Overloading of the routing protocols…

Kurtis Lindqvist
<kurtis@netnod.se>

Basic concepts of routing(protocols)

- For a given destination, which of the nodes that are adjacent to me is the shortest path
  - Can be influenced at will through cost
- Distribute the information of which nodes I can reach, and which nodes my neighbors can reach

Routing protocols in the Internet

- External connectivity - BGP
  - Between ISPs
  - Multihomed enterprises
  - Between two autonomous systems
- Internal connectivity - IS-IS, OSPF, RIP, EIGRP, etc…
  - Inside a network

Dependability

- The Internet at large will rarely by effected by IGP problems
- BGP problems can affect all and everyone
  - AS7007
  - Several incidents in Sweden
- The BGP invalidation problem
  - Ahuja / Labovitz

”We” decided to add functionality

- MPLS
  - TE
  - VPNs
- L2-VPNs
- L3-VPNs

…

- Pidgeons…

New demands…

- All the previous features requires distribution of their own topology information
- …we are all lazy and will go for the ”path of least resistance”

New demands

- Existing routing protocols modified to also distribute ”generic” information, or topology information for specific applications
- Both for internal and external routing
  - Tex BGP-MP, OSPF-TE, etc

What are the criteria for allowing certain information to be distributed with a routing protocol?

IETF-57

- Open Routing area meeting…
- …that was fun!
- Very opposing views
  - People have very different back-grounds
  - Developers vs operators vs ”customer demand”

The problem

- The Internet today is very reliable on one single routing protocol
  - Implementation
  - Configuration
  - Operation
- If this is the default, also for new applications and ”non-reachability information” we risk the Internet

The problem

- Nothing ever got more stable by adding code…
  - Routing software is not really stable today
- Convergence times of BGP and IGPs are already a problem
  - Adding parameters will not help
- With running multiple applications over the same protocol, I loose control of executed code

Routing AD summary

- Separate apps do not share a single transport connection,which implies that:
- They don't compete for socket buffers. No starvation at TCP
- They don't compete for protocol-level buffers and queues
- They don't compete for a possible per-connection processing time budget, hence don't starve each other at the intra-process scheduling level
- they don't compete for resources within the network
- they likely do not compete for resources in other parts of the local system that may be protocol-aware (such as line cards putting all BGP segments in a separate rate-limited queue)

Routing AD summary

- Separate apps are likely to not share a single scheduling context, which implies that they will not compete for the same per-process time budget
- Separate apps are likely to have their memory segments protected from each other in the OS'es supporting this. Hence, the risk of non-routing apps corrupting routing data is decreased

Routing AD summary

- Separate apps are likely to have their own pools of resources, such as memory heaps, so the risk of memory starvation is decreased
- In OS'es supporting protected memory, the above results in fate separation

Arguments from the other side

- Code reuse is good engineering
- Demux is done at higher level
- This is implementation details, and not for the IETF

Next steps

- Design team working on problem statement
- Decision will have impact on both IGP and EGP

?

Contact

Netnod Internet Exchange i Sverige AB

Bellmansgatan 30^{I}
SE-118 47 Stockholm
Sweden

Office address: Bellmansgatan 30^{I}

Telephone: +46-8-615 85 70
Telefax: +46-8-442 09 67
E-mail: kurtis@netnod.se
URL: http//www.netnod.se/