Global Load Balancing (GLB) in CLOS Fabric

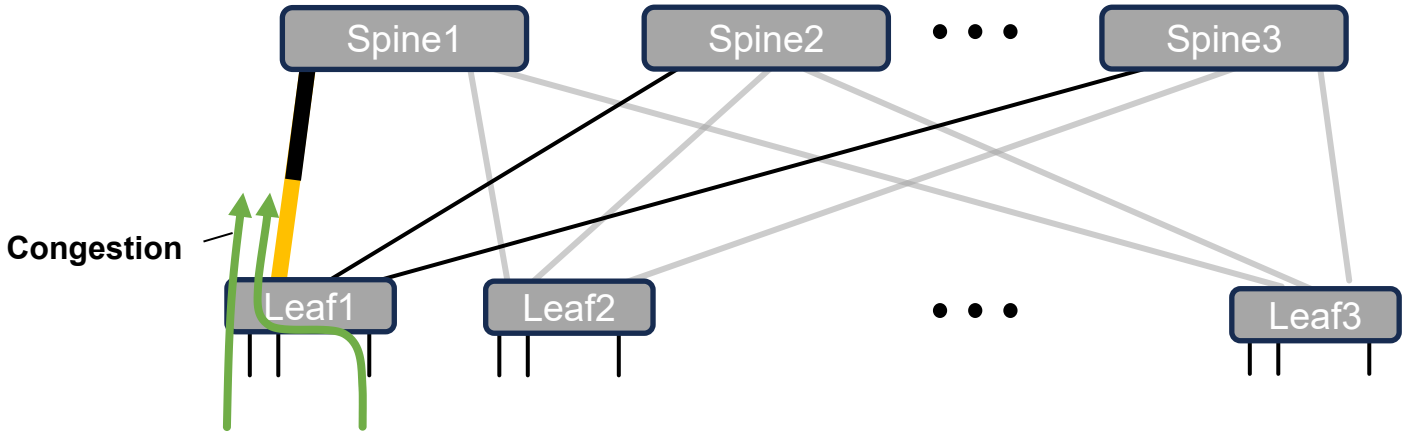
Jeffrey Zhang, HPE

FANTEL BoF
IETF123, Madrid

Introduction

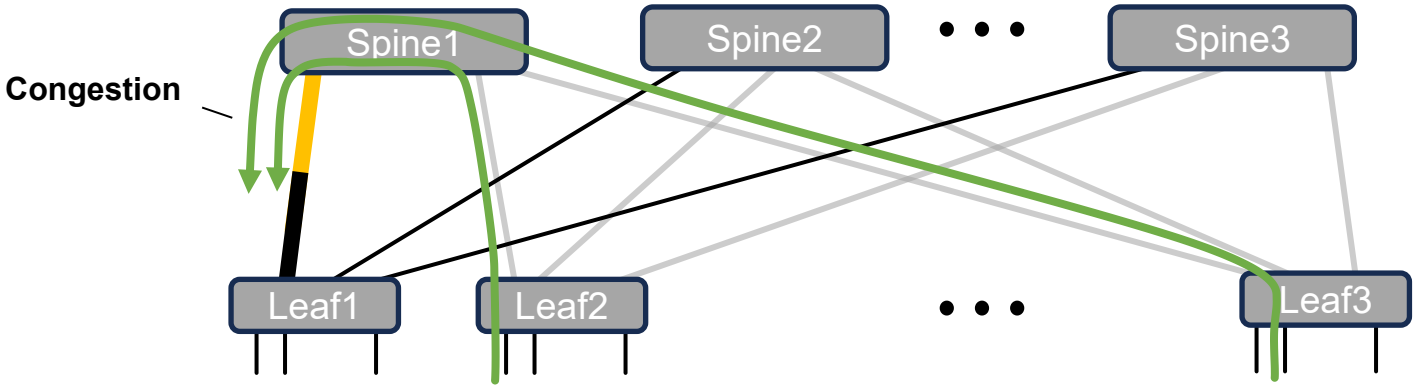
- Data Centers (DC) usually use CLOS IP fabrics with EBGP
- AI/ML traffic needs low latency, lossless capabilities
- AI/ML flows usually have high throughput and low entropy, which could cause fabric congestion.
  - Sub-millisecond mitigation anywhere in the fabrics is desired
- Dynamic Load Balancing based on link load information helps
  - Both local and remote links

Local Congestion



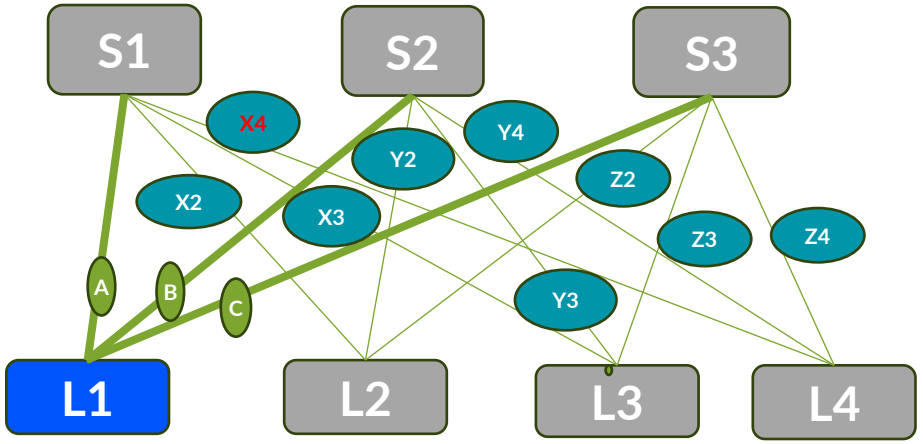
- Random ECMP hashing can't guarantee each link has the same load
- When congestion is detected locally (Leaf1 to Spine1), Dynamic Load Balancing (DLB) can help to mitigate the congestion
  - Leaf1 avoids using Spine1

Remote Congestion



- When Congestion happens on a remote link (Spine1 to Leaf1), DLB can't help on Leaf3
- Global Load Balancing (GLB) is needed to mitigate remote congestion
  - Leaf3 realizes that Spine1-Leaf1 link is congested so it sends traffic to Spine2

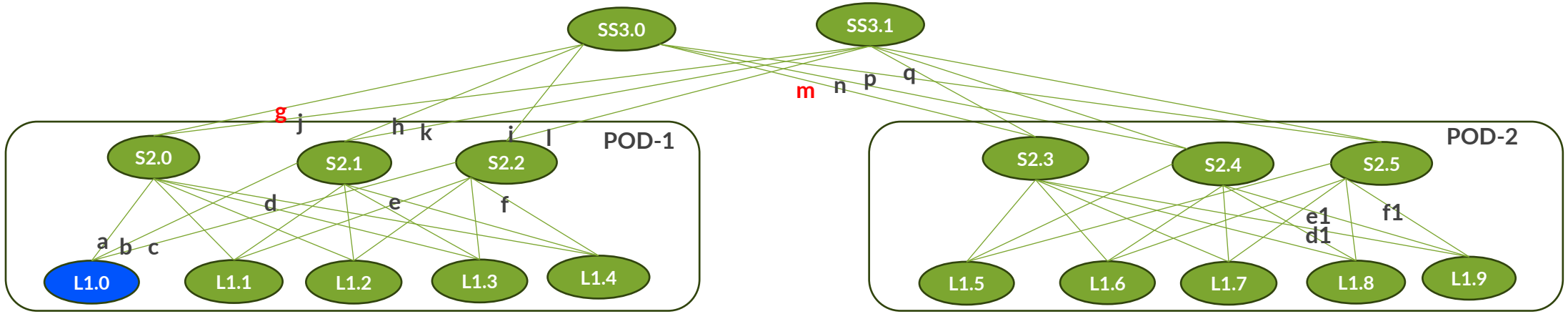
GLB: 3-CLOS



Routes on L1		
Destination	Next-hop	Next-next-hop
L2	A	X2
	B	Y2
	C	Z2
L3	A	X3
	B	Y3
	C	Z3
L4	A	X4
	B	Y4
	C	Z4

- For routes received from L2, L3, L4, L1 keeps track of the <next-hop, next-next-hop> tuple for each path
- Each spine broadcast its link qualities to all leaves
- L1 uses the combined link qualities of the <next-hop, next-next-hop> tuples for load balancing decisions

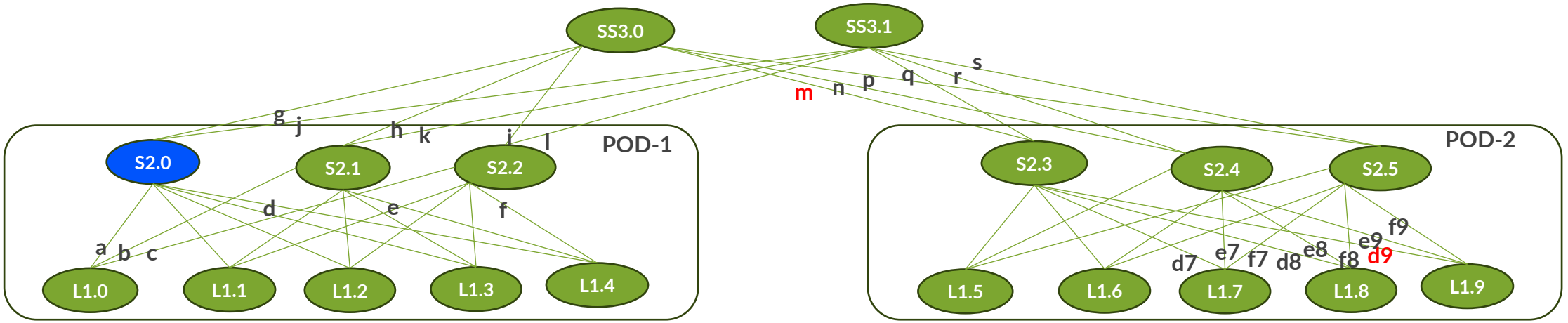
GLB: 5-CLOS (1/2)



- For routes received from leaves in the same pod, POD-1, GLB is done same as 3-CLOS
- For routes received from leaves in the other pod, POD-2, there are more than 2 hops, but we still just need to keep track of the next-hop and next-next-hop
  - When L1.0 knows link *g* is congested, it will try to avoid hashing L1.8 and L1.9 flows towards link *a* to mitigate the congestion on link *g*
  - L1.0 does not care about the congestion on link *m* because the congestion of link *m* affects the load balancing decisions of S2.0, S2.1, S2.2 equally. So L1.0 would still hash flows to S2.0, S2.1, S2.2 same as before

Routes on L1.0		
Destination	**Next-hop**	**Next-next-hop**
L1.4	a	d
	b	e
	c	f
L1.8	a	g, j
	b	h, k
	c	i, l
L1.9	a	g, j
	b	h, k
	c	i, l

BGP GLB: 5-CLOS (2/2)



- Unlike 3-CLOS where next-next-hops only exist on leaves, 5-CLOS has next-next-hops on spines and super spines too
- Next-next-hop link qualities are also tracked on spine S2.0
  - When S2.0 knows link *m* is congested, it will try to avoid hashing L1.7, L1.8, L1.9 flows towards link *g* to mitigate the congestion on link *m*
  - S2.0 does not care about the congestion on link *d9* because the congestion of link *d9* affects the load balancing decisions of SS3.0, S3.1 equally. So S2.0 would still hash flows to SS3.0, SS3.1 same as before

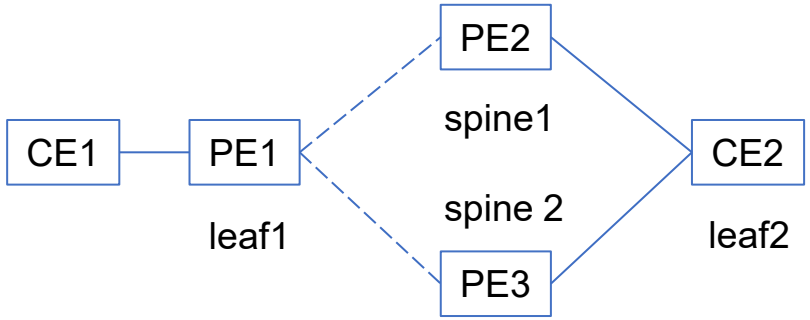
Routes on S2.0		
Destination	**Next-hop**	**Next-next-hop**
L1.7	**g**	**m, n, p**
	j	**q, r, s**
L1.8	**g**	**m, n, p**
	j	**q, r, s**
L1.9	**g**	**m, n, p**
	j	**q, r, s**

GLB: n-CLOS

- Due to the special topology of the CLOS networks, link congestions only affect the load balancing decisions of the local node and the previous-hop nodes
- Each node only needs to keep track of the <next-hop, next-next-hop> for each route for GLB purpose
- Congestions beyond the next-next-hop do not affect GLB
- This makes it possible for non-link-state protocols, such as BGP, to figure out the topology for GLB
- A node only needs to broadcast link quality information towards its direct neighbors
- There is no need to propagate link quality information beyond the direct neighbors, therefore there is no scaling issues like IGP flooding

Use Case for Overlay Services

- For overlay services, the ingress PE, egress PE, and egress CE can be viewed as a 3-CLOS at the *overlay* level
- The dynamic egress PE-CE link load info can be propagated to the ingress PEs to allow dynamic GLB via multihoming egress PEs



Summary

- For GLB, remote link load information is needed
- For CLOS, the link load information only needs to be propagated one hop away
- The propagation needs to be fast for sub-millisecond congestion mitigation
  - Preferably handled in the forwarding path