DUAL-STACK IS-IS DEPLOYMENT
GLOBAL TRANSIT COMMUNICATIONS

Presented by Mark Tinka

Chief Network Architect

Global Transit

Kuala Lumpur, Malaysia

GLOBAL TRANSIT

Presentation Overview

- Reasons for choosing IS-IS
- High-level IS-IS design choices
- Cisco & Juniper IS-IS BCP

IS-IS: The Interest

IS-IS: The Interest

- Main reason; to be able to “stretch” the network regionally and internationally.
- Limitations imposed by OSPF’s requirement to link all areas back to Area 0 (Backbone Area) was an impediment to network expansion.
- OSPF Virtual Links were not a viable option.
- Simplicity of the IGP was key.

Design Choice

Design Choice

- We chose a multi-level hierarchy in order to separate the core backbone from each individual PoP’s infrastructure.
- Level-1
  - Used within each PoP
  - Area value unique to each PoP
  - Area value similar for IS’s in the same PoP
- Level-2
  - Used on core links (between & through PoP’s)

Design Choice

- IS-IS used to carry Loopback and infrastructure addresses.
- iBGP used to carry customer prefixes.
- Traffic Engineering extensions for IS-IS were enabled for MPLS-TE.
- Route leaking was enabled for more optimal routing between levels.
- BFD support for IS-IS was used for fast(er) link failure detection.

Design Choice

- IETF Graceful Restart support for IS-IS was enabled to maintain traffic forwarding during reconvergence.
- Dual-stack IS-IS implemented; supporting both IPv4 and IPv6 address families.
- IS-IS MT (Multi-topology) was used to maintain adjacencies when transitioning single-stack links (IPv4-only) to a dual-stack state (IPv4 & IPv6).

Pre-IS-IS Network

Pre-IS-IS Network

- Multi-vendor network (Cisco & Juniper).
- OSPFv2 for the IPv4 address family.
- OSPFv3 for the IPv6 address family.
- Deployment was easy as new equipment was being deployed. However, a “ships-in-the-night” approach would have been just fine as well.
- Production code was current at the time, IOS 12.2(33)SRC2, 12.2(33)SXH3 & JunOS 9.2R2.

Deployment

Deployment

- Deployment details will be based on a narration of features used as interoperated between Cisco IOS and Juniper JunOS.
- While some “knobs” may not be available for certain features for one of these vendors, it does not mean these features are not fundamentally supported – that their default implementation may be more than satisfactory.

Deployment (Fast RIB Purge)

- For routing protocols capable of responding to link failures, the “less efficient” RIB process is used to delete next-hops that can no longer be used due to the associated interface being deleted from the routing table.
- If the routing table is large enough, this process could use up a number of CPU cycles and potentially slow convergence.

Deployment (Fast RIB Purge)

- This feature enhances the RIB infrastructure and causes it to (more) quickly delete routes associated with a failed interface, from the RIB.

Deployment (Fast RIB Purge)

- IOS implementation:

```
ip routing protocol purge interface
```

Deployment (Fast RIB Purge)

- JunOS implementation:

  (No equivalent command)

Deployment (Authentication)

- IS-IS Authentication
  - Authentication provides protection for IS-IS ajdacencies.
  - HMAC-MD5 password authentication recommended.
  - Authentication can be deployed for both Level-1 and Level-2.

Deployment (Authentication)

- IOS implementation:

```
key chain isis-security-l1
 key 1
   key-string xxxxx
key chain isis-security-l2
 key 1
   key-string xxxxx

router isis 1
 authentication mode md5
 authentication key-chain isis-security-l1 level-1
 authentication key-chain isis-security-l2 level-2
```

Deployment (Authentication)

- IOS implementation:

```
interface GigabitEthernet0/1
 isis authentication mode md5
 isis authentication key-chain isis-security-l1 level-1
 isis authentication key-chain isis-security-l2 level-2
```

Deployment (Authentication)

- JunOS implementation:

```
[edit protocols isis]
level 1 {
    authentication-key xxxxx
    authentication-type md5
}
level 2 {
    authentication-key xxxxx
    authentication-type md5
}
```

Deployment (Loopback Interface)

- Create a Loopback interface.
- This interface will not run IS-IS, although addresses configured on it will be propagated into IS-IS.
- Disabling IS-IS on this interface scales the IGP because LSP/Hello frames are not unnecessarily generated for it.
- It also prevents an IS-IS metric from being set (defaults to 0), which would have been ambiguous to the network.

Deployment (Loopback Interface)

- IOS implementation:

```
interface Loopback0
 ip address 192.168.0.1 255.255.255.255
 ipv6 address 2001:db8:192:168:0::1/128
```

Deployment (Loopback Interface)

- JunOS implementation:

```
[edit interfaces]
lo0 {
    unit 0 {
        family inet {
            address 192.168.0.1/32;
        }
        family inet6 {
            address 2001:db8:192:168:0::1/128;
        }
    }
}
```

Deployment (NET)

- Create the NET (Network Entity Title).
- This is made up of a private AFI (49), an Area part, a System ID (usually taken from the padded Loopback IPv4 address) and an N-SEL of 0.
- The Area part should be the same within a Level-1 domain, or else adjacencies will not form between L1 routers.
- The Area part can be different between Level-2 domains.

Deployment (NET)

- IOS implementation:

```
router isis 1
 net 49.0001.1921.6800.0001.00
```

Deployment (NET)

- JunOS implementation:

```
[edit interfaces]
lo0 {
    unit 0 {
        family iso {
            address 49.0001.1921.6800.0001.00;
        }
    }
}
```

Deployment (Activate IS-IS on Interface)

- Since IS-IS is a link state routing protocol, it needs to run directly on the interface.
- As such, it should be enabled to do so.

Deployment (Activate IS-IS on Interface)

- IOS implementation:

```
interface GigabitEthernet0/1
 ip address 192.168.1.1 255.255.255.192
 ipv6 address 2001:db8:192:168:1::1/112
 ip router isis 1
 ipv6 router isis 1
```

Deployment (Activate IS-IS on Interface)

- JunOS implementation:

```
[edit interfaces]
ge-0/0/0 {
    unit 0 {
        family inet {
            address 192.168.1.1/26;
        }
        family iso;
        family inet6 {
            address 2001:db8:192:168:1::1/112;
        }
    }
```

Deployment (Interface MTU)

- In order for an IS-IS adjacency to form on a common data link between IS’s, certain criteria contained in IS-IS Hello frames should match between IS’s.
- One of these matching criteria is MTU size.
- Should the MTU size differ between IS’s, an adjacency will not transition to the “Up” state.

Deployment (Interface MTU)

- Our implementation example will assume a link MTU of 9,000 bytes on Ethernet and POS circuits (8,997 bytes for Ethernet, 9,000 bytes for POS).

Deployment (Interface MTU)

- IOS implementation:

```
interface GigabitEthernet0/1
 mtu 9000
!
interface POS2/0
 mtu 9000
```

Deployment (Interface MTU)

- JunOS implementation:

```
[edit interfaces]
ge-0/0/0 {
    mtu 9014;
    }
}
so-0/3/0 {
    mtu 9004;
    }
}
```

Deployment (IS-IS Levels)

- IS’s can be deployed as L1, L2 or L1/L2 routers.
- Both Cisco IOS and Juniper JunOS default to running in an L1/L2 configuration.
- L2 routers simply need to share the same data link and be configured to run at Level-2.
- L1 routers need to share the same data link and be configured with the same Area value to run at Level-1.
- L2 IS’s don’t require identical Area values.

Deployment (IS-IS Levels)

- IOS implementation (L1 only):

```
interface GigabitEthernet0/1
 isis circuit-type level-1
!
router isis 1
 is-type level-1
```

- IOS implementation (L2 only):

```
interface GigabitEthernet0/1
 isis circuit-type level-2-only
```

- IOS implementation (L1/L2):

  Default operation.

  None of the above configurations required.

Deployment (IS-IS Levels)

- JunOS implementation (L1 only):

```
[edit protocols isis]
interface ge-0/0/0.0 {
    }
    level 2 disable;
}
```

- JunOS implementation (L2 only):

```
[edit protocols isis]
interface ge-0/0/0.0 {
    }
    level 1 disable;
}
```

- JunOS implementation (L1/L2):

  Default operation.

  None of the above configurations required.

Deployment (IS-IS Metric)

- IS-IS’s default metric is “cost”.
- Unlike other link state routing protocols like OSPF, IS-IS doesn’t automatically calculate its link metric.
- The default metric for all IS-IS links is 10 (unless those links are configured as “passive” within IS-IS, e.g., Loopback interface, in which case those become 0).
- There are two (2) styles of metrics in IS-IS.

Deployment (IS-IS Metric)

- Old-style metrics:
  - Can have an interface cost of 1 – 63.
  - Is a 6-bit metric value.
  - The total path metric is limited to 1,023 (the sum of all link metrics along a path).
  - Old-style metrics do not scale to support large networks.
  - They also do not enable the use of more advanced features such as:
    - MPLS-TE
    - Route leaking.

Deployment (IS-IS Metric)

- Wide metrics:
  - a.k.a "Extended" or "new-style" metrics.
  - Provides for a 24-bit metric field.
  - Link metrics can now have a maximum value of 16,777,215.
  - The total path metric can be as high as 4,261,412,864.
  - Is the recommended metric-style in modern IP networks.

Deployment (IS-IS Metric)

- IOS implementation:

```
router isis 1
 metric-style wide
!
interface GigabitEthernet0/1
 isis metric 400 level-1 (L1-only)
 isis ipv6 metric 400 level-1 (L1-only)
 !
 isis metric 400 level-2 (L2-only)
 isis ipv6 metric 400 level-2 (L2-only)
 !
 isis metric 400 (L1/L2)
 isis ipv6 metric 400 (L1/L2)
```

Deployment (IS-IS Metric)

- JunOS implementation:

```
[edit protocols isis]
level 1
    wide-metrics-only;
}
level 2 {
    wide-metrics-only;
}
interface ge-0/0/0.0 {
    level 1 {
        metric 400;
        ipv6-unicast-metric 400;
    }
    level 2 {
        metric 400;
        ipv6-unicast-metric 400;
     }
}
```

Deployment (DIS)

- On BMA (broadcast multi-access) links, such as Ethernet, the concept of a DIS (Designated Intermediate System) or Pseudonode is used.
- A DIS, similar to a DR (Designated Router) in OSPF, is responsible for flooding LSP’s over the common media.
- This prevents every IS from forming an adjacency with every other IS, as such, helping to scale the IGP.

Deployment (DIS)

- Rather, all other IS’s will form an adjacency with the DIS.
- A DIS is elected based on an IS with the highest priority value (a configurable parametre between 0 – 127).
- Both Cisco IOS and Juniper JunOS default to a priority value of 64.
- The IS with the highest priority “wins”.

Deployment (DIS)

- In the case of a priority value tie (e.g., assuming the default priority is not altered), the IS with the highest SNPA (Subnetwork Point of Attachment), i.e., the MAC address on LAN’s and DLCI on Frame Relay, becomes the DIS.
- On Frame Relay, if the DLCI is the same on both sides of the link, the router with the highest System ID wins.
- However, unlike OSPF, IS-IS does not have the concept of a backup DIS.

Deployment (DIS)

- Should the DIS fail, the next router with the highest priority or SNPA becomes the DIS.

Deployment (DIS)

- Also, unlike OSPF, DIS election is pre-emptive. If a new IS with a higher priority joins the network, it automatically becomes the DIS.
- Recommended practice is to make the election process predictable by configuring a higher-than-default priority for your primary DIS (e.g., 127), and a slightly lower priority for your backup DIS (e.g., 126).

Deployment (DIS)

- IOS implementation:

```
interface GigabitEthernet0/1
 isis priority 127 level-1
```

Deployment (DIS)

- JunOS implementation:

```
[edit protocols isis]
interface ge-0/0/0.0 {
    level 1 {
        priority 127;
    }
}
```

Deployment (Network-Type)

- IS-IS supports two (2) types of networks:
  - Broadcast networks:
    - Typically Ethernet.
  - Point-to-point networks:
    - Typically WAN-type links, e.g., Serial, POS, e.t.c.
- As Ethernet becomes more prevalent in the WAN core, it is necessary to, in these cases, treat it as a typical WAN interface, i.e., run it as a point-to-point network.

Deployment (Network-Type)

- This has the advantage of preventing a DIS election on this link, given that it is Ethernet.
- It also helps reduce link failure detection times, scaling the IGP.
- Point-to-point mode prevents the use of CSNP’s for database synchronization, simplifies the SPF computations and reduces the memory footprint due to a leaner topology database.

Deployment (Network-Type)

- IOS implementation:

```
interface GigabitEthernet0/1
 isis network point-to-point
```

Deployment (Network-Type)

- JunOS implementation:

```
[edit protocols isis]
interface ge-0/2/0.0 {
    point-to-point;
}
```

Deployment (BFD)

- BFD (Bidirectional Forwarding Detection) is a protocol used to provide fast forwarding path detection failure times for all types of media, encapsulations, topologies and routing protocols.
- BFD is faster at detecting failure than link state IGP’s are (generally, between 50ms and 300ms).
- When configured, the IGP becomes a client of BFD.

Deployment (BFD)

- When BFD detects a failure of a link (forwarding plane), it quickly signals this transition to the IGP within a pre-configured interval.
- The IGP then re-computes the best paths and reconverges accordingly.
- It is important to note that BFD, in and of itself, will not provide for faster convergence. BFD only provides a faster method to signal failure to the IGP.

Deployment (BFD)

- The IGP “must” be tuned, if necessary, to respond to BFD’s signal as soon as it’s received – or else BFD is simply another overhead.
- Generally, enabling BFD for the IGP is sufficient, as most other protocols and applications rely on the IGP for their operation, e.g., BGP, MPLS, RSVP, e.t.c.
- While BFD is supported for most or all of these other protocols and applications, it is not necessary to run it beyond the IGP.

Deployment (BFD)

- At this time, BFD has different levels of support for interface, encapsulation and routing protocol types across vendors, and across different operating system revisions within each vendor's platforms.
- Please consult the vendor documentation to ascertain what BFD features your platform and operating system version support.

Deployment (BFD)

- While BFD can be configured to signal the IGP within 50ms of forwarding plane failure detection (we’re still chasing after SONET/SDH), it is recommended to configure a more conservative value to maintain stability in the long term, e.g., 250ms to 300ms.
- Since parts of BFD can be distributed to the forwarding plane, it can be less CPU-intensive than reduced timers for routing protocols which live fully in the control plane.

Deployment (BFD)

- It is recommended to consider the benefit of using a slightly higher multiplier value for the BFD session on “longer” links as they could have a higher latency.
- Side note: with IPoDWDM promising to be all it can be, BFD might not be necessary, as IP routers now get direct visibility into the optical network, and “could” quickly converge to a redundant path when signal degradation (before failure) is detected.
- Vendors are claiming 15ms – 25ms.

Deployment (BFD)

- IOS implementation:

```
interface GigabitEthernet0/1
 bfd interval 250 min_rx 250 multiplier 3
!
router isis 1
 bfd all-interfaces
```

Deployment (BFD)

- JunOS implementation:

```
[edit protocols isis]
interface ge-0/0/0.0 {
    bfd-liveness-detection {
        version automatic;
        minimum-interval 250;
        minimum-receive-interval 250;
        multiplier 3;
    }
}
```

Deployment (iSPF)

- iSPF (Incremental SPF) is a mechanism used to maintain an up-to-date topology database without a corresponding expense in CPU resources.
- There are several cases where recomputation of the entire SPT is not necessary, e.g., when only one node is added to or removed from the network.
- In such cases, a full SPF run is not required.

Deployment (iSPF)

- While iSPF requires that each IS maintains more information about the topology in order to apply the incremental changes, it typically reduces CPU load in the long run.

Deployment (iSPF)

- IOS implementation:

```
router isis 1
 ispf level-1-2 60
```

Deployment (iSPF)

- JunOS implementation:

(No equivalent command)

Deployment (Hello Padding)

- Since IS-IS doesn’t support data fragmentation, IIH padding allows routers to detect, early on in the adjacency formation process, any errors caused by an MTU mismatch.
- As adjacencies are raised, IS-IS ensures links can support a sufficient frame size by padding outgoing IIH’s up to interface’s supported MTU configuration.

Deployment (Hello Padding)

- However, IIH Padding presents two drawbacks to network operations:
  - On high speed links, IIH Padding could strain interface buffers.
  - On low speed links, IIH Padding wastes bandwidth, and as such, could affect time-sensitive applications, e.g., VoIP.
- It is recommended IIH Padding be disabled.
- Cisco IOS enables IIH Padding by default.
- Juniper JunOS has IIH Padding disabled by default.

Deployment (Hello Padding)

- In IOS, IIH Padding is safe because the router will still pad the first five (5) frames to the full MTU, to aid in MTU mismatch discovery.
- JunOS also checks for MTU mismatch in the initial IIH’s that are transmitted and received between adjacent routers.

Deployment (Hello Padding)

- It is critical to note that since IS-IS runs directly at the data link layer, fragmentation and reassembly is not supported, as there are no other packets for the flooded LSP’s to be encapsulated into, e.g., like what IP does for OSPF.
- This means that all LSP’s in the IGP domain should be of a size less than the smallest MTU link in the network.

Deployment (Hello Padding)

- This is necessary because there is no way to check for the MTU value between IS's not directly adjacent.
- If the minimum MTU is not considered when deploying IS-IS, it is possible to "lose" some LSP's between non-adjacent IS's, leading to an inconsistent IS-IS topology database.

Deployment (Hello Padding)

- IOS implementation:

```
router isis 1
 no hello padding
```

Deployment (Hello Padding)

- JunOS implementation:

  No configuration required.

  IIH Padding disabled by default.

Deployment (Passive Interface)

- There are situations where it is not desirable to run IS-IS on an interface, but you still want to propagate that interface’s IP address within the IGP.
- This is accomplished by telling IS-IS to treat the interface as passive.
- Passive interfaces are not included in the IS-IS flooding process, and as such, help to scale the IGP.

Deployment (Passive Interface)

- Typical situations where a passive interface is required:
  - Loopback interface, as this is merely a software interface that does not really carry user traffic, but is used as part of the control plane operations.
  - VLAN sub-interfaces attaching Layer 2 switches to Layer 3 routers for purposes of managing the switch.
- For both Cisco IOS and Juniper JunOS, a passive Loopback interface yields a metric of 0.

Deployment (Passive Interface)

- Cisco IOS sets a metric of 0 for passive physical interfaces, while Juniper JunOS sets the IS-IS default metric of 10 for the same.

Deployment (Passive Interface)

- IOS implementation:

```
router isis 1
 passive-interface Loopback0
```

Deployment (Passive Interface)

- JunOS implementation:

```
[edit protocols isis]
interface lo0.0 {
    passive;
}
```

Deployment (Adjacency Logging)

- Logging is great, at least for adjacency changes.

Deployment (Adjacency Logging)

- IOS implementation:

```
router isis 1
 log-adjacency-changes all
```

Deployment (Adjacency Logging)

- JunOS implementation:
  - No configuration required
  - Adjacency logging enabled by default

Deployment (Ignoring LSP Errors)

- The IS-IS protocol requires an LSP which arrives with an incorrect data-link checksum be purged by the receiving IS.
- LSP purging causes the initiating IS to regenerate that LSP.
- If perpetuated due to a bad link that causes data corruption, while still delivering LSP's with a correct checksum, this could overload the IS.
- So rather than purge them, ignore them.

Deployment (Ignoring LSP Errors)

- IOS implementation:

```
router isis 1
 ignore-lsp-errors
```

Deployment (Ignoring LSP Errors)

- JunOS implementation:

(No equivalent command)

Deployment (LSP Lifetime)

- This is the maximum amount of time LSP’s can remain in the link state database before they have to be refreshed.
- This value is set to 1,200 seconds (20 minutes) by default, for both Cisco IOS and Juniper JunOS.
- Setting this feature to its maximum value of 65,535 seconds (18.2hrs) would help reduce control traffic, making IS-IS more robust.

Deployment (LSP Lifetime)

- IOS implementation:

```
router isis 1
 max-lsp-lifetime 65535
```

Deployment (LSP Lifetime)

- JunOS implementation:

  ```
  [edit protocols isis]
  lsp-lifetime 65535;
  ```

Deployment (LSP Refresh)

- The LSP refresh interval determines the frequency of the flooding of LSP’s that carry routing topology information.
- Its purpose is to keep the link state database from becoming too old.
- This value is set to 900 seconds (15 minutes) by default for both Cisco IOS and Juniper JunOS.
- Both IOS and JunOS also include a random jitter timer of 25%.

Deployment (LSP Refresh)

- Setting this feature to a high value of 65,000 seconds (18hrs) reduces link utilization.

Deployment (LSP Refresh)

- IOS implementation:

```
router isis 1
 lsp-refresh-interval 65000
```

Deployment (LSP Refresh)

- JunOS implementation:

  (No equivalent command)

Deployment (PRC Delay)

- PRC (Partial Route Calculations) allows IS-IS (as well as OSPF) to scale in a large network.
- There are cases where an IS may signal the addition, deletion or change of metric of an IP prefix.
- Fundamentally, a routing protocol records the distance and direction of an IP prefix from each router. However, the actual prefixes, themselves, have no bearing on an SPF calculation.

Deployment (PRC Delay)

- After each node has been identified, and the cost to each node on the SPT has been determined, it’s simply a matter of recording what prefixes are attached to or announced by what node.
- So rather than unnecessarily re-run the SPF calculation, the other nodes on the SPT just record the change.
- Depending on the number of prefixes scanned, PRC can provide up to 10x the performance in processing hundreds of nodes.

Deployment (PRC Delay)

- <begin IGP War: IS-IS>

  !@#$%^&*()?><{}|\” “

  - PRC efficiency is more substantial in IS-IS than OSPFv2.
  - IS-IS announces all prefixes in IP Reachability TLV’s.
  - Node information (used for SPF calculations) is encoded in IS Neighbors or IS Reachability TLV’s.
  - Provides a clean separation between prefix information and topology information.
  - PRC easily applied to any IP address change.
- <end IGP War: IS-IS>

Deployment (PRC Delay)

- <begin IGP War: OSPFv2>

  !@#$%^&*()?><{}|\” “

  - OSPFv2 carries IP address information into Type 1 & Type 2 LSA’s.
  - Makes routers announce both their IP addresses and topology information in the same LSA’s.
  - A change in an IP address means a Type 1 LSA is originated. But because Type 1 LSA’s also carry topology information, a full SPF is run in the local OSPF area – unnecessary; only IP address is affected.
  - So only Type 3, 4, 5 and 7 LSA’s trigger PRC in OSPFv2, as their only purpose is to signal prefix information (external areas).
- <end IGP War: OSPFv2>

Deployment (PRC Delay)

- IOS implementation:

```
router isis 1
 prc-interval 5 50 100
```

Deployment (PRC Delay)

- JunOS implementation:

  (No equivalent command)

  (But see SPF Delay, ahead)

Deployment (SPF Delay)

- In an IS-IS domain, LSP’s will be flooded regularly due to random expiration of refresh timers around the network – this is when things are running smoothly.
- When “bad” things start happening in the network, routers can quickly become overwhelmed with processing LSP’s.
- If the LSP’s are arriving fast enough, and if they all require an incremental or full SPF run, a vast majority of CPU cycles would be expended on SPF.

Deployment (SPF Delay)

- The trick… how do you quickly react to the changing network without allowing SPF runs to dominate the router’s control plane? Well, tuning of SPF Delay, of course ☺ - consolidate SPF runs.
- Juniper JunOS implements a linear fast-slow delay scheme.
  - Normal period between SPF runs is short; 200ms by default.
  - If 3 (default) SPF runs are triggered successively, the delay period is automatically changed to 5,000ms by default.

Deployment (SPF Delay)

  - The router remains in this “slow” mode for 20 seconds since the last SPF run – means the network is now stable.
  - SPF Delay then switches back to “fast” mode.
- Cisco IOS initially implemented a similar fast-slow SPF Delay approach as Juniper, but now have taken an adaptive methodology and use a 3-stage exponential backoff algorithm.
  - Initial delay – the router waits for the initial delay period before running the first SPF.

Deployment (SPF Delay)

- Delay increment – further SPF runs are delayed by doubling the delay increment each time the SPF is run.
- Maximum delay – is the maximum value to which the delay can be incremented (light at the end of tunnel?).

Deployment (SPF Delay)

- IOS implementation:

```
router isis 1
 spf-interval 5 50 100
```

Deployment (SPF Delay)

- JunOS implementation:

  Default values found to be satisfactory

Deployment (LSP Generation Delay)

- Customization of the generation (not [re]transmission) of LSP’s is helpful in scaling IS-IS.
- Is important if multiple failures have occurred in a short space of time, e.g., failure of multiple links, failure of multiple nodes, e.t.c.
- Like the SPF and PRC Delay in IOS, an exponential backoff algorithm is used for LSP generation with Cisco.

Deployment (LSP Generation Delay)

- IOS implementation:

```
router isis 1
 lsp-gen-interval 5 50 100
```

Deployment (LSP Generation Delay)

- JunOS implementation:

  (No equivalent command)

Deployment (Fast Flood)

- Fast Flood tells the IS to flood a certain number of LSP's that invoke SPF, before actually running SPF.
- If you are running SPF and have configured aggressive values for the Initial delay (less than 40ms), SPF calculation may start before the LSP that triggered SPF is flooded to the IS's neighbors.
- The IS should "always" flood, at least, the LSP that triggered SPF before it runs the SPF calculation – improves convergence.

Deployment (Fast Flood)

- IOS implementation:

```
router isis 1
 fast-flood 10
!
! Original command was ‘ip fast-convergence’
! Is now deprecated since IOS 12.3(7)T
```

Deployment (Fast Flood)

- JunOS implementation:

  (No equivalent command)

Deployment (IETF Graceful Restart)

- When a router with a redundant control plane suffers a failure of one of them, it will generally lose any IGP adjacencies and BGP sessions to its neighbors.
- This will continue until the router has switched over to the redundant control plane (how [quickly] this occurs is unique to each vendor).
- During this time, all routes that were held in the FIB (in a distributed platform, normally on the line cards) will be deleted.

Deployment (IETF Graceful Restart)

- This leads to a catastrophic failure of the router not being able to forward any traffic. IETF Graceful Restart solves this problem.
- When supported by routing protocols, routers indicate their capability to "wait" for recovering routers to return to stability by continuing to hold and use routes they announced.
- The recovering router continues to process traffic as it "switches over" – stability!

Deployment (IETF Graceful Restart)

- IOS implementation:

```
router isis 1
 nsf ietf
```

Deployment (IETF Graceful Restart)

- JunOS implementation:

```
[edit routing-options]
graceful-restart;
```

Deployment (Ignore Attached Bit)

- In a multi-level IS-IS domain, the Attached bit is set by L1/L2 routers when they learn L1 routes in the same area across Level-2 backbones.
- The ATT-bit causes L1 routers to install a default route in their IS-IS RIB, as well as the FIB (assuming a “better” one does not exist).
- This signals to L1 routers that the closest L1/L2 router can be used to reach destinations unknown by the same Level-1 area.

Deployment (Ignore Attached Bit)

- This may lead to sub-optimal routing in cases where the /32 (v4) or /128 (v6) address of an IS needs to be reached across a Level-2 backbone.
- It may also lead to traffic being blackholed if a Level-1 area has multiple L1/L2 IS's. The L1 routers wouldn't really know which is the best.
- Recommended that the ATT-bit be ignored, in favour of Route Leaking (covered ahead)

Deployment (Ignore Attached Bit)

- IOS implementation:

```
router isis 1
 ignore-attached-bit
!
! This is a hidden and poorly documented
! command in IOS. But it works!
```

Deployment (Ignore Attached Bit)

- JunOS implementation:

```
[edit protocols isis]
ignore-attached-bit;
```

Deployment (IS-IS IPv6 Support)

- In Cisco IOS, most of the features enabled for IS-IS IPv4 (IS-ISv4) are consolidated and used for IS-ISv6 as well.
- IOS supports an IPv6 address family for IS-IS. Here, the only unique features that need to be configured are:
    - SPF Delay (similar to IS-ISv4)
    - PRC Delay (similar to IS-ISv4)
    - Multi-topology

Deployment (IS-IS IPv6 Support)

- In Juniper JunOS, with the exception of the IS-IS metric and multi-topology support, all other IS-ISv4 configurations apply to IS-ISv6.
- IS-IS has been extended to support IPv6 within the same implementation of the IS-IS protocol, through the use of new TLV’s.
- Single topology IS-IS is the default configuration. However, this restricts the network to ensuring both IPv4 and IPv6 topologies are congruent, i.e., the same.

Deployment (IS-IS IPv6 Support)

- During a dual-stack transition, IPv4 and IPv6 topologies are NOT congruent.
- This would lead to a complete failure of pre-established IS-IS adjacencies, as single topology IS-IS must be one of:
  - IPv4-only
  - IPv6-only
  - IPv4 & IPv6
- Multi-topology support for IS-IS removes this restriction.

Deployment (IS-IS IPv6 Support)

- Because IPv6 addresses are configured on an IS, one interface at a time, multi-topology permits pre-established IS-ISv4 adjacencies to stay up even when only one side of a link is being assigned an IPv6 address.
- This allows network operators to slowly enable IPv6 support in IS-IS, without affecting existing IPv4 traffic.

Deployment (IS-IS IPv6 Support)

- IOS implementation:

```
router isis 1
 !
 address-family ipv6
  multi-topology [transition]
  spf-interval 5 50 100
  prc-interval 5 50 100
 exit-address-family
```

Deployment (IS-IS IPv6 Support)

- JunOS implementation:

```
[edit  protocols isis]
topologies ipv6-unicast;
```

Deployment (L1/L2 IS’s)

- Recommended configurations for L1/L2 routers:
  - Route Leaking:
    - Recall the ATT-bit that L1/L2 routers send to L1 routers in the same area?
    - Recall that we said the ATT-bit should be ignored by all L1 routers?
    - Recall that we said optimal multi-level routing is achieved through Route Leaking?
    - Route Leaking permits L1/L2 routers to “leak” or install L1 routes learned from another Level-1 area into the local Level-1 area.
    - These are known as “IS-IS Inter-Area” routes.

Deployment (L1/L2 IS's)

- Route Leaking has the potential to cause routing loops, due to the flooding nature of link state routing protocols.
- To guard against this, the "Up/Down" bit is set each time a prefix is leaked into a lower level.
- Prefixes with the Up/Down bit set are **NEVER** propagated to an upper level.
- This prevents leaked prefixes from being re-injected into the backbone (Level-2).

Deployment (L1/L2 IS's)

- IOS implementation:

```
conf t
 ip prefix-list foo seq 10 permit 0.0.0.0/0 le 32
 !
 route-map FOO permit 10
  match ip address prefix-list foo
 !
 ipv6 prefix-list foo6 seq 10 permit ::/0 le 128
 !
 router isis 1
  redistribute isis ip level-2 into level-1 route-map FOO
  !
  address-family ipv6
   redistribute isis level-2 into level-1 distribute-list foo6
```

Deployment (L1/L2 IS's)

- JunOS implementation:

```
[edit policy-options]
policy-statement FOO {
    term 10 {
        from {
            protocol isis;
            level 2;
            policy isis-route-leaking;
        }
        to level 1;
        then accept;
    }
```

Deployment (L1/L2 IS's)

- JunOS implementation:

```
term 20 {
      from {
          protocol isis;
          level 2;
          policy isis-route-leaking6;
      }
      to level 1;
      then accept;
   }
}
```

Deployment (L1/L2 IS's)

- JunOS implementation:

```
policy-statement isis-route-leaking {
    term 10 {
        from {
            protocol isis;
            route-filter 0.0.0.0/0 upto /32;
        }
        then accept;
    }
    then reject;
}
policy-statement isis-route-leaking6 {
    term 10 {
        from {
            protocol isis;
            route-filter ::/0 upto /128;
        }
        then accept;
    }
    then reject;
}
```

Deployment (L1/L2 IS's)

- JunOS implementation:

```
[edit protocols isis]
export FOO
```

Deployment (Overload Bit)

- On core routers running iBGP that do not have redundant hardware properties, i.e., dual control plane, they should be configured to tell other IS's in the network not to use it as a transit path until it has fully recovered.
- Fully recovered = BGP has fully converged.
- Since IGP's will converge faster than BGP, the potential for blackholing traffic is high before BGP has fully converged on a recovering core router (the next-hop is already visible to other IS's).

Deployment (Overload Bit)

- This issue can be resolved with the use of the “overload bit” (a.k.a the hippity bit).
- The overload bit, when used, is set in the non-pseudonode LSP’s advertised to the network.
- This causes other IS’s to ignore the “unreliable” router, as a transit path, in their SPF calculations.
- However, all IP prefixes directly connected to the unreliable router are still reachable throughout the network.

Deployment (Overload Bit)

- IOS implementation:

```
router isis 1
 set-overload-bit on-startup wait-for-bgp
```

Deployment (Overload Bit)

- JunOS implementation:

```
[edit protocols isis]
overload timeout 600;
```

Deployment (Overload Bit - RR)

- Another reason to set the overload bit is when a network is using route reflectors for propagation of iBGP NLRI.
- Particularly, if route reflectors are positioned in such a way that they could potentially be in the forwarding path of transit traffic, within the core, setting the overload bit on them ensures they don’t advertise themselves as such.
- Ensures the route reflectors continue working only as route reflectors.

Deployment (Overload Bit - RR)

- IOS implementation:

```
router isis 1
 set-overload-bit
```

Deployment (Overload Bit - RR)

- JunOS implementation:

```
[edit protocols isis]
overload;
```

END

Thank you!

Q&A

mtinka@globaltransit.net