SOME OBSERVATIONS ON THE HISTORY OF THE BGP PROTOCOL

For NLNOG Day 2018

Jeffrey Haas, Distinguished Engineer

<jhaas@juniper.net>

JUNIPER NETWORKS | Engineering Simplicity

QUICK OBSERVATIONS ON LANGUAGE

Speaks three languages: Tri-lingual

Speaks two languages: Bi-lingual

Speaks one language: American

Audience members are encouraged to ask the speaker to speak slower, or more clearly, if there is any need.

A FEW DETAILS ABOUT THE PRESENTER

Started working at a small ISP (roughly tier-3) in Michigan, US, around 1996.

- 10Mbps of upstream bandwidth across 3 providers!
- Proteon and Livingston routers!

- Handled everything from dial-up troubleshooting for new users to BGP issues with upstreams. (Intro to BGP.)
- Carried a pager

Spent a year working on the old NSFnet route servers (Rsng)

Started working on BGP in the GateD routing stack as part of NextHop.

BGP, SNMP, and flow work at Arbor Networks

BGP again at Juniper.

STANDARDS WORK

Internet Engineering Task Force (IETF), participant in BGP-4 standards work that lead to RFC 4271

- This work currently happens in the “idr” Working Group for “inter-domain routing”
  - Work on the OSI IDRP protocol didn’t end up being popular long term.
- This working group used to be named bgp

Ironically, a “task force” often implies a group with a limited lifespan.

- It’s been 32 years

Co-chairs the BFD working group

PERSONAL STUFF

Mediæval style combat in the SCA

Making and drinking beer

ERGO…

WHAT CAME BEFORE

A few comments on what became the Internet

FROM ERIC ROSEN (AUTHOR OF EGP) [1/5]

Originally there were just a bunch of networks (ARPAnet, SATnet, etc.) with no interconnection. Each network had its own L2 and L3 protocol, generally with different UNI and NNI. E.g., ARPAnet had NCP and 1822 (AHIP) as its L3 and L2 UNIs respectively. ARPAnet also had "End-End Protocol" as its L3 NNI, and "IMP-IMP Protocol" as its L2 NNI. Generally the addressing scheme of a particular network was NodeNumber/PortNumber.

Then came the Internet Gateway. Each Gateway attached to two or more networks, looking like a host to each one. Each network was assigned a globally unique number from 1-255. Each Gateway on a given network would be configured with the IP address and Network address of all the other Gateways on that network. Generally the IP address was derived algorithmically from the network address. E.g., the host attached to ARPAnet node 53 port 4 would get the IP address 10.4..53.

IP was regarded at this time as a Layer 3.5 protocol. Most hosts on the ARPAnet only spoke NCP/1822, but a few Unix systems could speak IP.

FROM ERIC ROSEN (AUTHOR OF EGP) [2/5]

At the time, the Internet was a strictly hierarchical system, with the ARPAnet at the root.

Internet routing was done via a protocol known as GGP (Gateway-gateway protocol). This was a distance vector protocol, not much different than RIP. Each Gateway would report its distance to each of the 255 networks. Distance 1 for directly connected networks. For networks learned about via GGP, a gateway would add 1 to the distance before forwarding to other gateways. There was no split horizon, and unreachability required counting to infinity.

FROM ERIC ROSEN (AUTHOR OF EGP) [3/5]

So what could go wrong?

Well, there were a lot of loops. There were a number of different gateway implementations, and each implementor was convinced that he understood better how to implement routing than any of the other implementors. BBN's gateways would ***send routing updates as fast as possible*** when the distance to some network was changing. Another implementor ***decided that it was not necessary to send routing updates more often than once every 30 seconds***. Naturally this slowed down the convergence across the entire Internet for many minutes, leading to a lot of loops that lasted long enough to create user-visible disruptions. ***It was very hard to explain to this fellow that his clever implementation was causing loops all around the world.***

There were other cases where implementors put in their own optimizations, based on their local needs, ***without considering the global impact***.

FROM ERIC ROSEN (AUTHOR OF EGP) [4/5]

The lessons I drew from this sort of thing were:

- You don't want to be a routing peer with the clueless;
- You need to be really careful about exchanging information with the clueless;
- There's no way to prevent any particular network from being run by the clueless;
- There's no hope of getting agreement among the different networks on a routing metric;

This gave rise to the notion of "Autonomous System", and to the idea that you need pretty strong barriers between your own clueful network and everyone else's clueless network.

FROM ERIC ROSEN (AUTHOR OF EGP) [5/5]

BGP built on EGP's notion of Autonomous System, but added the AS-path. Using the AS-path length as a metric, and using AS-path for loop control, BGP thus removed the need for hierarchy. Of course, AS-path length is a pretty dumb metric, but it's easy enough to explain to the clueless ;-) Frankly, I don't think it would ever have occurred to me that you could just make up a metric that didn't really mean anything, and then use it for routing!

BGP also added a couple of other things that have proven to be very important:

- Policy. In the early days of the Internet, the idea of commercial ISPs and bilateral agreements just didn't exist, and the need for policy control at AS boundaries just wasn't apparent.
- TCP. EGP was based on periodic refreshing with datagrams, and hence would not scale to hundreds of thousands of networks. Of course, Postel's instructions were to design for a couple of hundred. At the time of EGP, routers generally did not have TCP implementations, so using TCP seemed like a non-starter.

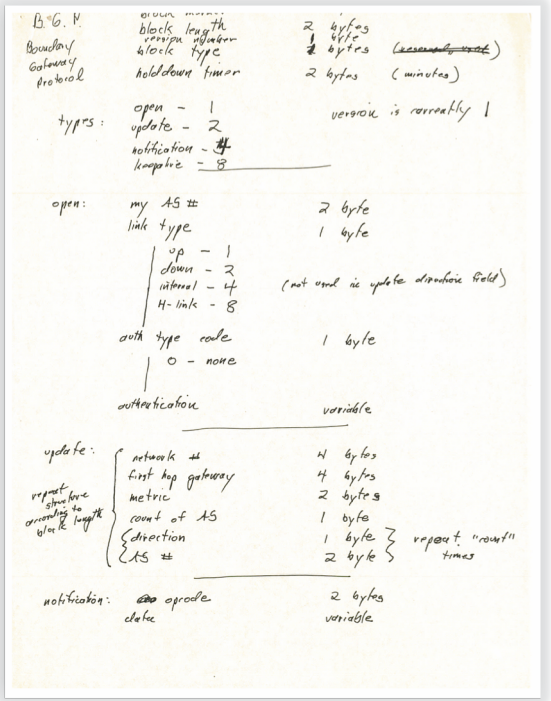
FROM JOHN SCUDDER

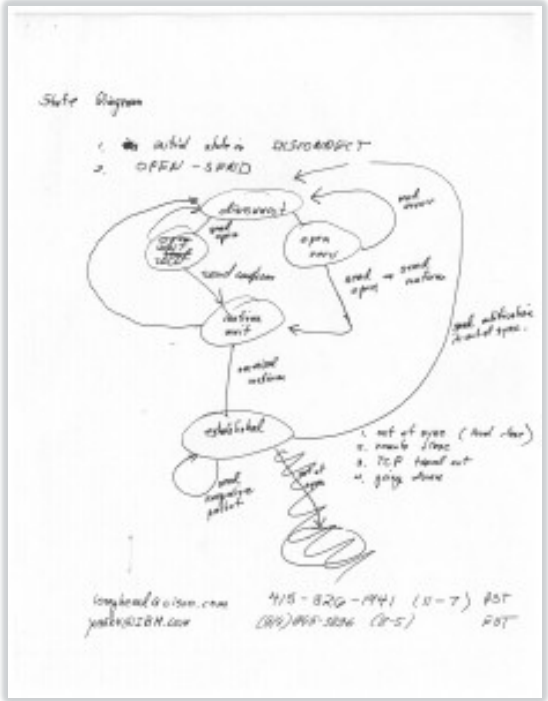
Regarding EGP's UDP transport, I remember how we had to hustle to transition various regionals to BGP (v3 at the time, IIRC) because the Ethernet cards on the routers the regional was using were capable of sending N back-to-back frames but the cards on the IBM PC-RTs our NSSes were built from could only consume N-1, and tail-dropped the Nth. I think N may have been 8. Anyway, when the regional's routing table grew to the point where its EGP update was fragmented across N frames, the NSS suddenly stopped seeing updates, because tail drop. Short-term mitigation was for the regional to reduce what they announced to us, followed by a scramble to move to BGP.

THE PATH TO BGP-4

Only optimists expect to get it right the first time

A TWO NAPKIN PROTOCOL





http://www.computerhistory.org/atchm/the-two-napkin-protocol/

BGP VERSION HISTORY

- RFC 1105: BGP-1, pub. 1989
- RFC 1163: BGP-2, pub. 1990
- RFC 1267: BGP-3, pub. 1991
- RFC 1654: BGP-4?, pub. 1994
- RFC 1771: BGP-4, pub. 1995
- RFC 4271: BGP-4!, pub. 2006

BGP-1 INTERESTING NOTES

Yakov wrote his name as Jacob at this point. ☺

"The initial BGP implementation is based on TCP [4], however any reliable transport may be used."

The maximum message size is 1024 bytes.

16 bits of marker

There was an OpenConfirm message in the protocol.

- This was subsumed later by the first keepalive.

Hold time was part of each message.

There was a link type. BGP-1 was directional!

- This may sound familiar to those trying to prevent route leaks today
- Eventually became "EGP, IGP, Incomplete"

Authentication was there from the beginning, but no other way for extensions

Network field (NLRI) was fixed 4-bytes

Networks had a metric!

Every receiver looked for AS-loops

The FSM had a transition table

BGP-2 INTERESTING NOTES

A strong caution about hop-by-hop routing is included in the intro.
A companion RFC 1164 was published:

- It mentions styles of ASes: stubs, transit, multi-homed.
- It mentions policy and even provides the start of a policy algebra.
- It mentions using performance based criteria for route selection. One form is shorter AS-path!
- BGP and the IGP interact. Tagging! Sync! (RFC 1364)

BGP and interaction with other protocols is called out. (E.g. EGP.)

Message limit grows to 4096
Marker grows to 16 bytes

We lose the OpenConfirm message
We still have authentication, but no extensibility.
Path Attributes are introduced:

- Origin as we have today in BGP-4
- AS_PATH loses directionality, but not typed. Loop detection done by each receiver for whole path.
- NEXT_HOP introduced
- UNREACHABLE – We didn't put this as WD_NLRI
- INTER-AS METRIC – Roughly what we have as MED today.

Version negotiation!
Hold time moved to the Open message.
Notification splits into a code and a sub-code.

BGP-3 INTERESTING NOTES

Open added the BGP Identifier (router ID) to deal with connection collisions

NEXT_HOP is loosened to not have to be from the same AS. (Portents of the route-servers.)

Advice about frequency of route advertisement and route selection

BGP-4, RFC 1654 INTERESTING NOTES

IDRP work was going on at this time. BGP admits to borrowing ideas. ☺

IGP interaction doc is updated to RFC 1745.

Support for CIDR! (And thus, variable length fields in NLRI.)

Aggregation. (Yay! or Boo-hiss!)

The introduction of the Adj-Rib-(In|out)/Loc-Rib naming from OSI. The idea is not explicitly mentioned in prior versions.

WD_NLRI introduced, and thus the idea of an implicit/explicit withdrawal.

INTER-AS METRIC renamed to MULTI_EXIT_DISC

Add LOCAL_PREF

Add ATOMIC_AGGREGATE

Add AGGREGATOR

Connection collision gets more sophisticated

An extensive section on route selection is introduced.

BGP-4, RFC 1771 INTERESTING NOTES

Authentication becomes “Optional Parameters”.

- There is a long IETF history of abusing “security” fields for extensibility.

It pretends RFC 1654 didn’t exist

The additional changes are unclear here?

BGP-4, RFC 4271

The goal was to deal with all of the edge cases and get the document to a level of maturity to call it “done”.

- It took 12 years and a massive number of revisions.
- https://tools.ietf.org/html/draft-ietf-idr-bgp-issues-06 will have more than you’d ever thought possible.

Route reflection and Confederations had happened, and needed accommodation.

Route refresh happened and had FSM impact

AS_PATH prepending was formalized

Lots of clarifications on nexthop, route selection.

Authentication and its use of the marker were deprecated

The FSM… ugh, the FSM.

ATOMIC AGGREGATE

Like the human appendix, ATOMIC_AGGREGATE is an organ that wasn't well understood and able to be lived without fairly well. But knowing that it's there, you try to accommodate it. Trying to deal with it, and the implications on aggregation (which mostly never happened), lead to a large number of the revisions toward RFC 4271.

The reverse engineered intent was that when routes were leaked from classful networks to classless in BGP-4, they shouldn't be (programmatically?) de-aggregated. People generally didn't do this, and the procedures involved in setting this attribute were consistently wrong.

Ironically, programmatic de-aggregation and the headaches it causes is a common operational tool and headache these days.

THE ROAD TO EXTENSIONS IS PAVED WITH GOOD INTENTIONS

Marking up routes; some general comments

PATH ATTRIBUTES

As early as BGP-2, Path Attributes provided a mechanism by which new attributes could be carried.

Everything that wasn't part of the core spec was considered optional.

Path Attributes cleanly used a TLV and thus provided a way for the unknown contents to be passed along without understanding them. You validate the lengths and will mark the attribute "partial" if you didn't understand it.

The extension mechanism didn't gain any uses until BGP-4.

TRANSITIVITY

In practice pretty much most new features were defined as transitive, meaning that the propagated by implementations that didn’t understand the contents.

This eased deployment of new features in scenarios where not every BGP router had to understand the new feature.

OPTIONAL TRANSITIVE NONSENSE

The main fault of transitivity as a BGP extension is that when invalid contents of a new Path Attribute are passed along by ignorant speakers and validated further downstream, RFC 4271 has you drop the BGP peering session you received it from.

“Killing the messenger”.

Bad attribute contents responsible for a number of global routing outages.

Eventually required an amended (and controversial!) new error procedure: RFC 7606.

This is particularly a problem in VPN technologies where routes begin their life in one domain, and pick up or should leave attributes as they move into another. E.g. L3VPN.

NON-TRANSITIVITY

In practice, non-transitive attributes aren't common.

They require each speaker to understand them. This is great when a new feature's deployment needs to be contiguous. However, it's often the case that a given network may have some number of updated devices with ignorant transports.

What's desired is better attribute scoping.

TOXIC EXTENSIBILITY

Four out of the eight flag bits for Path Attributes are allocated. Perhaps they can be used for extensions too!

draft-ietf-idr-optional-transitive-02, a predecessor to the BGP Error Handling RFC 7606 tried this.

Some implementations treated non-zero RESERVED flag bits as an error and dropped the session. While those implementations are non-conformant to RFC 4271 and earlier (all the way back to BGP-2), there's enough deployment of the buggy behavior to render this a toxic mechanism.

- Ideally this would have also have resulted in a CERT announcement that would have caused the offending implementation to get cleaned up.

THE ROAD TO EXTENSIONS IS PAVED WITH GOOD INTENTIONS

Session and route scaling

CONFEDERATIONS

RFC 1965 – BGP Confederations (EXPERIMENTAL status!)

- Stole the idea from IDRP. ☺
- The I-D vs. the RFC had the AS_PATH code points **reversed.**
- Considerations about what happened with next-hops wasn't consistent. And what about prepended confed segments?
- Fundamentally required all implementations to be okay with incompatible AS_PATH numbering. Effectively presumed a flag day in the Internet had already happened.
- Updated by RFC 3065. Some implementations didn't implement loop detection on the global-AS number properly.

ROUTE REFLECTION

RFC 1966 in 1996 (EXPERIMENTAL status!)

- Points out RFC 1863 route server experiment that never went anywhere.
- Didn’t require the edges to be upgraded (better incremental deployment)

RFC 2796 in 2000

- An edge case about what to do about ebgp received routes and reflection
- Suppress sending routes back to the originator during reflection
- Stronger requirement to not meddle with the NEXT_HOP
- Caveats about information hiding. (MED election.)

RFC 4456 in 2006

- Route selection is now impacted and documented. Turns out this wasn’t as transparent to the edges after all.

EBGP ROUTE SERVERS

RFC 7947.

This feature is as old as the Internet. RSes fell out of favor for years and have come back into fashion.

They weren't really documented, except in the academic papers ISI issued on the model.

Merit ran a second generation route server infrastructure at the old Internet NAPs that integrated with the RAdb IRR.

However, they don't have a mechanism to let them have redundancy. (RFC 1863 started going there.)

DAMPING NOISY REACHABILITY

In early RFCs, BGP recognized that the protocol scaled partially based on the number of UPDATE messages. Recommendations for minimizing messages through better packing were done very early in the protocol's life.

Somewhat early on, damping mechanisms were proposed for BGP:

- A form of this was documented in RFC 2439. However, no implementation is properly compliant with the document as written.
- In practice, damping was a large hammer that made people extremely afraid to cause BGP routes to flap.
- Also in practice, it was found that damping simply doesn't work for many scenarios due to the meshiness of the Internet. RIPE urged that damping stop being used. (RIPE-378)
- RFC 7196 proposed a way to make it actually usable – in principle.
- What remains irritating is a small number of prefixes on the Internet are responsible for the noise.

GETTING YOUR ROUTES BACK

BGP is stateful. If you throw routes away (e.g. policy), you don't get them back.

RFC 2918 introduced a new BGP message for Route Refresh that let you get your routes re-sent to you for a given AFI/SAFI.

It does the job.

It's very noisy and causes a lot of work at a time when routers are already very busy: reconfiguration.

RFC 7313 provided a light-weight mechanism to use route refresh as a lighter form of graceful restart.

THE ROAD TO EXTENSIONS IS PAVED WITH GOOD INTENTIONS

Carrying more than IPv4

THE ROAD TO IPV6 AND ELSEWHERE

BGP was quite popular for IPv4 (the Internet). But we'd known for some time that we were going to run out of IPv4 addresses.

The IETF eventually decided that IPv6 was the technology they were going to take forward.

- IPv6 wasn't the only proposal, but that's a different presentation.
- The stories say that IPv6 was proposed at a prior Amsterdam IETF.

How to carry the new reachability?

“EMBRACE AND EXTEND” VS. “SHIPS IN THE NIGHT”

When you reach the point where you’ve outgrown the core behavior of your protocol, your choice is often bump the version and work on backward compatibility, or find a clever hack to wedge in the new mechanism.

The motivating factor for what you choose in many cases is governed by your installed base.

The underlying question eventually becomes: When BGP-5?

CAPABILITIES AND MULTI-PROTOCOL

The “Optional Parameters” field in BGP-4 (remember that last minute change RFC 1654-> 1771?) provided a place to create the new BGP Capabilities Advertisement feature, which was standardized in RFC 2842.

- The feature was originally “capabilities ***negotiation***” which lead to all sorts of violently disagreeing code that couldn’t bring up sessions reliably. We still see bugs that are vestiges of this.

And once we had the ability to optionally support new features, we needed one that declared a way to carry new reachability.

- New reachability couldn’t fit into the existing IPv4 WD_NLRI/NLRI fields.
- We have plenty of Path Attribute code points available. Let’s put it there!
  (But remember that scoping issue! Capabilities used to limit route dissemination.)
- RFC 2858 standardized this. Just don’t ask too loudly what SNPA means. ☺

IPV6 IN BGP

After capability advertisement and multi-protocol extensions were crafted, some decided there was a need for BGP not only carry IPv6 global nexthops or link-locals, but ***both***.

RFC 2545 standardized this.

This was the first place where BGP's nexthop field might have two different sizes. Yay, bugs!

Many implementations also don't pay proper attention to the second link-local next-hop. Yay, bugs!

WHISPERS OF THE FUTURE - MPLS

Multi-protocol BGP not only enabled IPv6, but also provided the infrastructure for carrying the next big change for the industry: MPLS and MPLS enabled VPNs.

But we'll talk about that in a different presentation.

OR EVEN FIREWALLING

RFC 5575 – BGP Flowspec

THE ROAD TO EXTENSIONS IS PAVED WITH GOOD INTENTIONS

Marking up routes

COMMUNITIES

RFC 1997, 1998 were issued and provided two core services:

1. A way to mark some propagation control properties on routes that would be well known and acted upon without the need for explicit policy. (NO_EXPORT, etc.)

2. A way to arbitrarily mark routes so that actions could be taken on them as part of policy.

4 bytes wide, structured 2:2.

Clearly a popular feature. Easily deployed in an incremental fashion. Provided vendors differentiation in their policy languages.

EXTENDED COMMUNITIES

There was a need to have more structured route markup for things like VPNs.

RFC 4360 added extended communities for this.

8 bytes wide. Interpretation varied wildly based on first two bytes of data.

Yay structure!

Boo inconsistent structure! They look “formatted”, but can’t be counted on to be that way.

Have proven to be flexible and have been leveraged for all sorts of features, but inconsistent structure has made policy language interaction messy at best.

(And really for a different presentation, “magic” interpretation leads to some interesting protocol ambiguities. E.g. what do you do with two link-bandwidth communities on the same route?)

SEGUE TO ORF, RT-CONSTRAIN

A few features were developed to try to take advantage of route properties for automatic filtering. You send a receiving router a filter for what you're wanting to receive:

RFC 4684 – RT-Constrain

RFC 5292 – Outbound Route Filters

LARGE COMMUNITIES

For protocol developers, route marking is “just a number”. For operators, semantic convenience is attached to those numbers.

Regular RFC 1997 communities were usually treated as AS:AS. With the advent of 4-byte ASes, equivalence was needed for the use cases covered by 1997 communities.

Large communities went a step further: 4:4:4

CARRYING “INTENT” IN ROUTES

Common policies can require a jumble of policy on multiple vendors’ routers in a network to implement the intent.

Being able to carry structured information along with the intent in a route’s markup is already done using existing communities. But the lack of standardization both makes for flexibility, and more work in the implementation of the code and the policy.

flexible-communities and wide-communities are two discussions that have happened over the years.

Very controversial. Currently “science fiction”.

(And let’s save discussion about these for over beer rather than this session!)

MORE MARKUP! MORE!

XXC – Extra extended communities are starting to make the rounds in IETF as a discussion point.

There's not a lot of argument about the need for "more room" in a "community", especially for VPN purposes. Mostly the discussion is about formatting.

THE ROAD TO EXTENSIONS IS PAVED WITH GOOD INTENTIONS

Transport Security; Resiliency

TRANSPORT SECURITY

BGP works over TCP. This incredibly simplified some of the headaches that were seen in the EGP days. The protocol developer isn't (necessarily) responsible for the underlying TCP stack!

TANSTAAFL* - This tends to mean that most serious BGP developers are partial experts on the TCP protocol and many of its most common extensions.

BGP itself threw out its built-in security mechanism as part of the transition to BGP-4.

The Internet, even the early one, was full of Not-Nice people. TCP RST attacks were being used to knock over BGP peering sessions.

This resulted in RFC 2385, TCP-MD5. BGP was the main user of this extension. It was a really quick hack to protect against a narrow class of attacks.

BGP works fine across IPsec, but IPsec is problematic for routing protocols.

IETF invented TCP-AO. No one has implemented it. There's a lot of interesting politics here.

* There ain't no such thing as a free lunch

GRACEFUL RESTART

As the impact of a BGP router continued to grow, mechanisms were looked for to increase resiliency.

As part of a large push across IETF, BGP got a graceful restart mechanism along with a lot of other protocols. This was RFC 4724.

GR requires help from the devices you talk to. As such, it's not a very popular feature.

Non-stop routing like features appeared in many vendor portfolios to cover some of this resiliency story. They often cause massive penalties in operations and performance, don't work as well as they sometimes should, and mostly make software developers angry.

FUTURE DEVELOPMENTS

Science fiction or operational fact?

FUTURE DEVELOPMENTS IN BGP

Routing security. RPKI (ROA validation) and bgpsec. We're starting to see RPKI in better use! bgpsec… not so much.

Route leak prevention has been a popular topic for years. See BGP-1. ☺ This is one part protocol glue, and ninety-nine parts operational glue.

Better route and attribute scoping. We're regularly seeing this as a conversation in new BGP feature discussions. Lots of proposals, nothing with sufficient traction yet. Is this what pushes us to BGP-5?

Automation infrastructure. Bootstrapping BGP sessions automatically; zero-touch provisioning.

LESSONS

GOOD STUFF

Rough consensus and running code worked well for BGP in the early days. The protocol evolved quickly.

An extension mechanism built in up front gave the protocol a significant life without needing a major version number bump.

The ability to carry new things keeps BGP a popular protocol. VPNs are an example of this.

BAD STUFF

BGP as the “everything, and the kitchen sink too!” protocol means there’s a lot of competing interests vying for the stability and performance in potentially the same code base.

A BGP session carrying too many address families can have unexpected convergence behaviors.

Inadequate scoping has lead to experiments or too-early code leaking into the Internet and causing damage. Similarly, VPN scoped behaviors often leak outside the VPNs.

Code point management is critical. Deployed code squatting on improperly allocated code points can ruin the deployment of a feature. Better ways of deploying experiments/early code are needed.

UGLY STUFF

BGP is a deployed protocol.

“Don’t break my networks with your new stuff!”

Incremental deployment of new features is tricky, and much of what newer BGP protocol authors needs to understand. Most protocol developers don’t operate networks. Most operators don’t understand protocol development. ☺

THANK YOU

JUNIPER NETWORKS | Engineering Simplicity