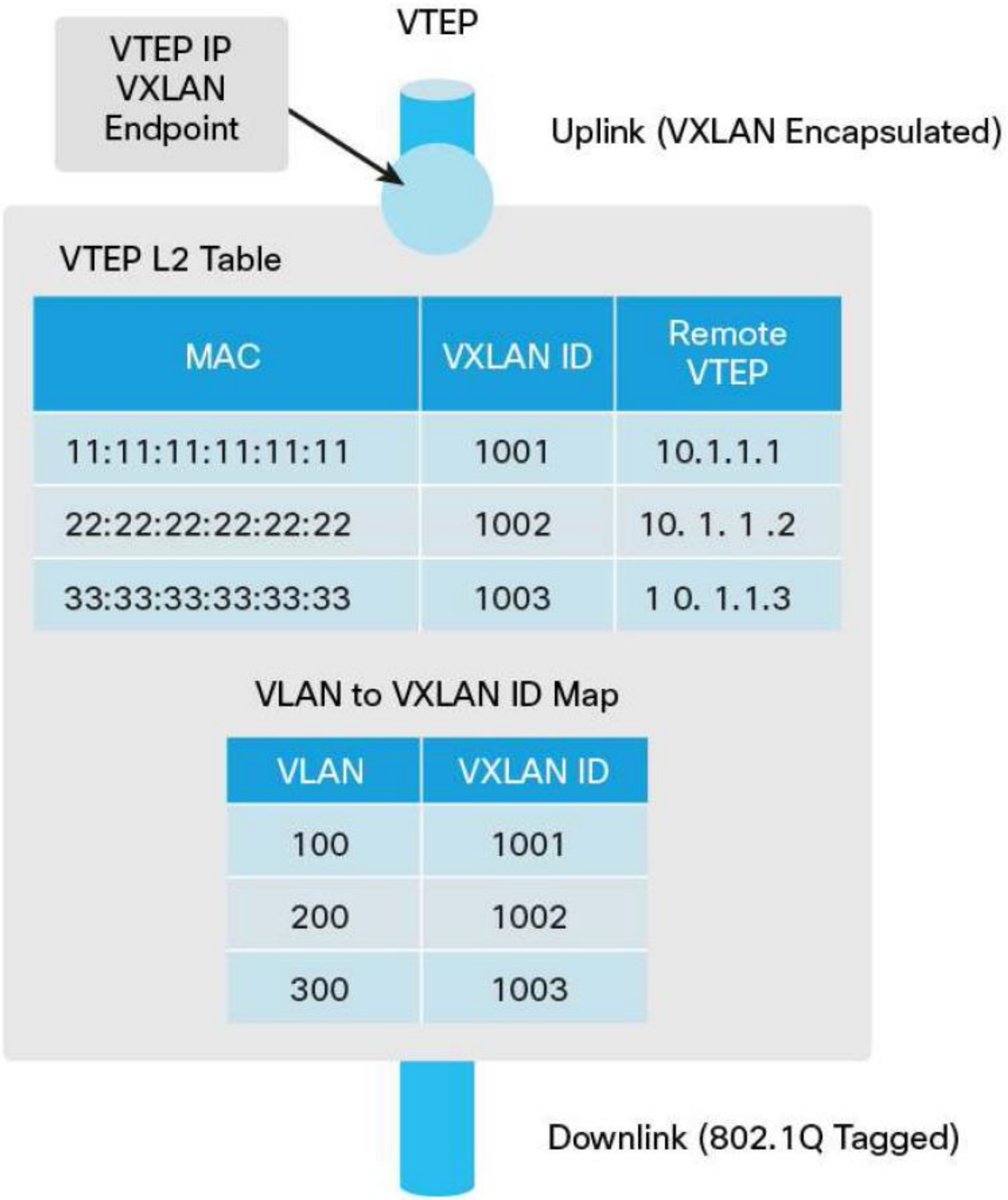


Virtual Extensible LAN, or **VXLAN**, is a Layer 2 overlay scheme over a Layer 3 network. It uses an IP/UDP encapsulation so that the provider or core network does not need to be aware of any additional services that VXLAN is offering. A 24-bit VXLAN segment ID or VXLAN network identifier (VNI) is included in the encapsulation to provide up to 16 million VXLAN segments for traffic isolation and segmentation, in contrast to the 4000 segments achievable with VLANs. Each of these segments represents a unique Layer 2 broadcast domain and can be administered in such a way that it can uniquely identify a given tenant’s address space or subnet.

➢ IETF Virtual Extensible LAN (VxLAN)

VXLAN can be considered a stateless tunneling mechanism, with each frame encapsulated or de-encapsulated at the VXLAN tunnel endpoint (**VTEP**) according to a set of rules. A VTEP has two logical interfaces: an uplink and a downlink.

The uplink is responsible for receiving VXLAN frames and acts as a tunnel endpoint with an IP address used for routing VXLAN encapsulated frames. These IP addresses are infrastructure addresses and are separate from the tenant IP addresses for the nodes that use the VXLAN fabric. The VTEP can be located either on a physical switch or within the hypervisor virtual switch in a server virtualization deployment.



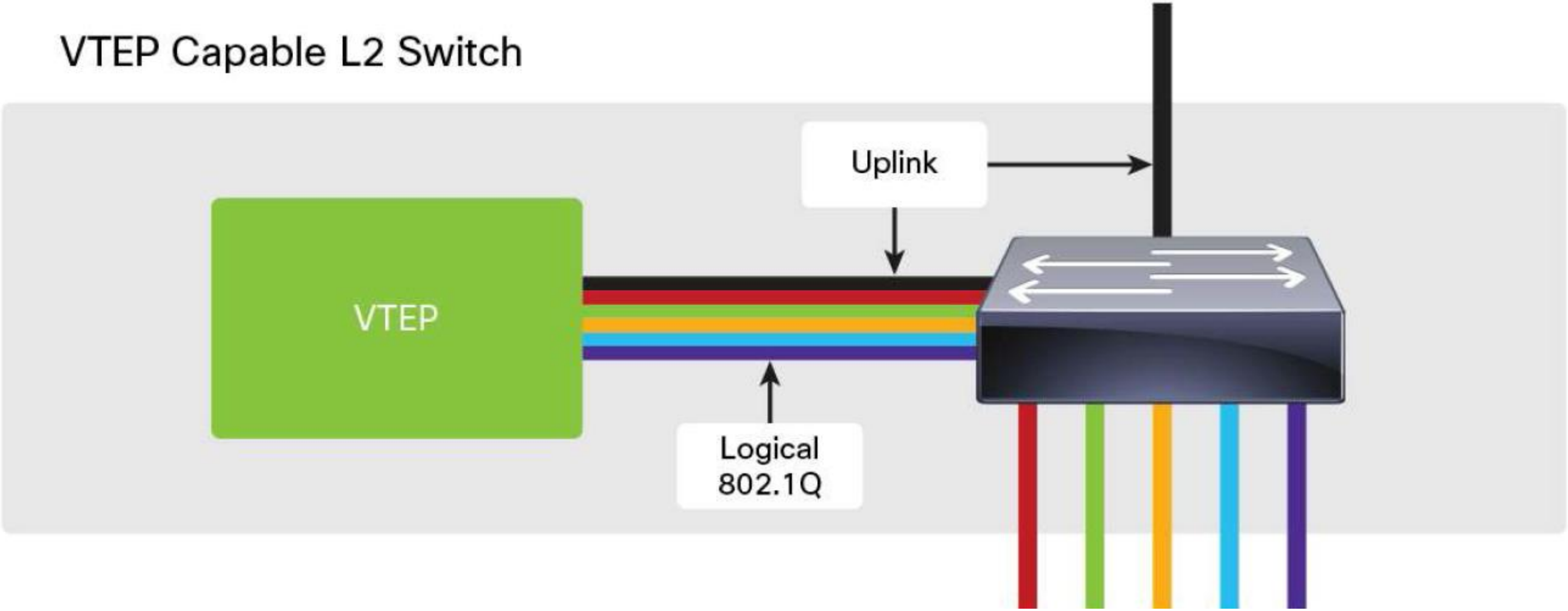
VTEP L2 Table

MAC	VXLAN ID	Remote VTEP
11:11:11:11:11:11	1001	10.1.1.1
22:22:22:22:22:22	1002	10. 1. 1 .2
33:33:33:33:33:33	1003	1 0. 1.1.3

VLAN to VXLAN ID Map

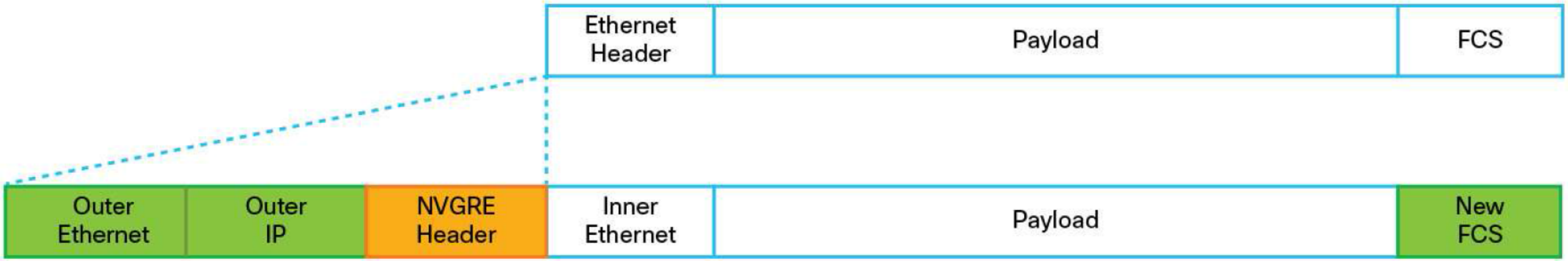
VLAN	VXLAN ID
100	1001
200	1002
300	1003

➤ IETF Virtual Extensible LAN (VxLAN)



VXLAN frames are sent to the IP address assigned to the destination VTEP; this IP address is placed in the outer IP destination address packet. The IP address of the VTEP sending the frame resides in the outer IP source address packet. Packets received on the uplink are mapped from the VXLAN ID to a VLAN, and the Ethernet frame payload is sent as an IEEE 802.1Q Ethernet frame on the downlink. During this process, the inner source MAC address and VXLAN ID are learned in a local table. Packets received on the downlink are mapped to a VXLAN ID using the VLAN of the frame. A lookup is then performed in the VTEP Layer 2 table using the VXLAN ID and destination MAC address; this lookup provides the IP address of the destination VTEP. The frame is then encapsulated and sent out the uplink interface

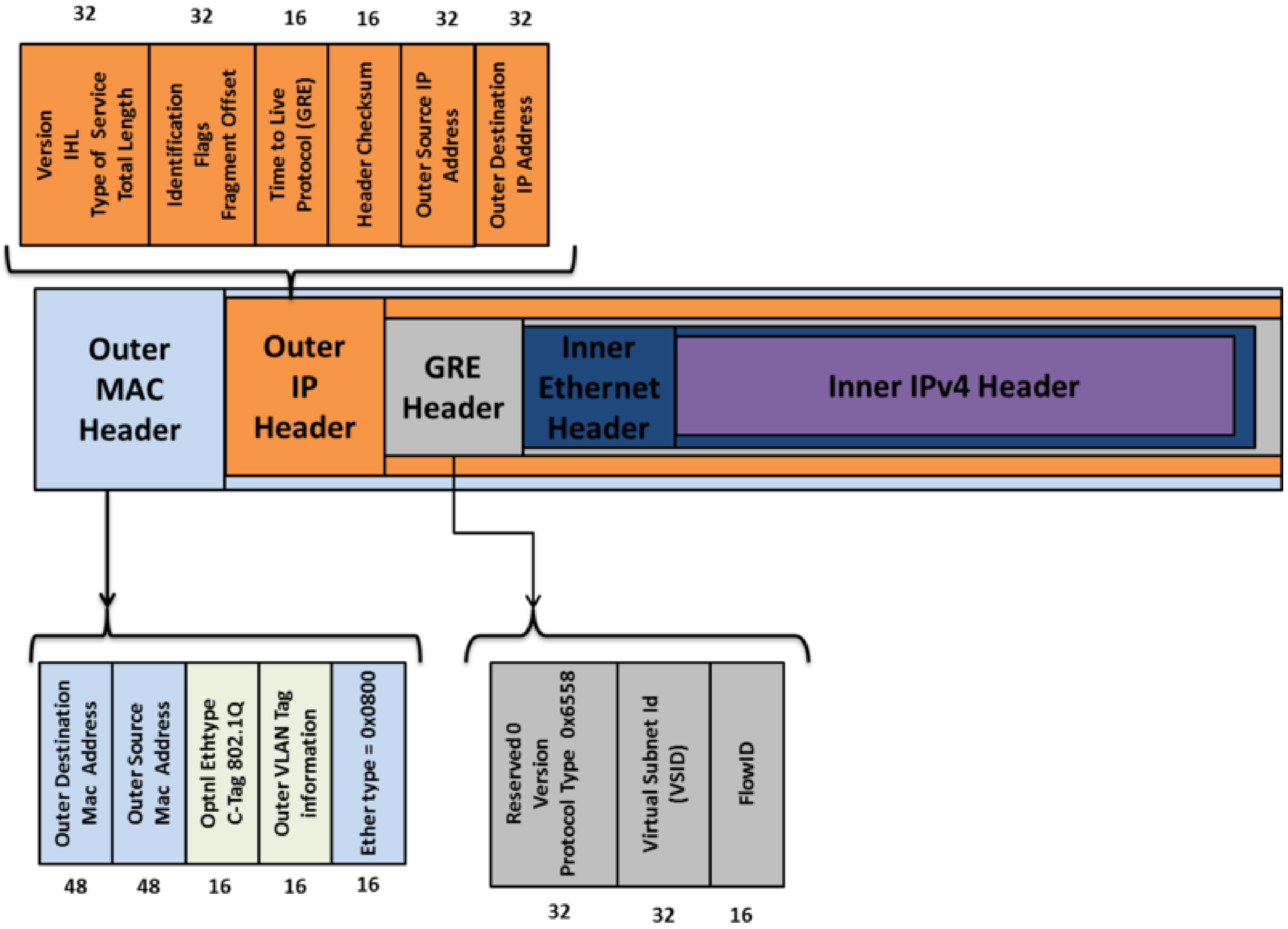
➢ Network Virtualization Using Generic Routing Encapsulation (NVGRE)



Network Virtualization Using Generic Routing Encapsulation, or **NVGRE**, allows the creation of virtual Layer 2 topologies on top of a physical Layer 3 network. This design is achieved by tunneling Ethernet frames inside an IP packet over a physical network. NVGRE supports a 24-bit segment ID or virtual subnet identifier (VSID), providing up to 16 million virtual segments that can uniquely identify a given tenant's segment or address space

The NVGRE endpoints are responsible for the addition or removal of the NVGRE encapsulation and can exist on a network device or a physical server. NVGRE endpoints perform functions similar to those performed by VTEPs in a VXLAN environment, and they are also responsible for applying any Layer 2 semantics and for applying isolation policies based on the VSID.

➢ Network Virtualization Using Generic Routing Encapsulation (NVGRE)

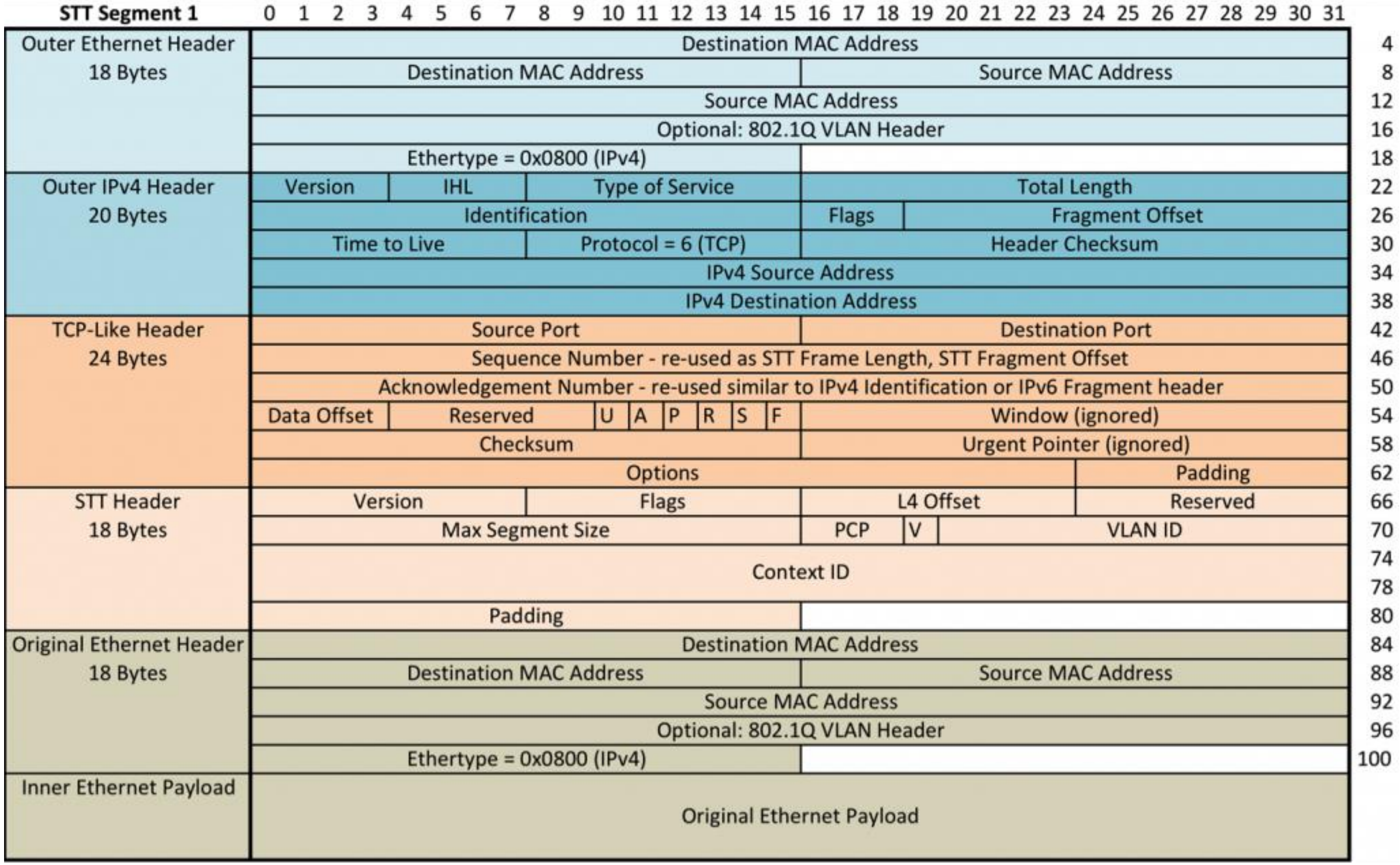


A main difference between VXLAN and NVGRE is that the NVGRE header includes an optional flow ID field. In multipathing deployments, network routers and switches that can parse this header can use this field together with the VSID to add flow-based entropy, although this feature requires additional hardware capabilities.

As with VXLAN, the NVGRE draft standard does not specify a method for discovering endpoint reachability. Rather, it suggests that this information can be provisioned through a management plane or obtained through a combination of control-plane distribution or data-plane learning approaches.

VXLAN	NVGRE
VNI – VXLAN Network Identifier (or VXLAN Segment ID)	TNI – Tenant Network Identifier
VxLAN header + UDP header + IP header + Ethernet header = 8+8+40+16 = 72 bytes addition per Ethernet frame	GRE header + IP header + Ethernet header = 8+40+16 = 64 bytes addition per Ethernet frame
VTEP - VXLAN Tunnel End Point - originates or terminates VXLAN tunnels	NVGRE endpoint
VXLAN Gateway - forwards traffic between VXLAN and non-VXLAN environments	NVGRE gateway
New protocol	Extends existing protocol for new usage
Multipath using different UDP ports	No multipath since GRE header is same

➢ IETF Stateless Transport Tunneling (STT)



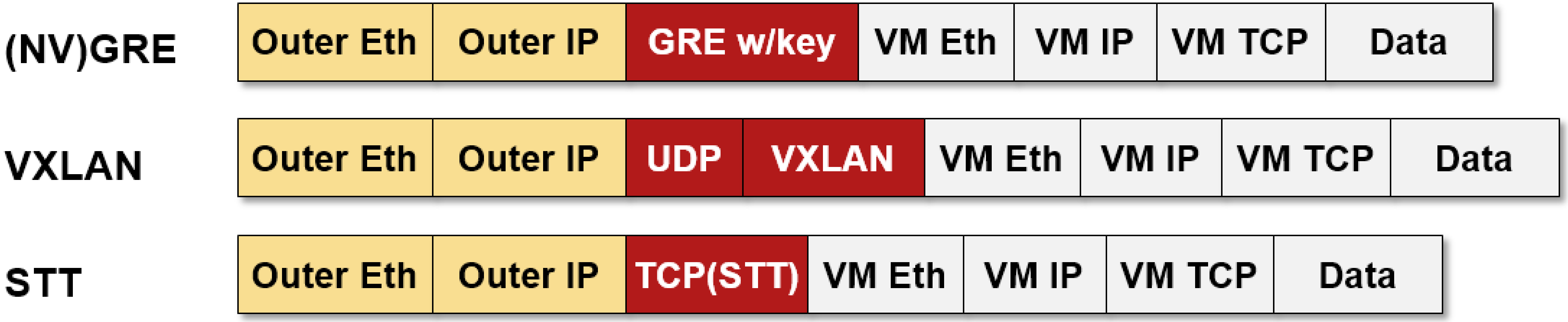
Stateless transport tunneling (**STT**) is an overlay encapsulation scheme over Layer 3 networks that use a TCP-like header within the IP header. The use of TCP fields has been proposed to provide backward compatibility with existing implementations of NICs to enable offload logic, and hence STT is specifically useful for deployments that are target end systems (such as virtual switches on physical servers). Note that, as the name implies, the TCP fields do not use any TCP connection state.

➢ IETF Stateless Transport Tunneling (STT)

One area that STT specifically addresses is the size mismatch between Ethernet frames and the maximum transmission unit (MTU) supported by the underlying physical network. Most end-host operating systems today set the MTU at a small size so that the entire frame plus any additional (overlay) encapsulations can be transported over the physical network. This setting may result in a potential performance degradation and additional overhead compared to frames that can be transmitted with their desired maximum segment size (MSS). STT seeks to exploit the TCP segmentation offload (TSO) capabilities built into many NICs today to allow frame fragmentation with appropriate TCP, IP, and MAC address headers, and also the reassembly of these segments on the receive side.

Similar to other encapsulations discussed earlier, STT contains a virtual network identifier that is used to forward the frame to the correct virtualized network context. This identifier is contained in a 64-bit context ID field and has a larger space to address a variety of service models and allow future expansion.
Host-based overlay networks address many of the challenges posed by rigid underlay networks and their associated protocols (Spanning Tree Protocol, etc.,), but the overlay network needs to be integrated with the physical network.

A major and unfounded assumption about host-based overlay networks is that the underlying network is extremely reliable and trustworthy. However, an overlay network tunnel has no state in the physical network, and the physical network does not have any awareness of the overlay network flow. A feedback loop is needed from the physical network and virtual overlay network to gain end-to-end visibility into applications for performance monitoring and troubleshooting.



- Three competing encapsulations
- Minor technological differences (load balancing, TCP offload)
- None supported by legacy networking hardware or IDS/IPS gear
- No security features ➔ transport network MUST be secure
- What really matters is the control plane

Q&A