Segment Routing in Massively Scalable Data Center (MSDC)

By Soumik Bhattacharya (soumibha@cisco.com)

Agenda

- BGP in MSDC
- DC Architecture evolution
- MSDC Problem Statement/Requirements
- Solution brief : Segment Routing
- BGP Segment Routing 101
- Use Case #1 : MPLS Data Plane
- Use case #2 : IPv6 Data Plane
- References
- Appendices

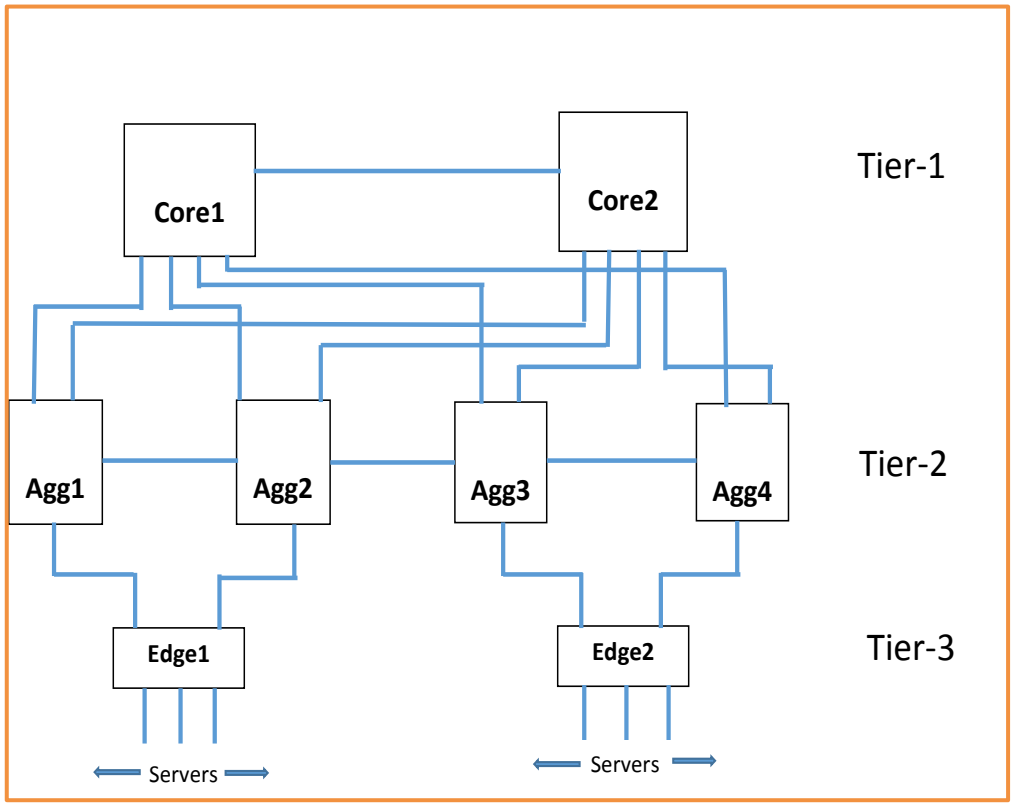
BGP in MSDC – base prerequisite

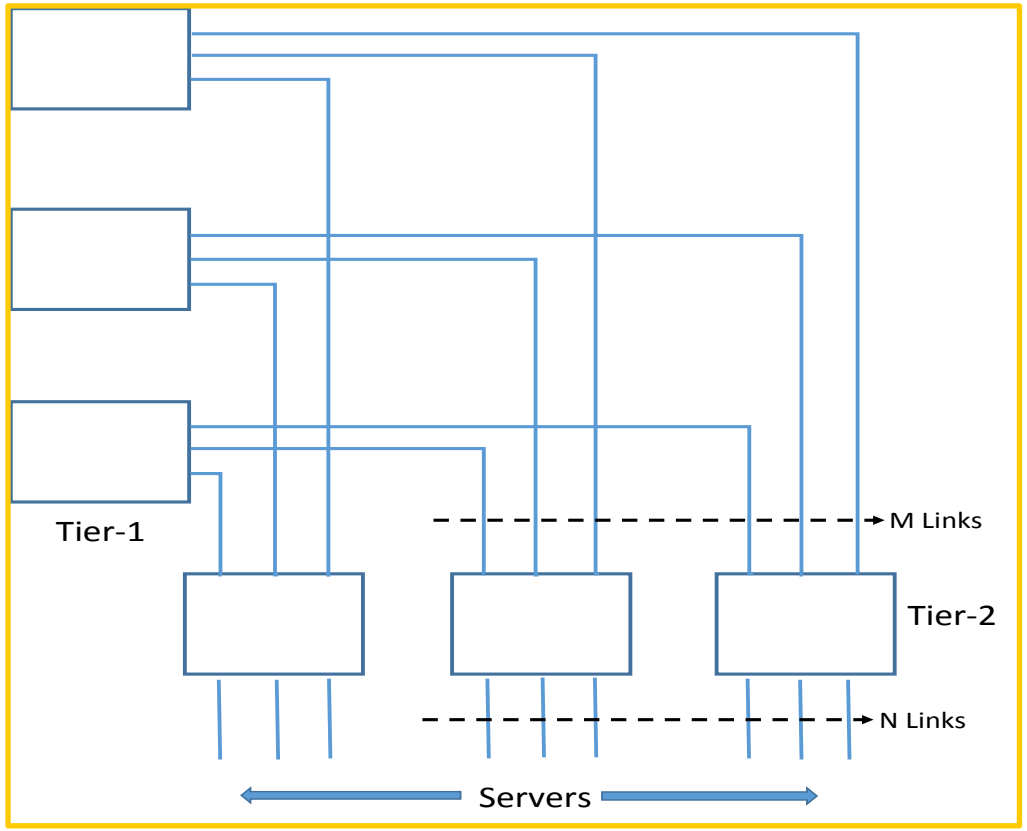
- draft-ietf-rtgwg-bgp-routing-large-dc
- Proposes EBGP as the routing protocol to be used with L3-only design
- EBGP caters to current and emerging DC requirements imposed by new technologies such as :
  - ✓ Big Data / Hadoop
  - ✓ Application-driven massive data replication among clusters
  - ✓ Virtual Machines (VMs) migrations
  - ✓ Increasing E-W traffic among servers

BGP in MSDC (Contd.)

- Why BGP :
  - ✓ Less complex protocol design (than IGPs, ex: TCP)
  - ✓ Less information flooding overhead
  - ✓ Supports manipulation of multipath with recursive nexthops.
  - ✓ Well-defined Autonomous System Number (ASN) scheme & AS_Path loop detection.
  - ✓ EBGP with minimal routing policy is easier to troubleshoot
  - ✓ Convergence time reduction by techniques like Bi-Directional Forwarding Detection (BFD)

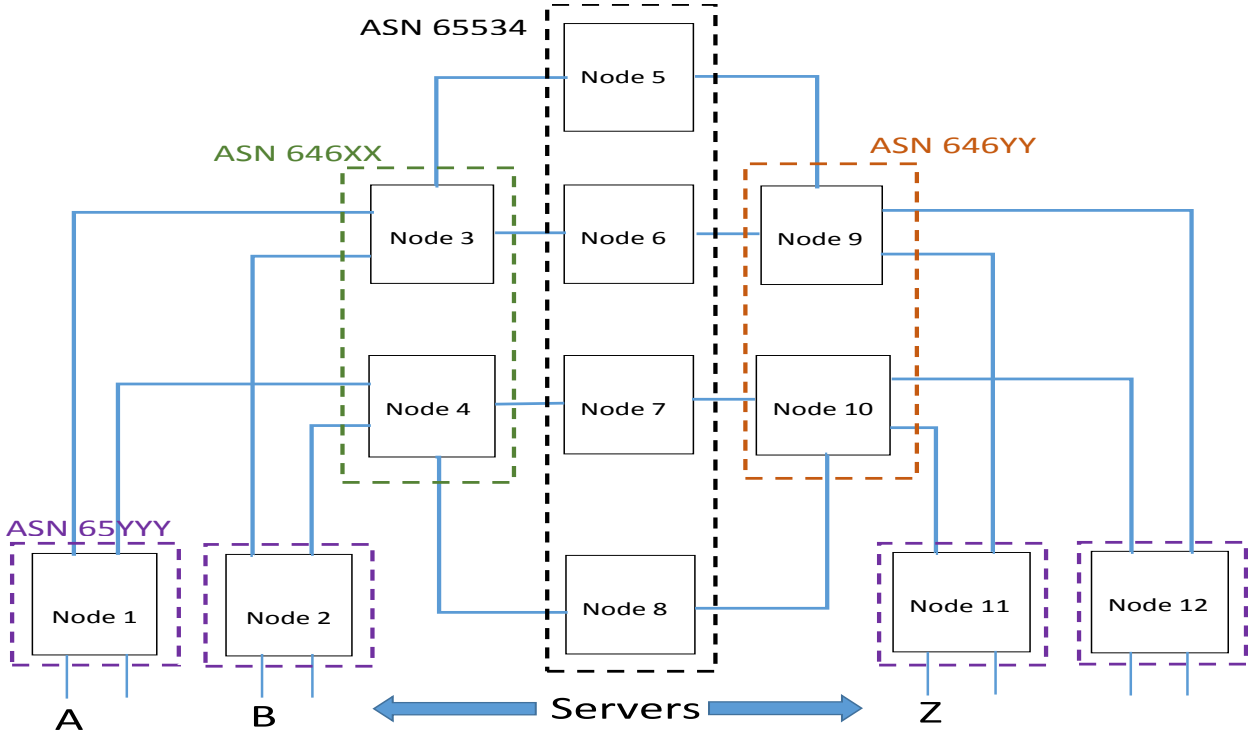
Evolution of classic “Inverted Tree” DC to CLOS / “Leaf & Spine”





- Fully non-blocking if M >= N or oversubscribed by N/M
- Server-to-Server (E-W) traffic is load-balanced using ECMP over all available paths
- Can both “scale up” (higher port-density) as well as scale horizontally (adding more stages)
- Helpful in catering increase of E-W traffic (among servers)

BGP in MSDC



- Each T3/TOR, each Cluster of T2/Leaf and all T1/Spine are in their own AS
- Private use ASNs (64512 – 65534) recommended to avoid ASN conflicts
- Hop-by-hop EBGP “underlay” among peer nodes (can use IBGP + RR w/ cluster-id replacing ASN)
- ECMP fan-out over all available paths between servers

MSDC Requirements

- ✓ Per-packet/per-flowlet ECMP routing (vis-à-vis traditional per flow ECMP routing – “elephant & mouse flows” problem)
- ✓ Performance-aware routing and dynamic fault avoidance (vis-à-vis network-imbalance-oblivious ECMP SPF – prominent during failures)
- ✓ Path visibility (“white-box view” vs “black-box view” of network)
- ✓ Deterministic network probing

Elephants vs Mice flows

- **Elephant flows :**

→ Very Large size (total # of bytes)

→Long-lived

→Typically large transfers, throughput-sensitive

Example : DB cloning, backup stored data, VM migration etc.

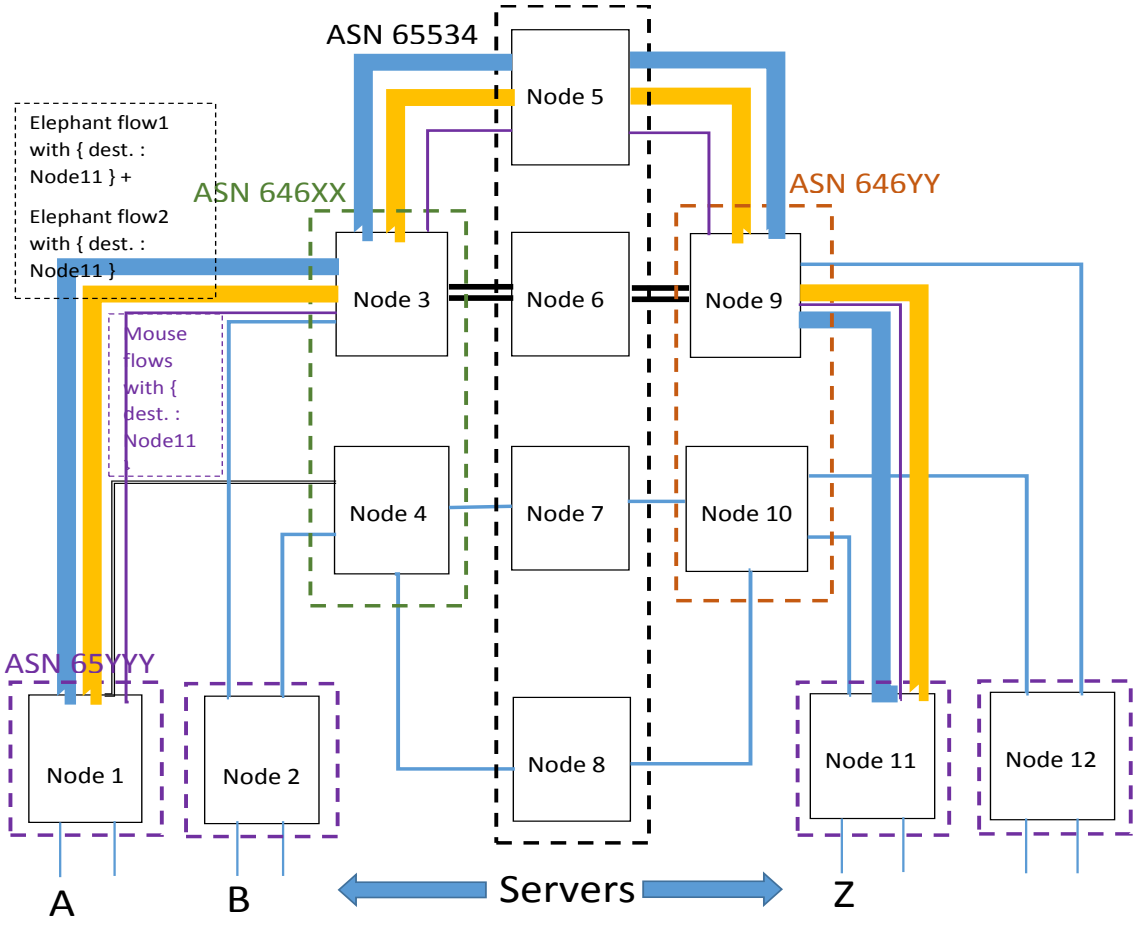
- **Mice flows :**

→ Smaller size

→ Short-lived

→ Typically bursty, latency-sensitive

Example : Web browsing, emails, social media posts etc.



- **Problems :**

→ Majority of flows are “mice” in DC, but majority of packets belongs to few “elephants”

→ Per-flow ECMP causes all flows with { dest. : Node11 } to be hashed via same ECMP path(s)

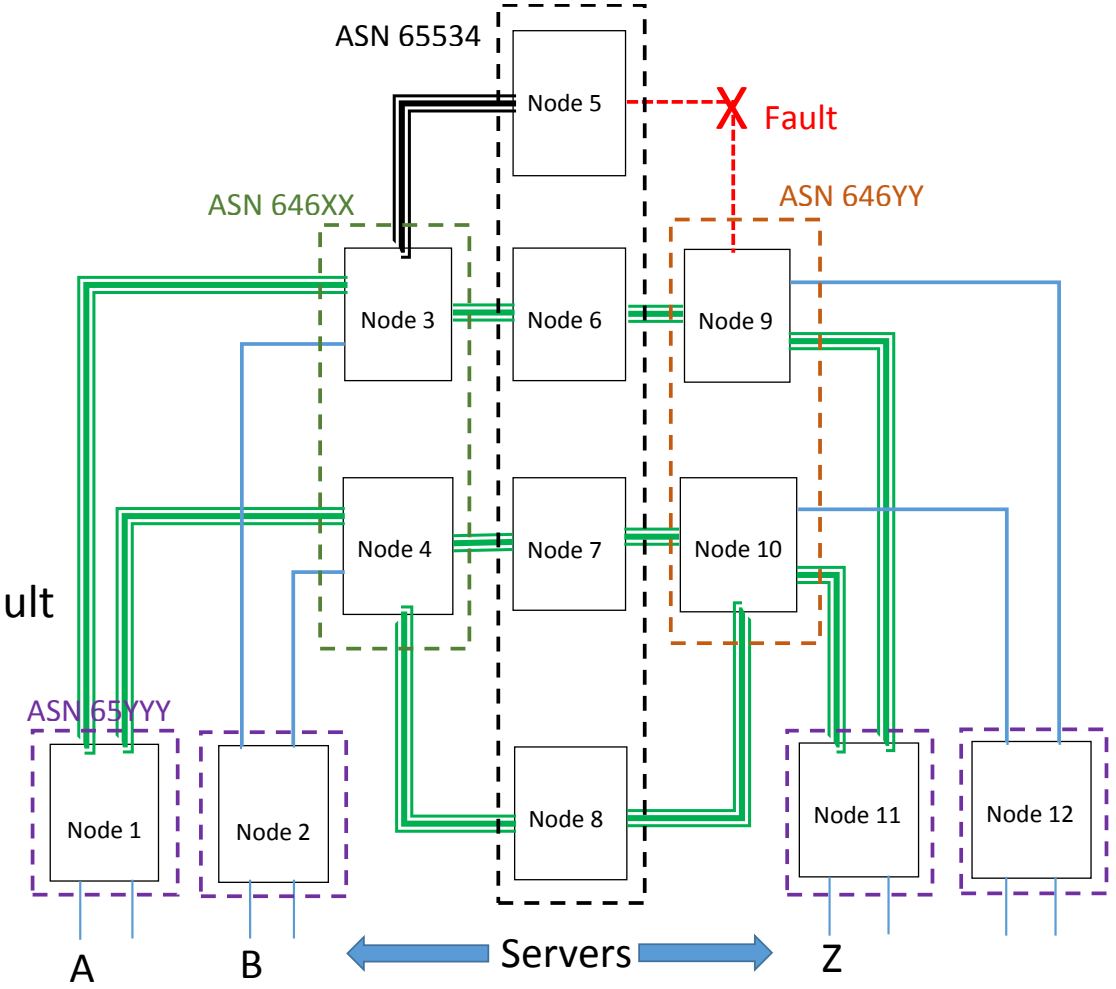
→ “elephant” flows eat up network buffers, introducing queueing delay for latency-sensitive “mice” flows

→ Adaptive routing techniques not very efficient for bursty “mice” flows

→ Hash-function (traditionally 5-tuple based) inefficiency causing several “elephants” to use same link(s), while other link(s) are unused

Network Imbalance-Oblivious ECMP SPF

Node 1 is oblivious of fault between Nodes 5 & 9



- **Problem:** Node 3 has alternate path towards Node11 → Node 1 is unaware of fault between 5 & 9 →keeps load-balancing A – Z traffic *equally* over Nodes 3, 4

Solution brief – Segment Routing basics

- SPRING (Source Packet Routing In NetworkingG)
- Source steers packets thru ordered list of instructions (local/ global, topology/service-based) :
  - ✓Enforced flow through any path & service chain
  - ✓Per-flow state only at the ingress node.
  - ✓No need for LDP/RSVP-TE. IGP/BGP-based control plane.
  - ✓Allows incremental deployment
- Applicable seamlessly to MPLS and IPv6-based data plane.

Solution brief – Segment Routing definitions

- Segment : Instruction executed on incoming packet by a node (e.g. push/pop/swap MPLS label, decrement header and modify IPv6 DA)
- Segment Identifier (SID) : Number identifying a segment(e.g. MPLS Label/IPv6 Address)
- Segment List : Ordered SID list encoding source route (e.g. MPLS Label stack / IPv6 address list)
- Global/Local SID : Scope of SID – supported by all capable nodes vs. only the originator node

SR Solution brief – Segment/SID types

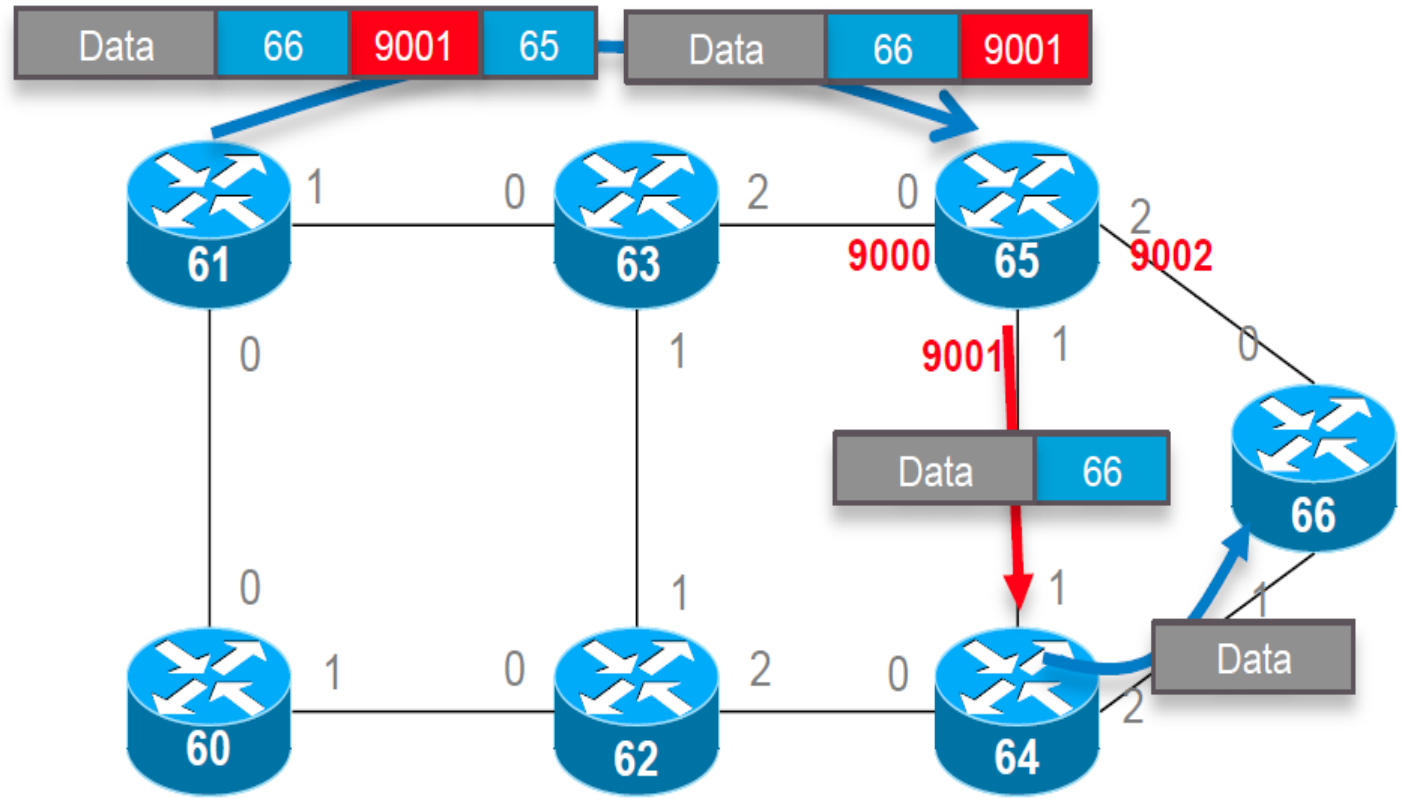
A. Prefix SID : SID attached to a prefix (Global)

→Subtypes :

- ➢Node SID : Identifies a specific router (typically a loopback address)
- ➢Anycast SID : Identifies set of routers (attached to anycast IP)

B. Adjacency SID : SID attached to an interface/Link/Service in a specific node

SR Solution brief – Sample topology (showing Node and Adjacency SIDs)



63

Node (Prefix) SID

9001

Adjacency SID

MSDC Solution brief – BGP SR

- BGP Prefix Segment IDentifier (Prefix-SID) :

draft-ietf-idr-bgp-prefix-sid-01 (SR Prefix SID extensions for BGP)

- BGP Prefix-SID :

→Advertised by BGP

→Global within the BGP domain/AS

→Identifies ECMP-aware shortest-path computed by BGP to the prefix

→Used primarily as Node-SID where the prefix identified is the node identifier (typically lo0 address/router ID)

→Represented either as a label (MPLS data plane) or IPv6 address (IPv6 data plane)

Protocol Extensions for SR – BGP Prefix SID attribute

- BGP Prefix-SID Attribute (optional, transitive) :

Used with AFI/SAFI :

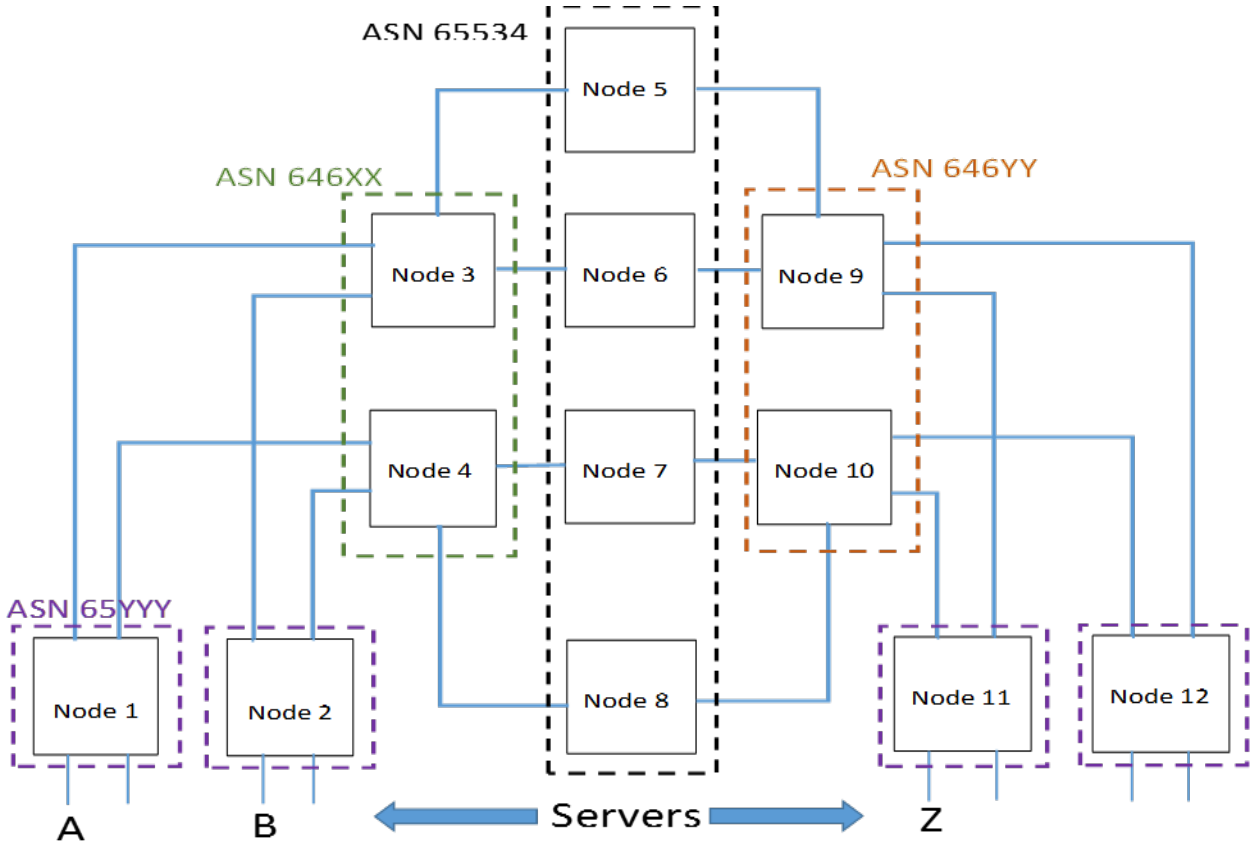
→ MP-BGP Labelled IPv4/IPv6 unicast

- Label-Index TLV
- Originator SRGB TLV (Optional)
- The computed local label is either min(SRGB) + Label-index, or a dynamic RFC3107 label if Label-index > SRGB size OR if SR is not supported in the node (useful for incremental deployment)

→ BGP IPv6 unicast with Segment-Routing Header

- IPv6 SID TLV (with IPv6 Data-plane) with S-flag to advertise IPv6 Segment Routing Header (SRH) processing capability

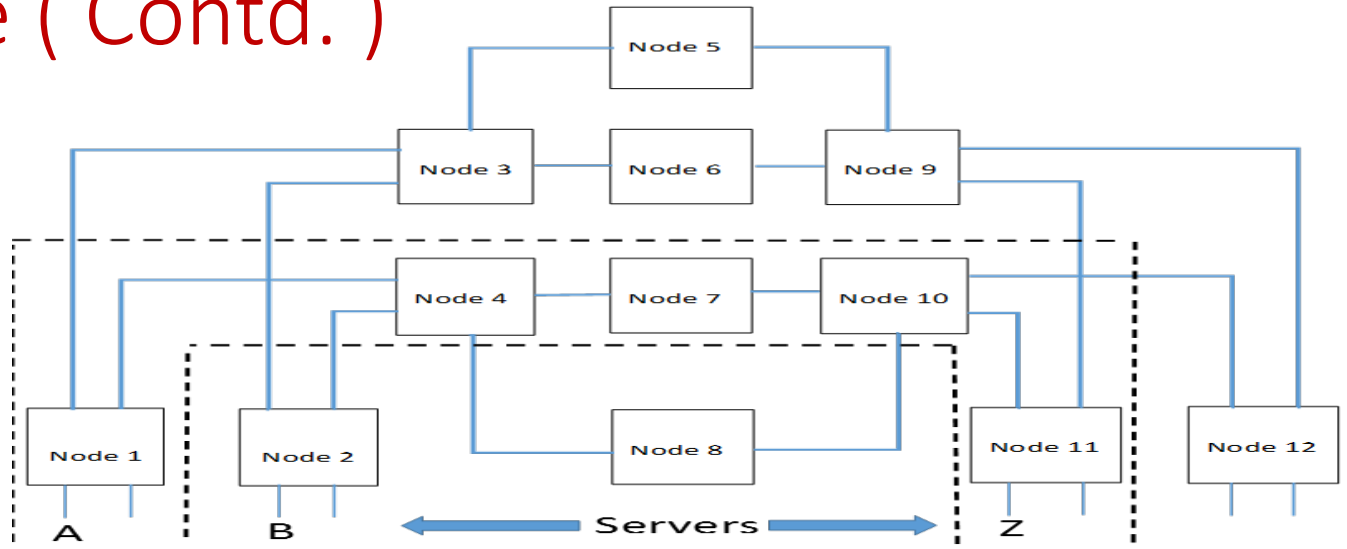
MPLS Use Case – Massive Scaled Data Center (MSDC)



- Each TOR node in own AS, Leaf Nodes in same cluster in same AS, all Spine nodes in same AS
- EBGP Labeled Unicast (RFC3107) hop-by-hop everywhere
- Can be applied to IBGP as well with ASN allocation replaced by cluster id allocation, route reflection and next-hop-self

SR MSDC MPLS Use Case (Contd.)

- For the following flow A - Z :
- The loopback of any Node x is 192.0.2.x/32



- Node11 → Node10 BGP Labelled Unicast update :
  - NLRI: 192.0.2.11/32, Label: Implicit-Null
  - Next-hop: Node11-Node10 interface in Node11
  - Attribs. : AS Path = [11], BGP-Prefix Attribute = label-Index 11

- Node10 → Node7 BGP LU update :
  - NLRI: 192.0.2.11/32, Label: Node10 Local SRGB + 11, NH: Node10-7 interface, Attribs. : AS Path = [10,11], Index 11

- Node7 → Node4 BGP LU update :
  - NLRI: 192.0.2.11/32, Label: Node7 Local SRGB + 11, NH: Node7-4 interface, Attribs. : AS Path = [7,10,11], Index 11

- Node4 → Node1 BGP LU update :
  - NLRI: 192.0.2.11/32, Label: Node4 Local SRGB + 11, NH: Node4-1 interface, Attribs. : AS Path = [4,7,10,11], Index 11

- Node1 FT : In Label/IP : Node1 SRGB + 11 / 192.0.2.11/32, Out Label : Node4 SRGB + 11, Out I/F : ECMP{3,4}
- Node4 FT : In Label/IP : Node4 SRGB + 11 / 192.0.2.11/32, Out Label : Node7 SRGB + 11, Out I/F : ECMP{7,8}
- Node7 FT : In Label/IP : Node7 SRGB + 11 / 192.0.2.11/32, Out Label : Node10 SRGB + 11, Out I/F : 10
- Node10 FT : In Label/IP : Node10 SRGB + 11 / 192.0.2.11/32, Out Label : POP for PHP / N/A, Out I/F : 11

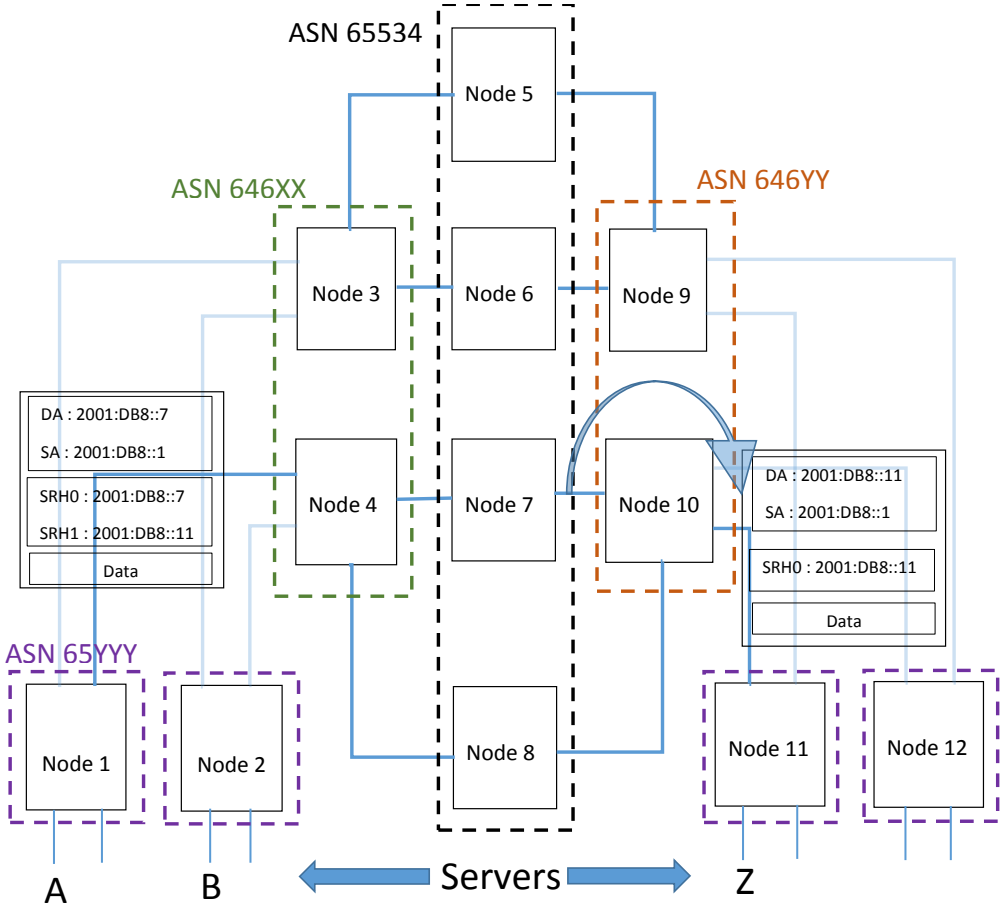
→ Labels can be stacked to steer traffic including (or excluding) certain nodes/links.

IPv6 Use Case

- Uses IPv6 segment (node address) stack instead of MPLS label stack
- Requires the IPv6 Segment Routing (Extension) Header (SRH) as described in draft-ietf-6man-segment-routing-header-00
- The SR Source Node originates IPv6 Prefix with SRH capability advertised in the Prefix SID attribute
- Ingress node uses SRH to stack up node IPv6 addresses in the computed path with the Source node SID as the last, and Destination Address (DA) same as the first/top one in the stack.
- Transit nodes with address = DA of received packet would inspect SRH and decrement it with resetting DA = current top of stack node address.
- When the packet reaches the Source node no further segments (node addresses) are left in the SRH.

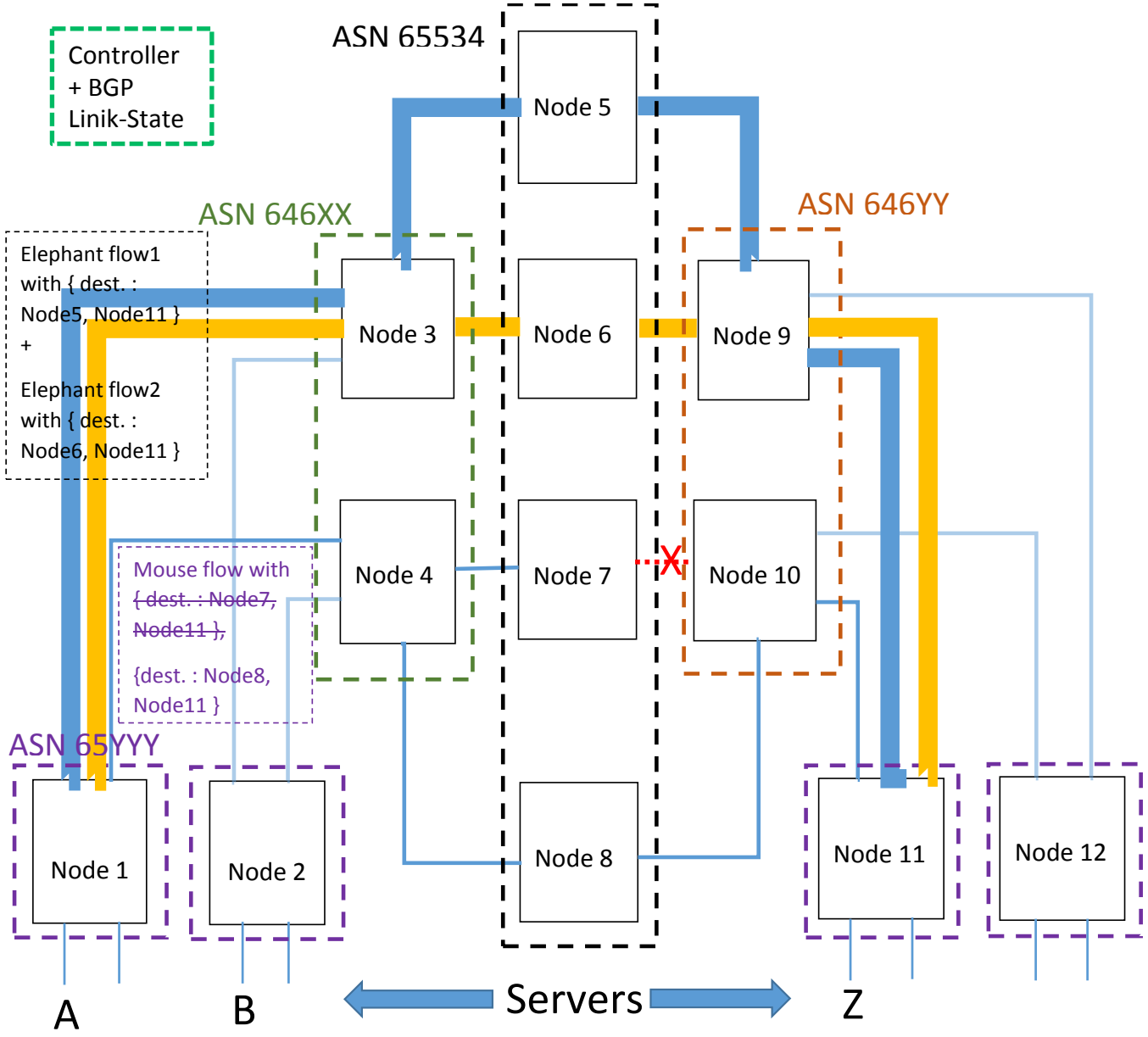
IPv6 Use Case - Example

- Spine node 7 : IPv6 lo0 address 2001:DB8::7
- TOR node 11 : IPv6 lo0 address 2001:DB8::11
- Both advertises these prefixes with IPv6 SRH capability via BGP
- an application in host A can send traffic to host Z via node 7 by adding DA & top of SRH segment = 2001:DB8::7, next (last) SRH segment = 2001:DB8::11



- When packets reach node 7 via ECMP, it sees it's own IPv6 lo0 address at top pf SRH → decrements SRH with new SRH top segment = 2001:DB8::11 → copy new top SRH onto DA of forwarded packet
- Packet reaches node 11 where it sees it's own IPv6 lo0 address as DA → decrements SRH → No more segments in SRH → absorbs packet

Looking back at the problems : Solution



- Per-flowlet or per-packet ECMP because of more granular view of the path to destination
- Hashing not dependent on destination node only
- Can pin flows to a more specific path and collect info such as packet loss either at the end of flow (mouse) or periodically (elephant)
- Dynamic fault avoidance possible by assigning negative preference to paths with loss
- Non-oblivious routing integrating with a centralized controller (SDN)
- Fast correlation & detection of failed paths by prescribing exact paths to series of probing agents at a time
- Knowledge of exact links/nodes traversed by the probe helps to isolate faulty or degraded sections

References

- www.segment-routing.net
- Current IETF drafts for SR : http://www.segment-routing.net/home/ietf
- BGP in Large Scale DCs : https://tools.ietf.org/html/draft-ietf-rtgwg-bgp-routing-large-dc-07
- BGP SR Prefix SID extension : https://tools.ietf.org/html/draft-ietf-idr-bgp-prefix-sid-01
- BGP-Prefix Segment in large-scale data centers : https://tools.ietf.org/html/draft-ietf-spring-segment-routing-msdc-00
- Carrying Label info in BGP (RFC 3107) : https://tools.ietf.org/html/rfc3107
- IPv6 SR Header : https://tools.ietf.org/html/draft-ietf-6man-segment-routing-header-00

Appendix #1

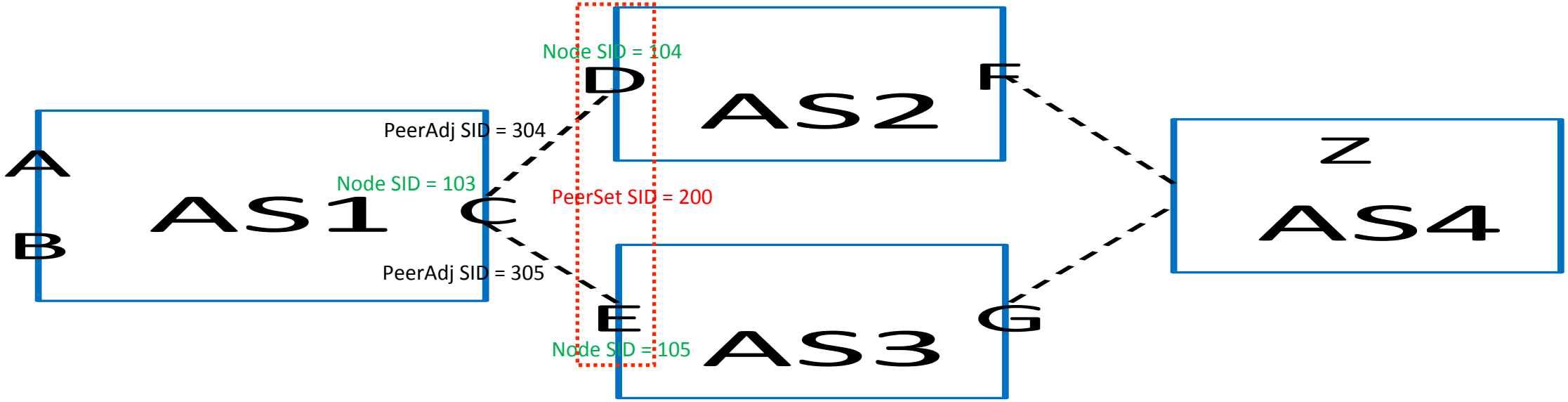
- BGP Egress Peer Engineering (EPE) SIDs :

→Local SIDs in egress ASBR advertised to ingress ASBR

→Steers traffic to specific egress ASBR (BGP node-prefix SID) + nexthop peering SID details :

  - ➢ PeerNode SID : attached to connected peering node of the egress ASBR
  - ➢ PeerAdj SID : attached to the peering node and interface of the egress ASBR
  - ➢ PeerSet SID : to load-balance across any connected interface to any peer in the set

Appendix #2 : Solution brief – BGP EPE example



- At C : own Node SID = 103 (advertised with C lo0 IP)
- At C : for D, PeerNode SID = 104, PeerAdj SID = 304, PeerSet ID = 200
- At C : for E, PeerNode SID = 105, PeerAdj ID = 305, PeerSet ID = 200
- C advertises these to A or B in AS1, A/B steer traffic to D [103,304] or E [103,305]
- Local FRR protection at C can be provided by using the [Node SID, PeerSet ID] = [103,200] where CE link can protect CD link failure and vice-versa.

Appendix #3 : Protocols Extensions for SR – BGP-LS for EPE

- Adds following Sub-TLVs to Link NLRI which encodes Peer-Node/Peer-Adj/Peer-Set Segments :
  - Local Node Descriptors (used for Peer-Node Segment and Peer-Adj Segment) :
    - →BGP Router ID Sub-TLV
    - →ASN Sub-TLV
    - →Member ASN Sub-TLV (when Confederation is used for IBGP)
  - Remote Node Descriptors (used for Peer-Node Segment and Peer-Adj Segment) :
    - → BGP Router ID Sub-TLV
    - → ASN Sub-TLV
    - → Member ASN Sub-TLV (when Confederation is used for IBGP)
  - Link Attributes (used for Peer-Node, Peer-Adj and also for Peer-Set Segments) :
    - → Adj-SID TLV
    - → Peer-SID TLV
    - → Peer-set-SID TLV

Color-based architecture using Anycast SIDs & Node SIDs