The bridge to possible

Multi-Tier Fabrics

Network Designs for the Modern Data Center

Marina Ferreira – Builder of Things
@_marinalf

BRKDCN-2999

cisco Live!

Cisco Webex App

https://ciscolive.ciscoevents.com/ciscolivebot/BRKDCN-2999

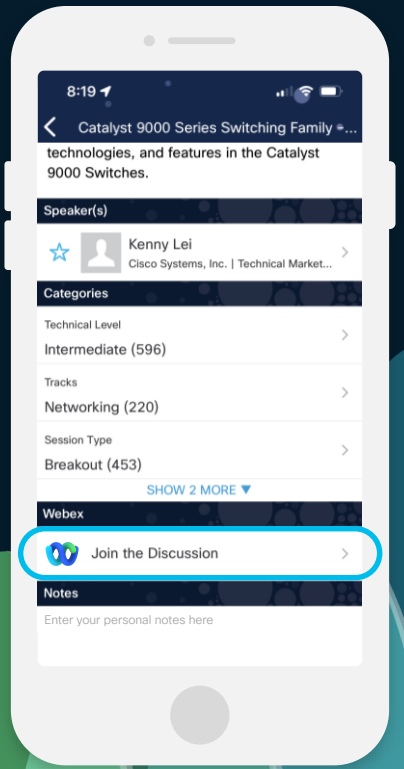
Questions?

Use Cisco Webex App to chat with the speaker after the session

How

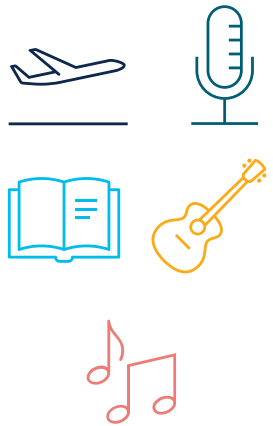
1. Find this session in the Cisco Live Mobile App
2. Click “Join the Discussion”
3. Install the Webex App or go directly to the Webex space
4. Enter messages/questions in the Webex space

Webex spaces will be moderated by the speaker until June 7, 2024.



About The Speaker

- 15+ times in Vegas: Tourist, Partner, or Cisco
- Cisco Live attendee since 2014, and speaker since 2022
- DevNet Advocate
- Reader, Occasional Writer
- Absolute Beginner Guitar Player
- Adele’s Fan!



Abstract

Have you ever asked yourself what "Clos" is or where that Leaf/Spine thing comes from? If yes, this is the right session for you. We are going to cover Fat-Tree, Clos and Leaf/Spine designs and expand beyond just the Spine layer. We will spend some time on the Super-Spine and even Super-Spine fabrics. How you can cost-effectively use 100G/400G and where fixed vs. modular Switches make sense.

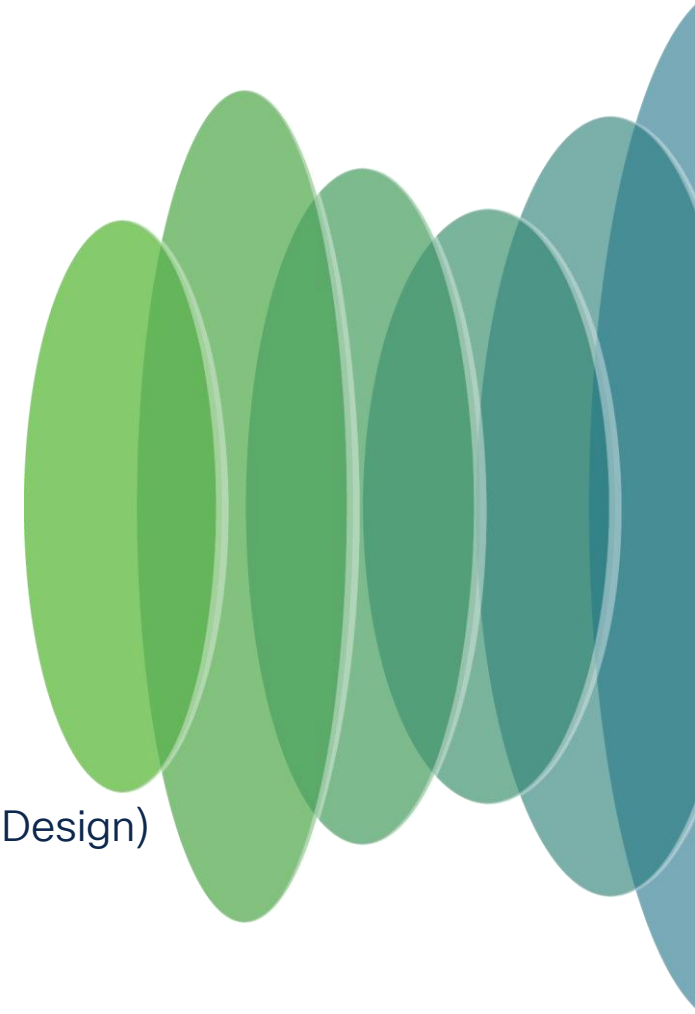
[Special thanks to **Lukas Krattiger** and **Max Ardica** for the original session idea and inspiration]

Agenda

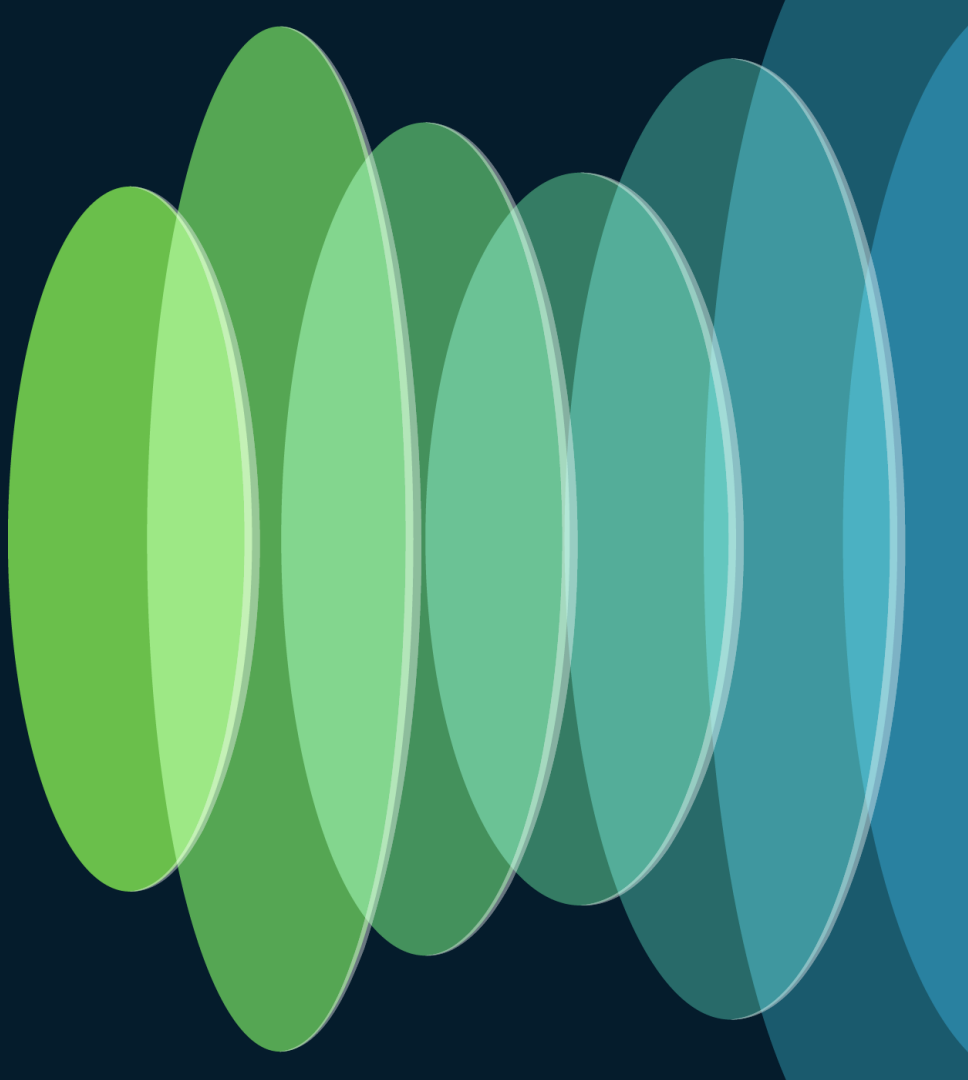
- Introduction
- Paradigm and Fundamentals
  - Leaf-Spine Topologies (2-Tier, 3-Stages)
  - Sizing Considerations
- Design Evolution
  - Building Distributed Architectures
  - Super-Spine and Spine Plane (5-Stages)
- Mapping to Architectural Options
  - ACI, VXLAN EVPN, and Heterogeneous Fabrics
- Routed Fabrics
- Conclusion

"The place to begin in the data center design is not with the physical cabling, or cooling, or power, but with the concept of a fabric..."

Book: The Art of Network Architecture (Business-Driven Design)
Authors: Russ White, Denise Donohue

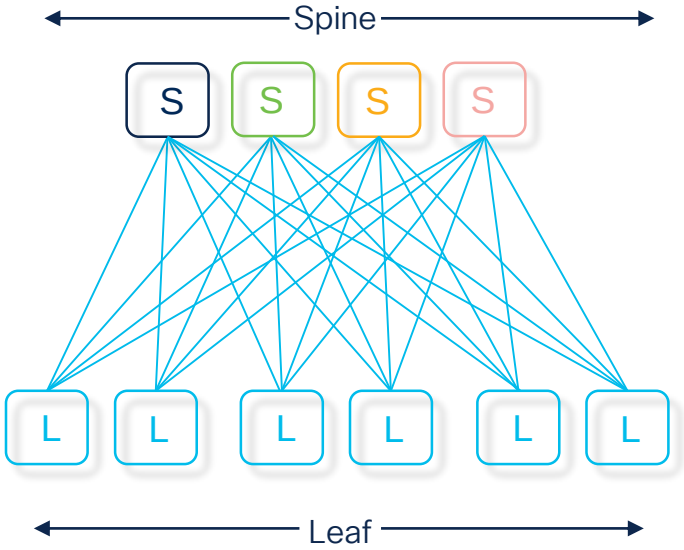


Paradigm and Fundamentals



The Paradigm

A Leaf and Spine Topology

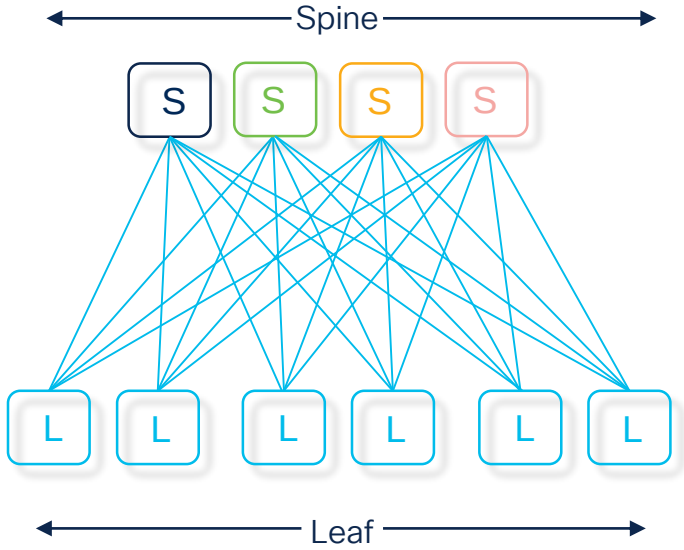


The Paradigm

A Leaf and Spine Topology

Variations

- Fat Tree
- Folded Clos
- 3-Stage Clos
- 2-Tier Network



History

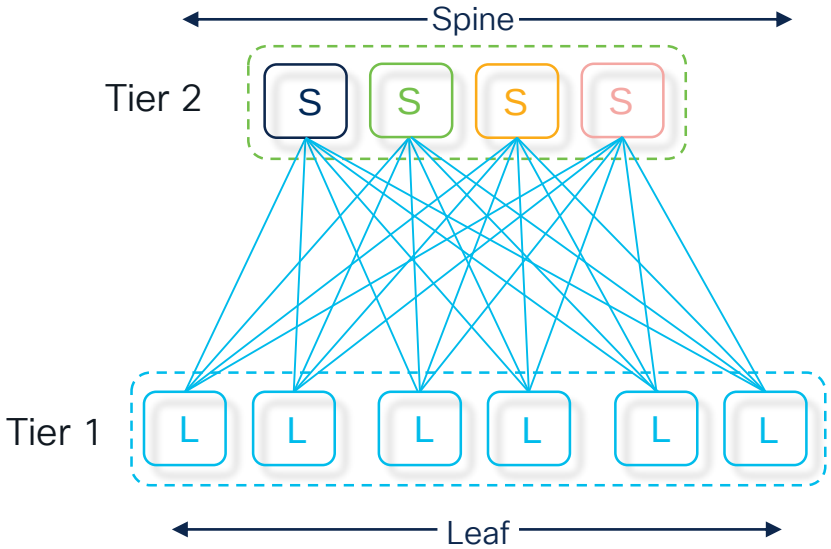
Invented by Edson Erwin – 1938

Formalized by Charles Clos – 1953

The Paradigm

A Leaf and Spine Topology

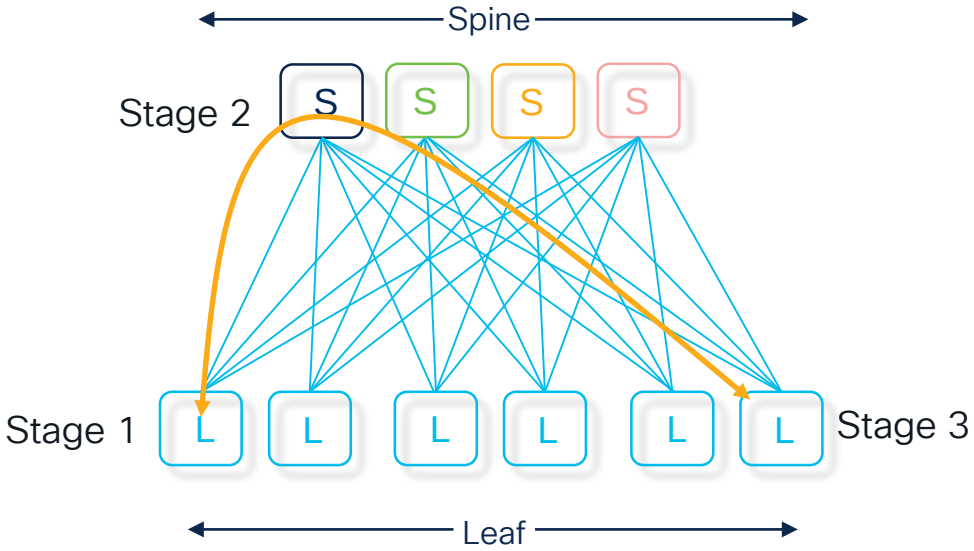
2 Tiers
3 Stages



The Paradigm

A Leaf and Spine Topology

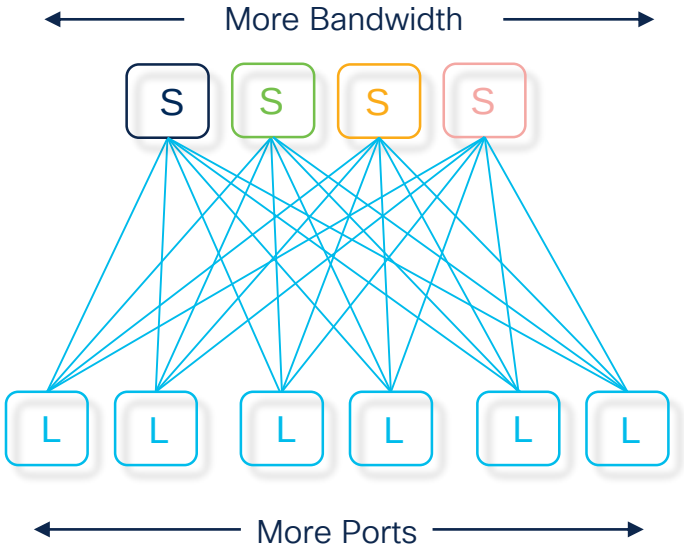
2 Tiers
3 Stages



The Paradigm

A Scale Out Architecture

More Leaf = More Ports
More Spine = More Bandwidth



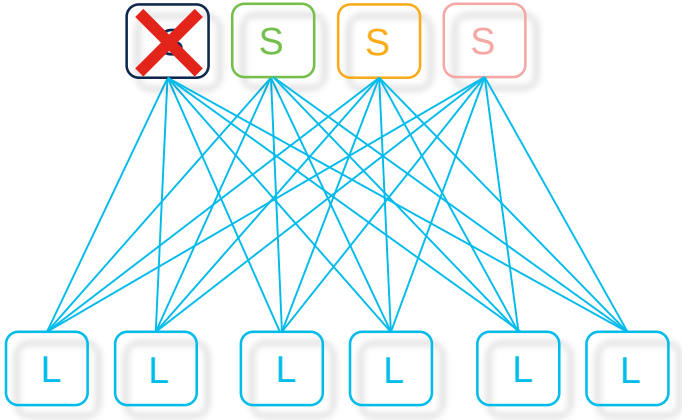
The Paradigm

N+1 Redundancy

Redundancy increases by building out the Topology

On Spine Failure

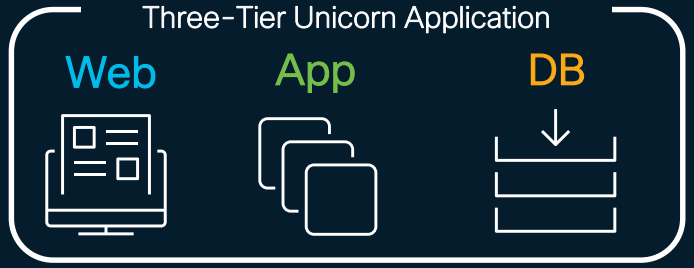
- 4 Spine = 25% impact
- 8 Spine = 12.5% impact
- 12 Spine = 8.3% impact

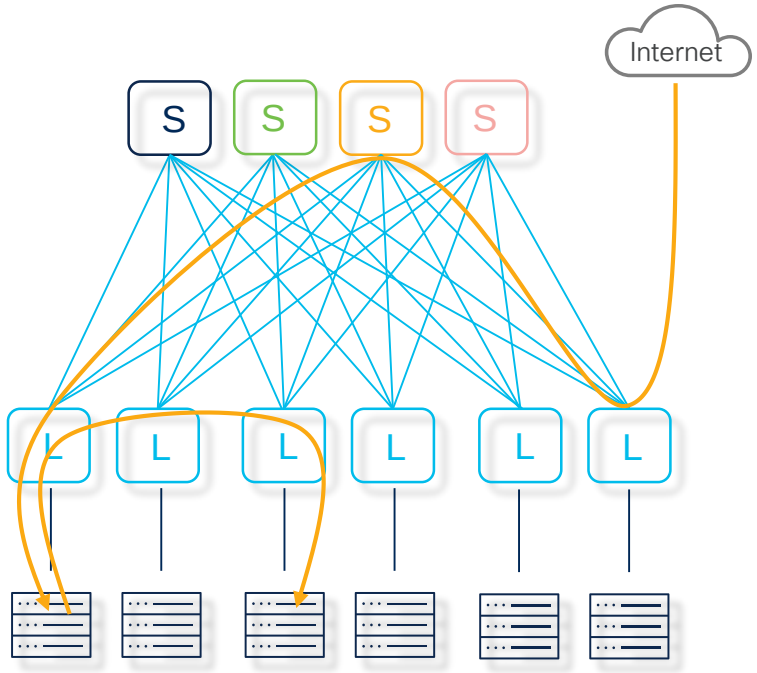


The Paradigm

Modern Applications Needs

- Every (1) North to South connection requires eight (8) East to West
- Application Frontend (User Access)
- Application Backend, DB, Storage





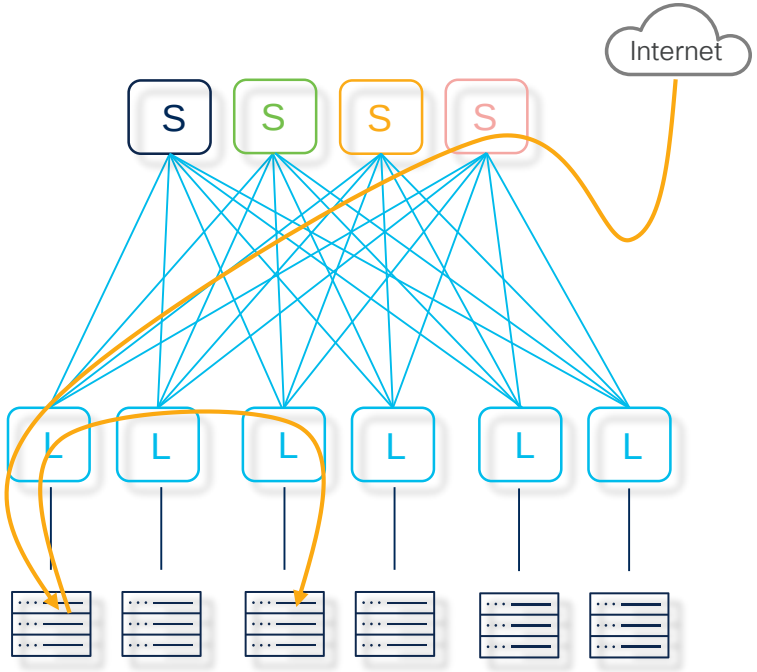
The Paradigm

Optimized for East to West

- Consistent latency from leaf to leaf
- Wide ECMP

Flexibility for North to South

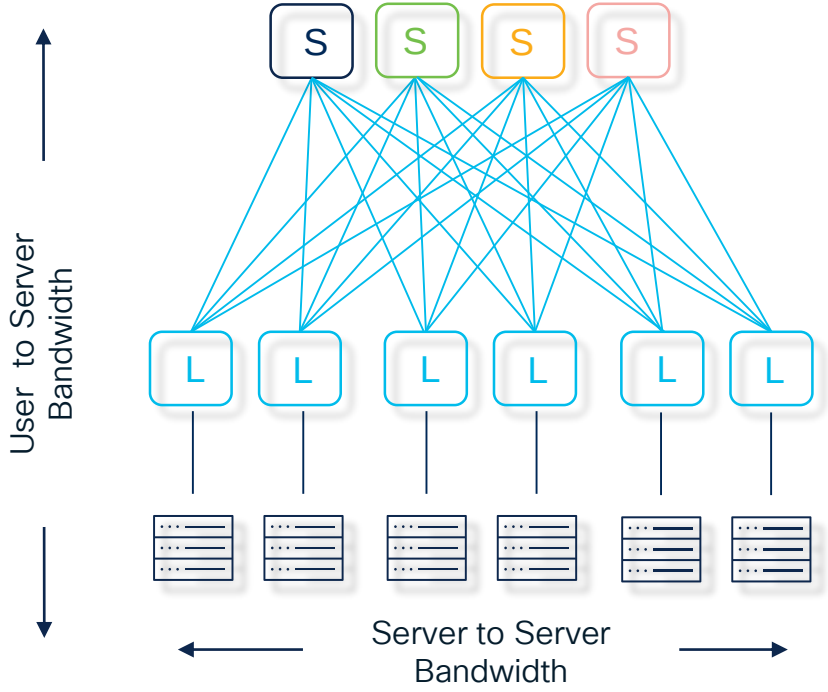
- External Connectivity at Leaf or Spine layer



The Paradigm

Bandwidth Requirements

Oversubscription



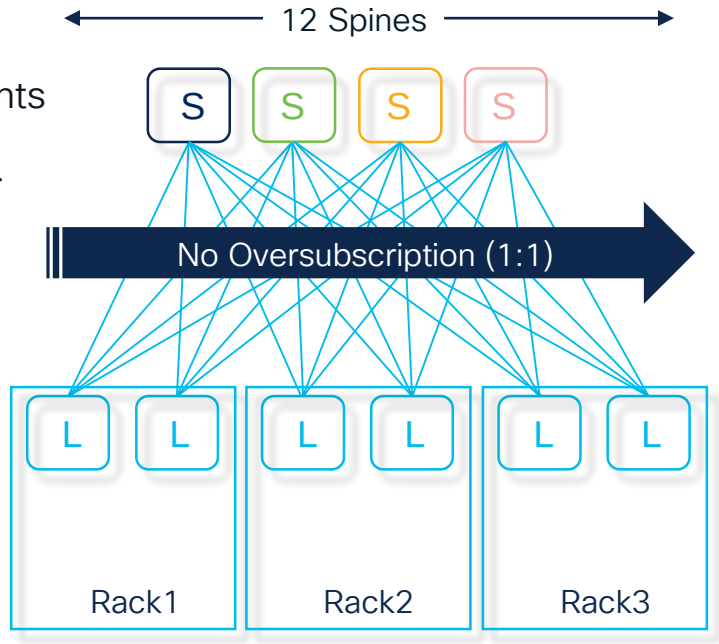
‘How Many Spines Do I Need?’

It Depends

How Many Spines do I need?

Oversubscription and Maximum Redundancy as the Criteria

Host Attachment Requirements
- 48 Server per Rack
- 2x 25Gbps NIC per Server
- 1x NIC per Switch

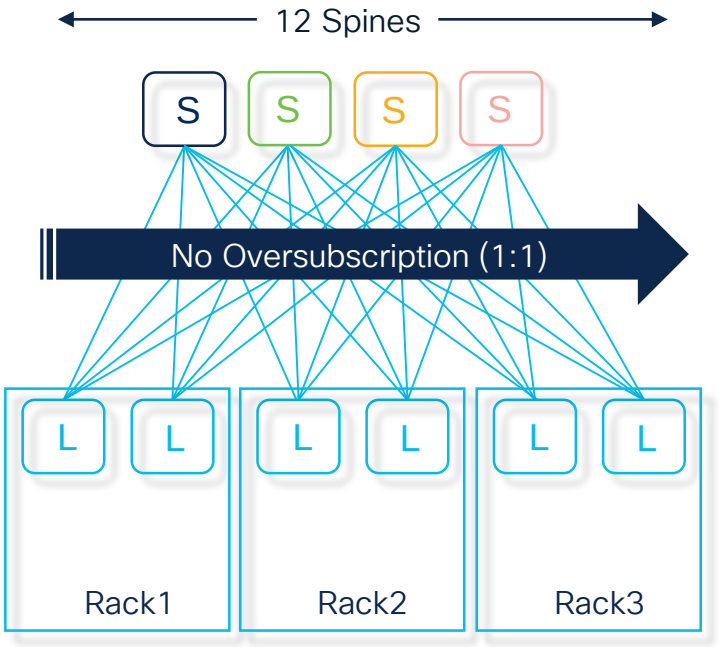


Resulting Uplink Requirements
- 48x 25Gbps per Leaf
- 1.2Tbps Uplink from Leaf to Spine
- 12x 100Gbps towards Spine

Oversubscription Ratio: [Number of Downlink Ports * Speed] ÷ [Number of Uplink to Spine * Speed]: 1

Fabric Size – 12 Spines, 1:1 Oversubscription

Oversubscription and Maximum Redundancy as the Criteria



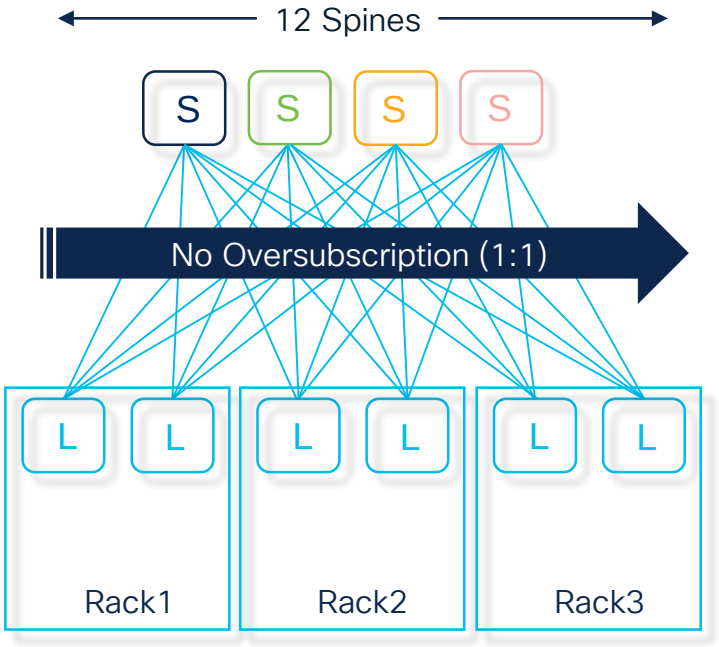
Let's Do Some Math
Spine
8 Slot Modular Chassis
36x 100Gbps Port per Linecard
Total: 288 Spine Ports
Leaf
288 Spine Ports = 288 Leaf Switch
48x 25Gbps Host Ports Per Leaf
Total: 13'824 Host Ports
Fabric Bandwidth
1:1 Oversubscription
1.2Tbps Uplink * 288 Leaf
Total: 345.6Tbps

‘What if I Replace the Chassis Size (8-slot to 16-slot chassis)?’

Is Scale Finite?

Fabric Size – 12 Spines, 1:1 Oversubscription

Oversubscription and Maximum Redundancy as the Criteria



Let's Do Some Math	
Spine	
8 Slot Modular Chassis	16 Slot Modular Chassis
36x 100Gbps Port per Linecard	36x 100Gbps Port per Linecard
Total: 288 Spine Ports	Total: 576 Spine Ports
Leaf	
288 Spine Ports = 288 Leaf Switch	576 Spine Ports = 576 Leaf Switch
48x 25Gbps Host Ports Per Leaf	48x 25Gbps Host Ports Per Leaf
Total: 13'824 Host Ports	Total: 27'648 Host Ports
Fabric Bandwidth	
1:1 Oversubscription	1:1 Oversubscription
1.2Tbps Uplink * 288 Leaf	1.2Tbps Uplink * 576 Leaf
Total: 345.6Tbps	Total: 691.2Tbps

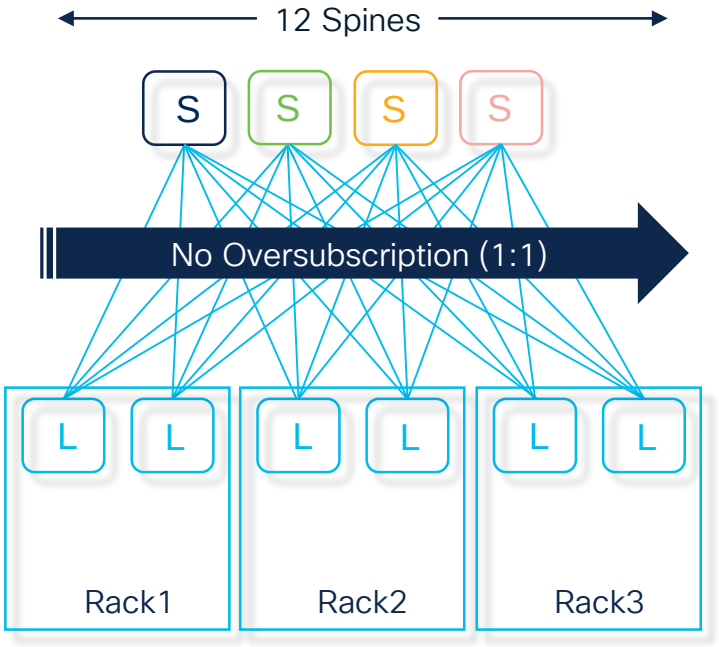
Doubling the Host Port Scale

‘What If I Replace the Spine Port Speed (100Gbps to 400bps)?’

Is Scale Finite?

Fabric Size – 12 Spines, 1:1 Oversubscription

Oversubscription and Maximum Redundancy as the Criteria



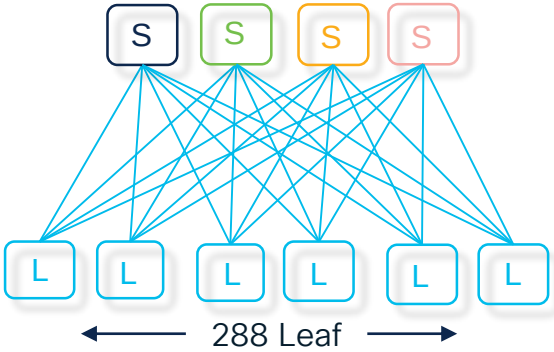
Let's Do Some Math	
Spine	
8 Slot Modular Chassis	8 Slot Modular Chassis
36x 100Gbps Port per Linecard	36x 400Gbps Port per Linecard
Total: 288 Spine Ports	Total: 1152 Spine Ports
Leaf	
288 Spine Ports = 288 Leaf Switch	1152 Spine Ports = 1152 Leaf Switch
48x 25Gbps Host Ports Per Leaf	48x 25Gbps Host Ports Per Leaf
Total: 13'824 Host Ports	Total: 55'296 Host Ports
Fabric Bandwidth	
1:1 Oversubscription	1:1 Oversubscription
1.2Tbps Uplink * 288 Leaf	1.2Tbps Uplink * 1152 Leaf
Total: 345.6Tbps	Total: 1'382.4Tbps

Quadrupling the Host Port Scale (Breakout 4x 100Gbps at Spine)

‘Scale is very Linear in 2-Tier Networks’

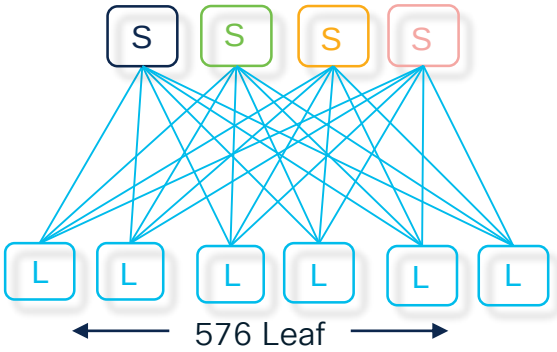
More Spines Ports Results in More -> Fabric Bandwidth, Leaf Count, Host Ports

Attributes to Scale



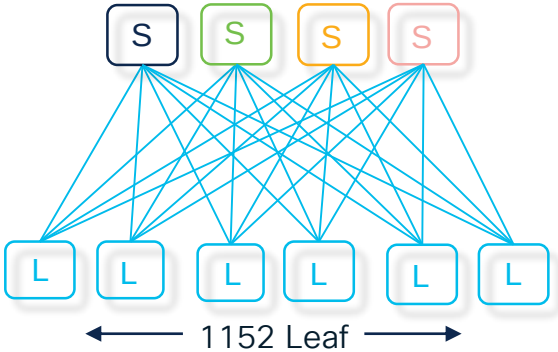
- 8 Slot Modular
- 36 x 100Gbps
- 1:1 Oversubscription
- 13’824 Host Ports

Scale Up to Fill Chassis



- 16 Slot Modular
- 36 x 100Gbps
- 1:1 Oversubscription
- 27’648 Host Ports

Scale Up to Bigger Chassis



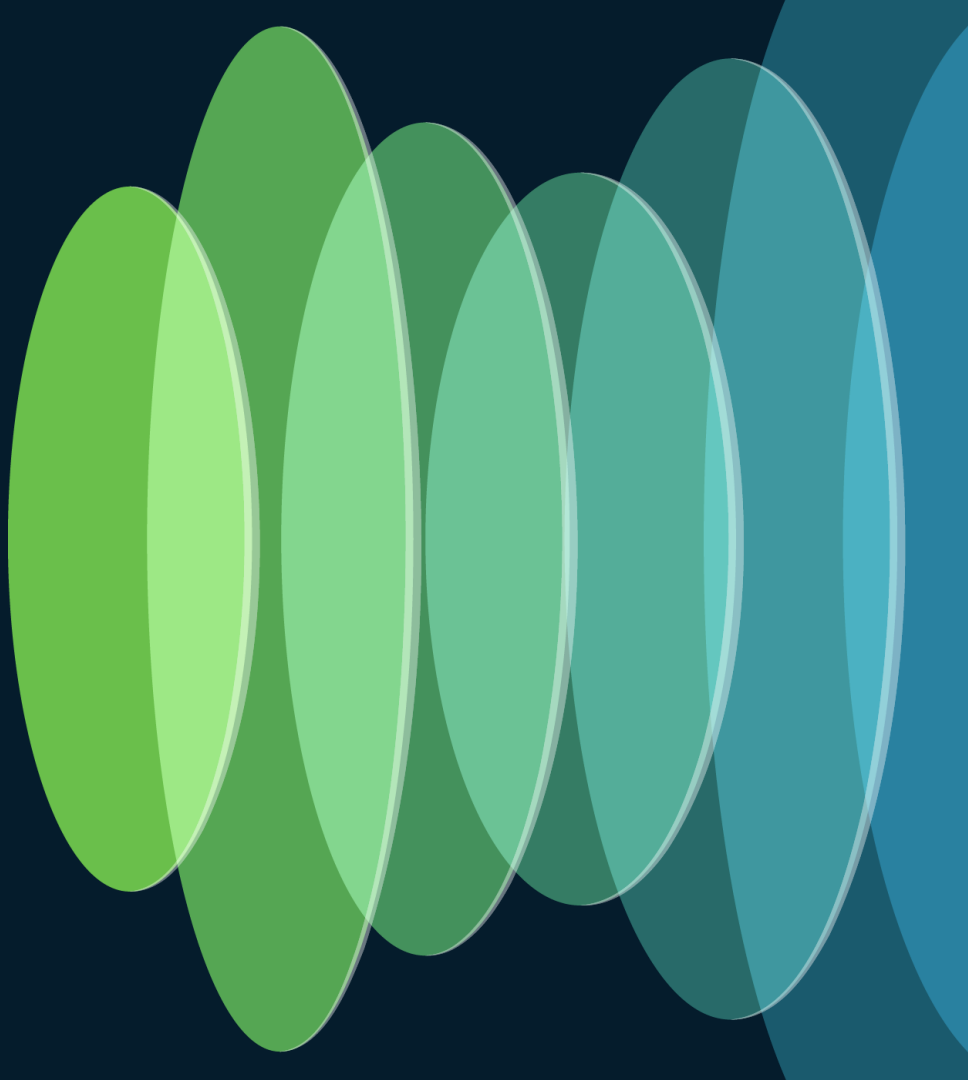
- 8 Slot Modular
- 36 x 400Gbps
- 1:1 Oversubscription
- 55’296 Host Ports

Scale Up to Faster Linecards

Oversubscription Ratio does not influence Host Port scale

Points to Think About

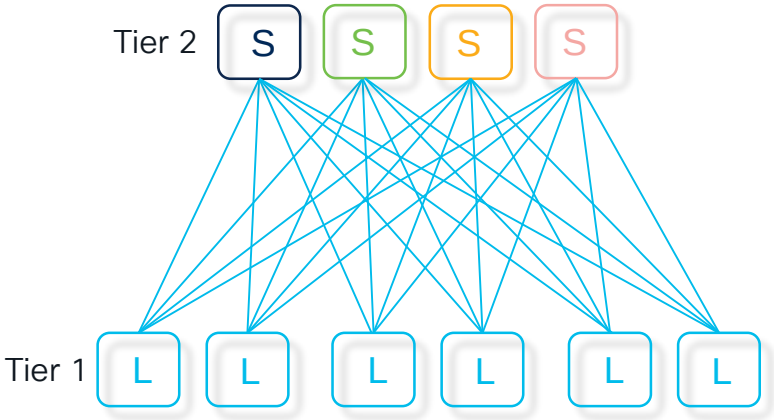
Modular Spines Considerations



cisco Live!

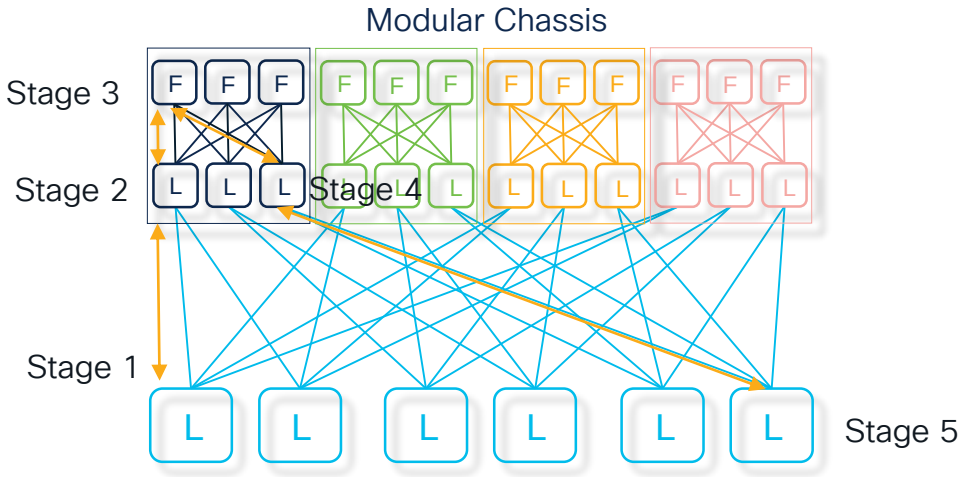
What did I really build – Untangling the Details

2 Tier / 5 Stage Network with Modular Spines



What you think you Built

2 Tier Leaf and Spine Network (3 Stage)
Spine: Modular Chassis (4, 8, 16 Slots)
Leaf: Fixed Switch (single ASIC)

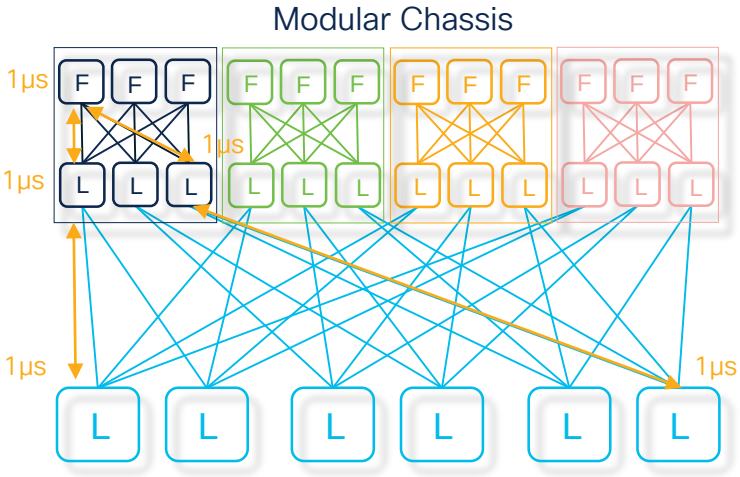


What you really Built

2 Tier Leaf and Spine Network (5 Stage)
Spine: Modular Chassis (4, 8, 16 Slots)
Leaf: Fixed Switch (single ASIC)

Latency Behavior

Latency Considerations with Modular Spines



What you really Built

2 Tier Leaf and Spine Network (5 Stage)

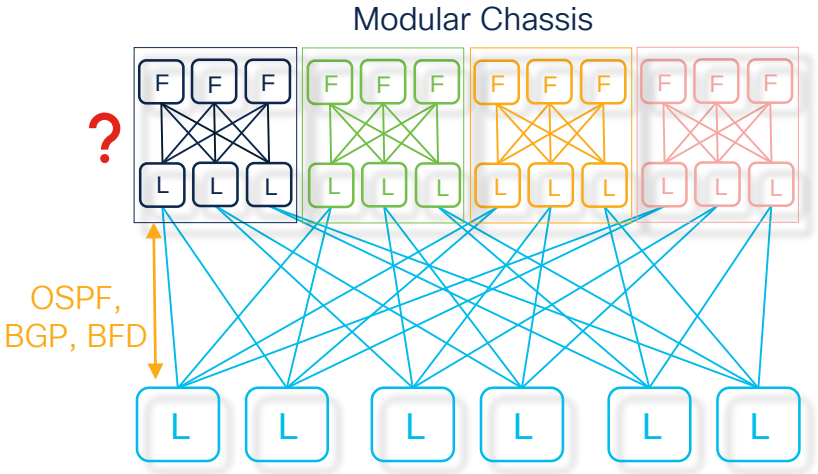
Spine: Modular Chassis (4, 8, 16 Slots)

Leaf: Fixed Switch (single ASIC)

- Generally, all Modular Switches operate in Store-and-Forward (SnF)
  - Packet Size dependent Latency
- Without Speed Change, Leaf operates in Cut-Through
  - Packet Size independent Latency
- Normalized, difference in Latency from Spine (Modular) to Leaf (Fixed) is 3:1

Intra-Chassis Behavior

Operational Considerations with Modular Spines



What you really Built

2 Tier Leaf and Spine Network (5 Stage)

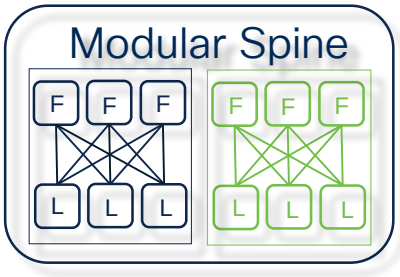
Spine: Modular Chassis (4, 8, 16 Slots)

Leaf: Fixed Switch (single ASIC)

- Within Leaf Tier and Between Leaf and Spine Tier
  - Full Behavior / Protocol Control
  - Layer-3 ECMP Load Balancing
  - Standards-based Routing Protocols
  - BFD for Fast Failure Detection
  - Minimal Exposure for Brownout
- Within Spine Tier
  - Intra-Chassis Load Balancing
  - Intra-Chassis Protocol
  - Intra-Chassis Failure Detection
  - Fully Redundant Components

Modular or Fixed Spines?

Summary

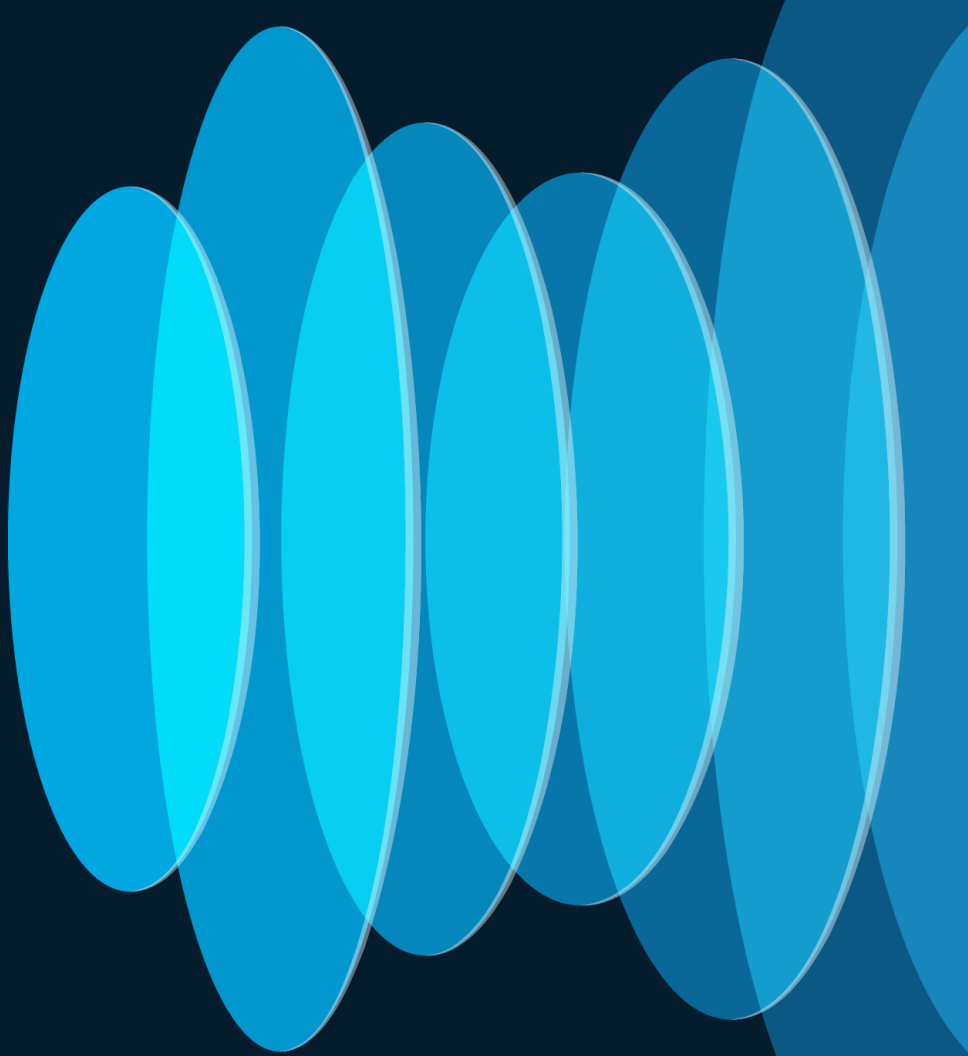


- OPEX savings (less devices to manage)
- Higher port density
- Limited visibility into intra-chassis failures
- Works only in store-and-forward mode

- CAPEX and power/cooling savings
- Deterministic latency (single-stage devices)
- Scale with breakouts
- More devices to manage

Design Evolution

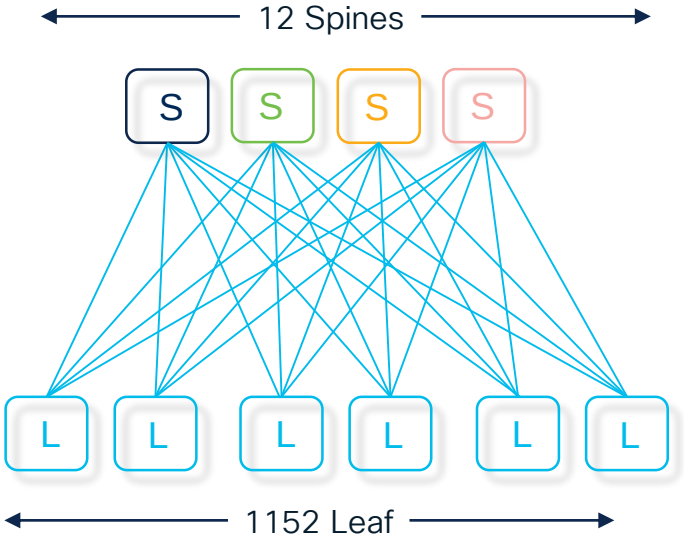
Building Distributed Architectures



cisco Live!

Design Evolution

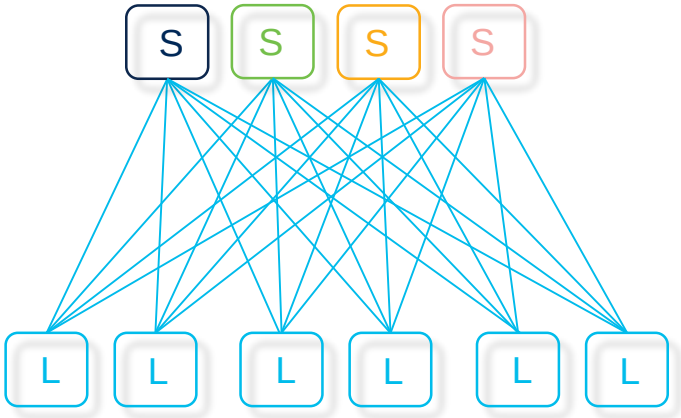
Various Considerations to Reduce the Fabric's Size



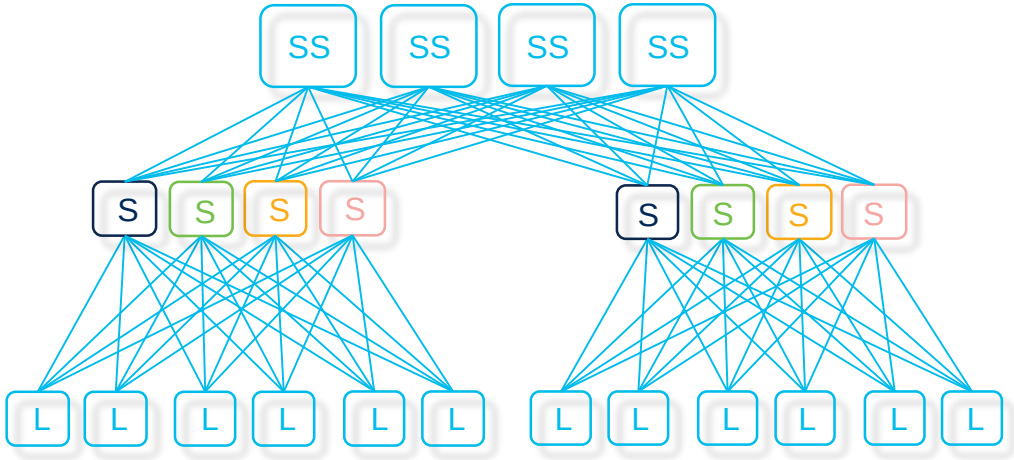
- What is my Failure Domain?
- What is my Change Domain?
- What is my Overall Scale?
- What is my Fabric Solution Scale?
- What is my Fabric SLA?
- What is my Maximum Downtime?

Design Evolution

From a Single Large Fabric to a Distributed Architecture



2-Tier Leaf Spine
(~5 Stages)

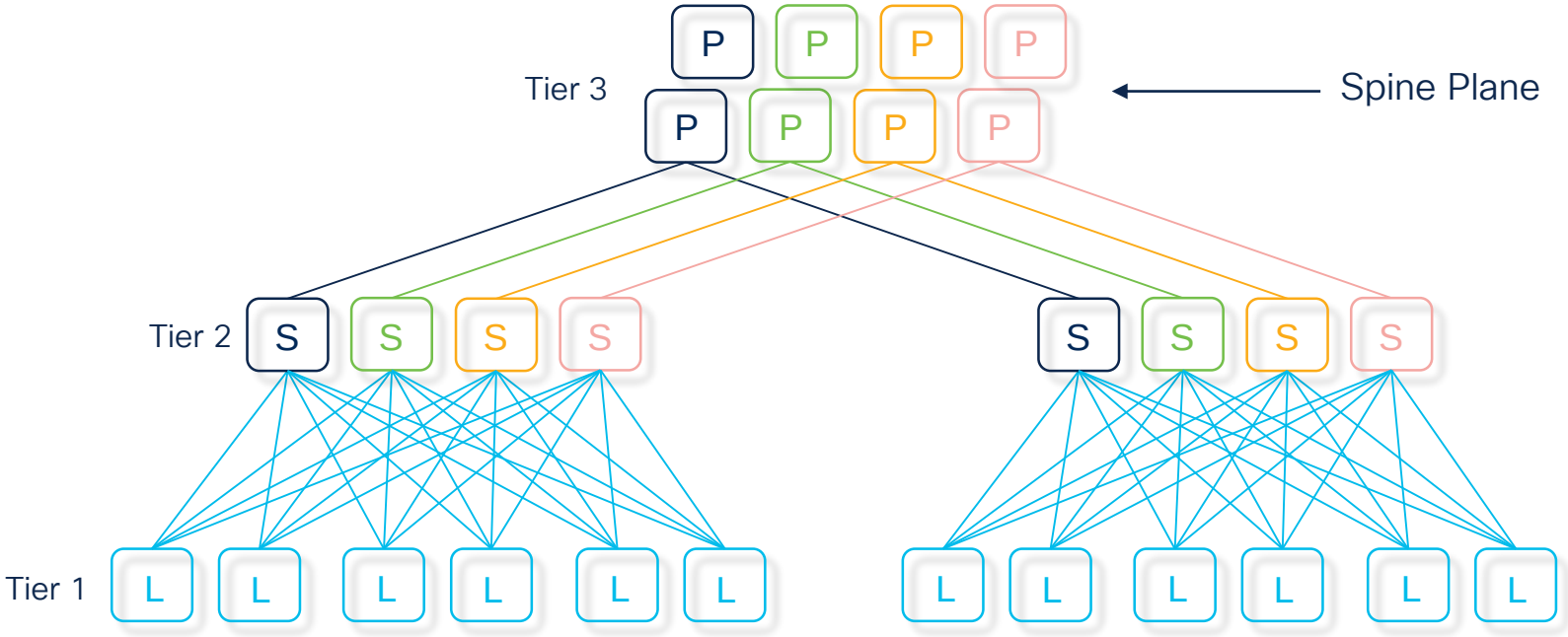


3-Tier Leaf Spine with Super Spine
(~11 Stages)

Tend to have a “Finite Scale”

Design Evolution

From a Single Large Fabric to a Distributed Architecture



3-Tier Leaf Spine with Plane Design
(5 Stages)

What we learned from the Cloud Titans

Building Scalable Data Center Networks

#1

Simplicity is Key

Simple Design Principals

#2

Scale as you Go

Scale is Never Finite

#3

Fail but Fail Fast

Reduce Brown-Out Exposure

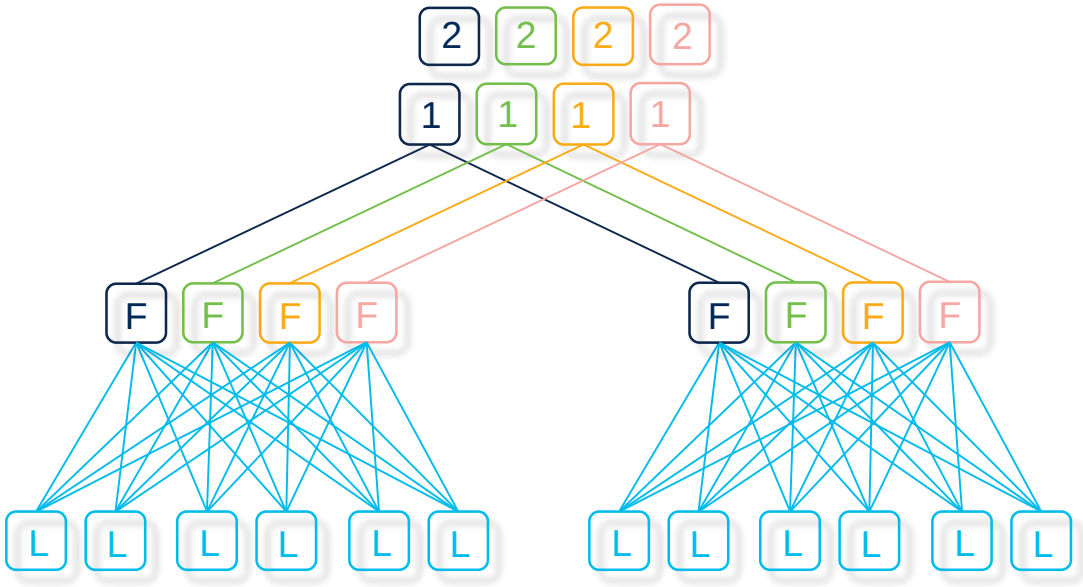
#4

Redundant and Repeatable

Risk is Never an Option

How the Cloud Titans build

- Increasing Scale-Out in all Tiers
- Reducing Complexity
  - Simple Design Principles
- Increasing the “Finite Scale”
  - Scale as You Go
- Disaggregated Redundancy
- Flexible Link and Bandwidth Distribution
- Further Possibility for Cost Optimization



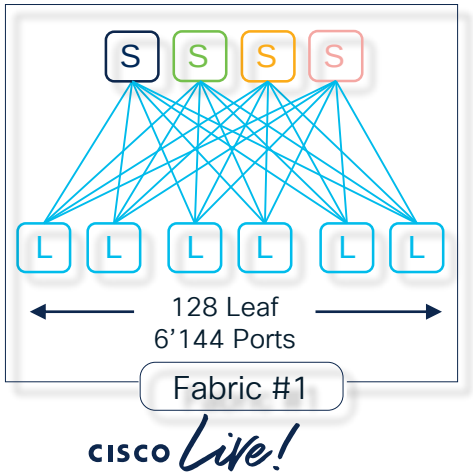
Hyperscale Fabric Plane Clos Design

Cisco’s Massively Scalable Data Center Network Fabric White Paper

Step 1 – Don’t Build Fabric For Maximum Leaf

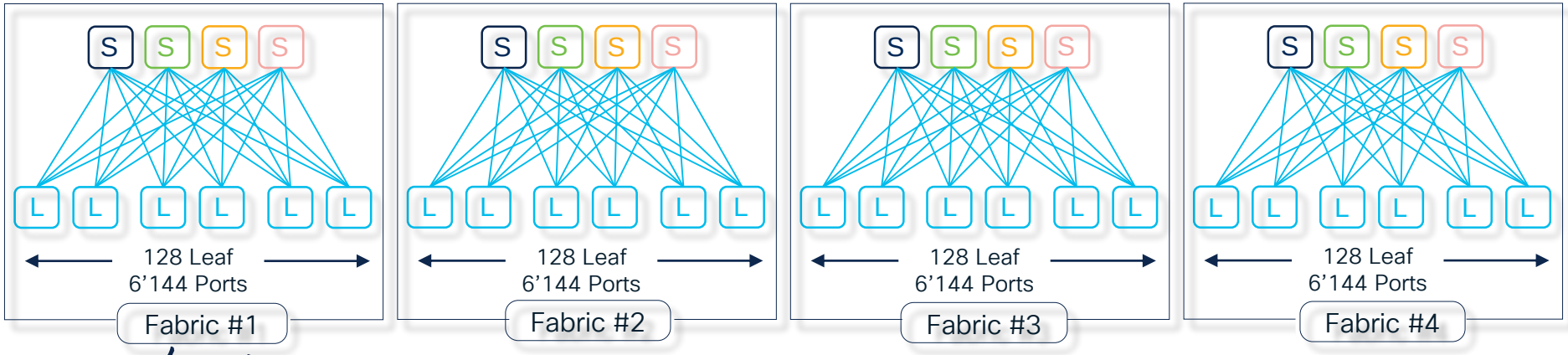
- Fixed Switch at the Spine (Tier 2)
  - Depending on the Oversubscription ratio, reserve Ports
    - 1:1 Oversubscription
    - Reserve 50% from Tier 2 to Tier 3

<u>Tier 2: Nexus 9364D-GX2B – 64x Ports 400Gbps</u>
50% Uplink to Tier 3 (32x 400Gbps)
50% Downlink for Leaf (128x 100Gbps)
Breakout: 32x 400Gbps = 128x 100Gbps



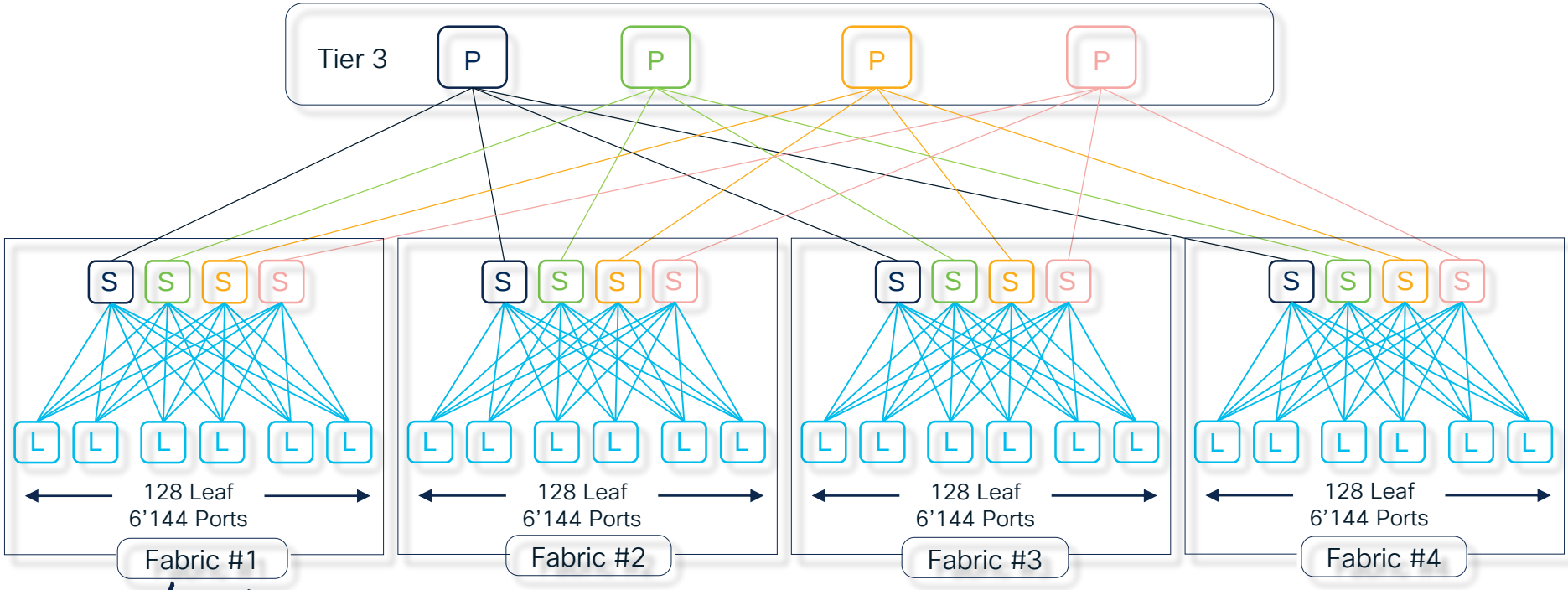
Step 2 – Repeat for Host Port Scale (Scale Out)

- Adding new Fabrics at need
  - Of Host Port
  - Of Oversubscription between Tier 2 and Tier 3
- Result Defines Tier 2 to Tier 3 Uplinks
  - And respectively Tier 3 Requirements



Step 3 – Design the Tier 3

- Introducing Tier 3 Planes
  - Blue, Green, Orange, Red
- **Rule 1:** Tier 2 Blue only connects to Tier 3 Blue
- **Rule 2:** Once entered a Plane, you stay in the Plane



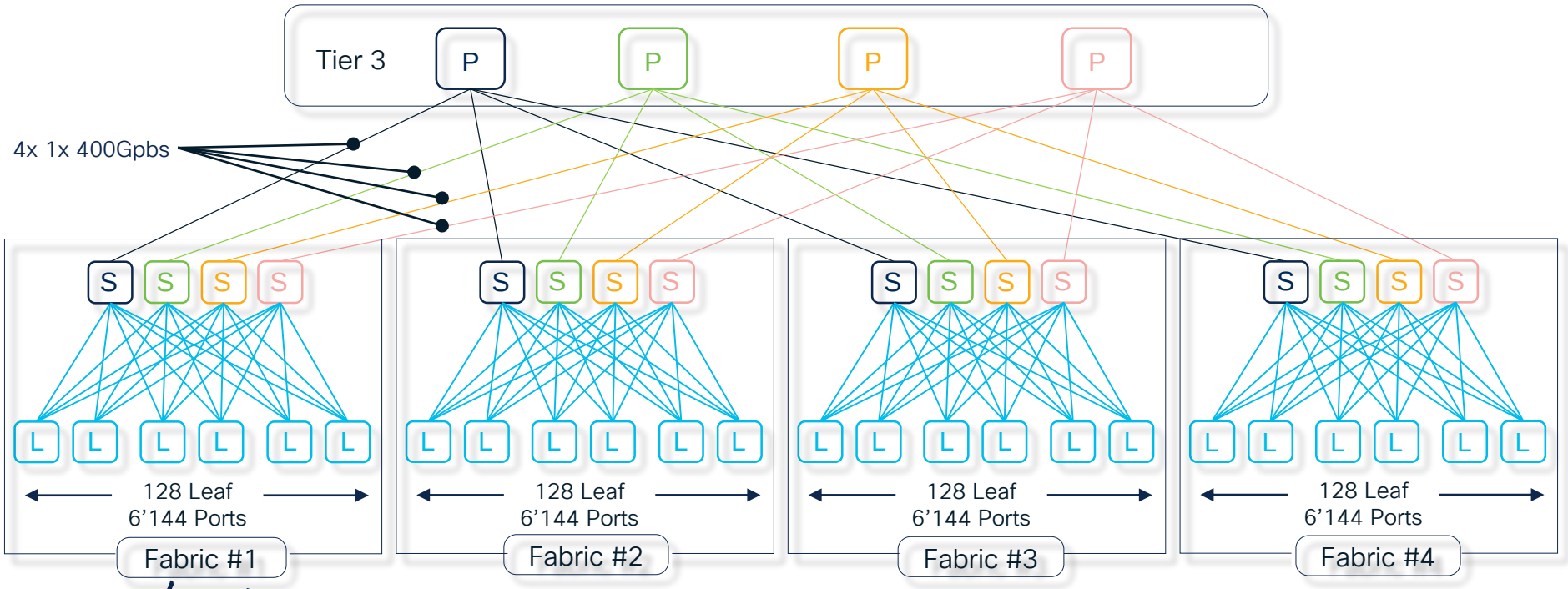
Step 3 – Design the Tier 3

Single Link

Up to 32 Fabrics
32x 6’144 = 196’608 Ports

Tier 3: Nexus 9332D-GX2B (4 per Plane) – 32x Ports 400Gbps

4x 1x 400Gbps = 1.6Tbps Inter-Fabric Bandwidth



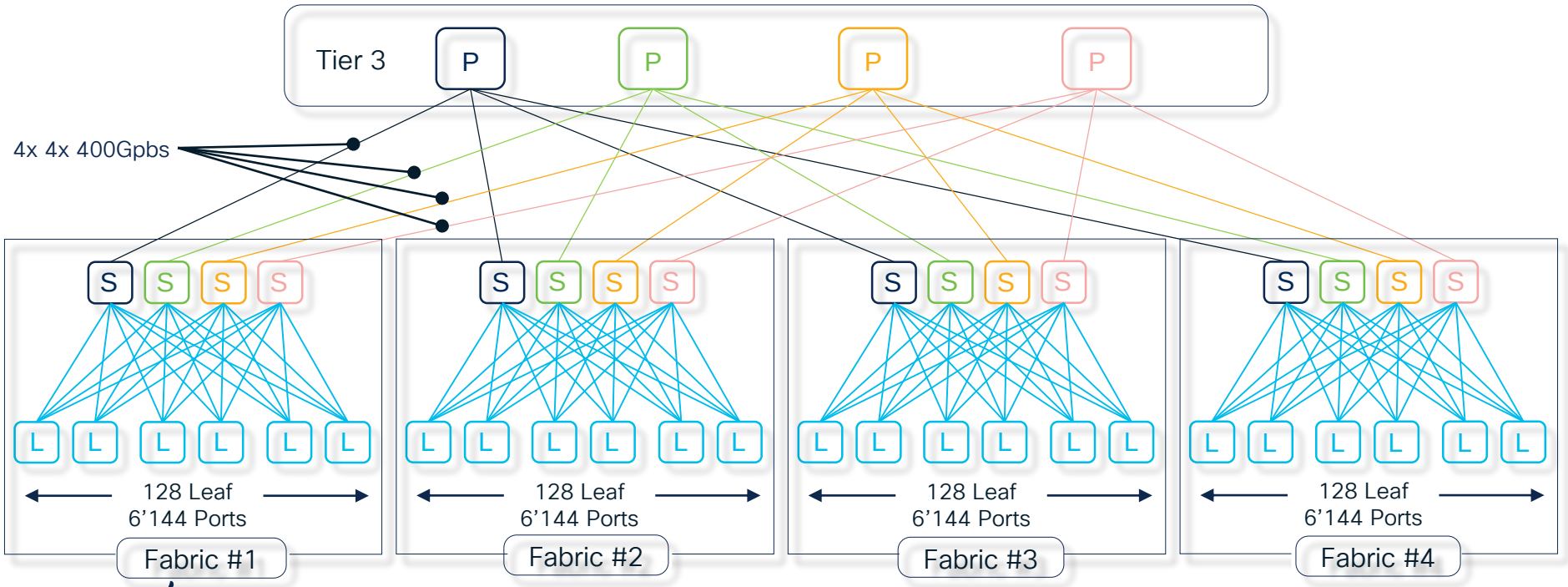
Step 3 – Design the Tier 3

Multi Link

Up to 8 Fabrics
8x 6’144 = 49’152 Port

Tier 3: Nexus 9332D-GX2B (4 per Plane) – 32x Ports 400Gbps

4x 4x 400Gbps = 6.4Tbps Inter-Fabric Bandwidth

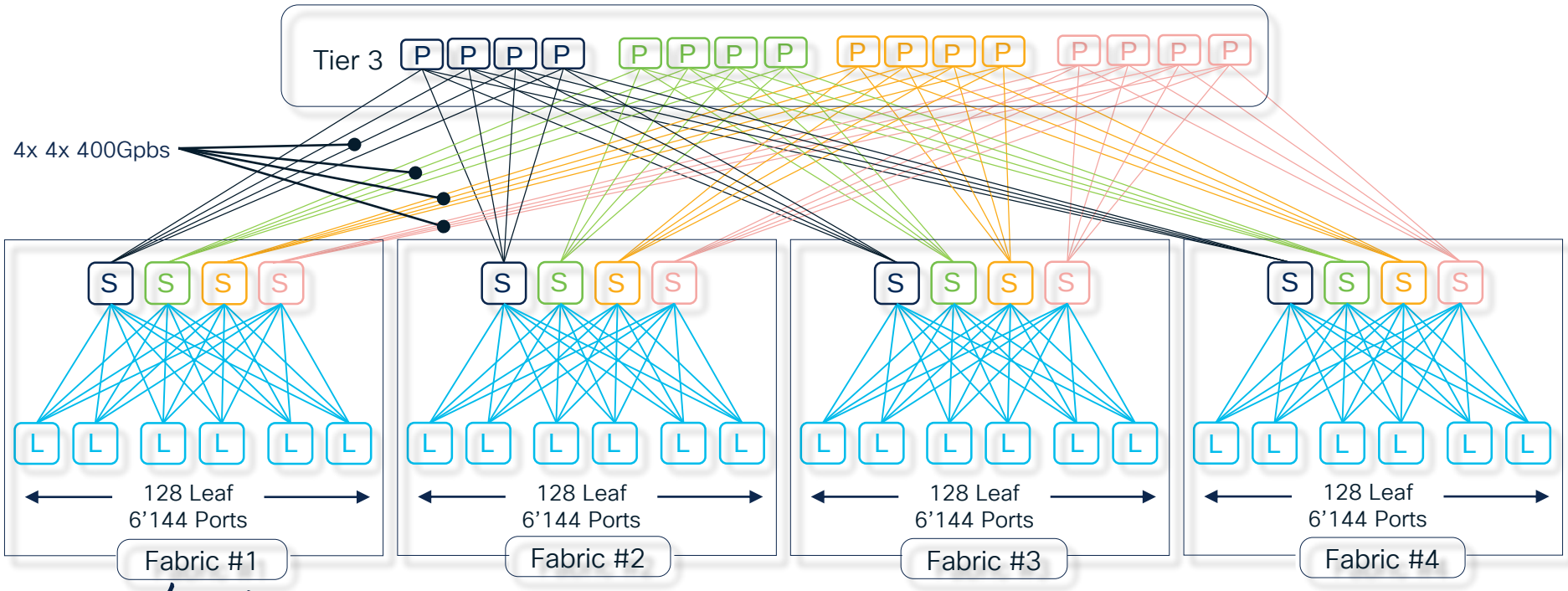


Step 4 – Scaling the Tier 3 Planes

Up to 32 Fabrics
32x 6’144 = 196’608 Ports

Tier 3: Nexus 9332D-GX2B (16 per Plane) – 32x Ports 400Gbps

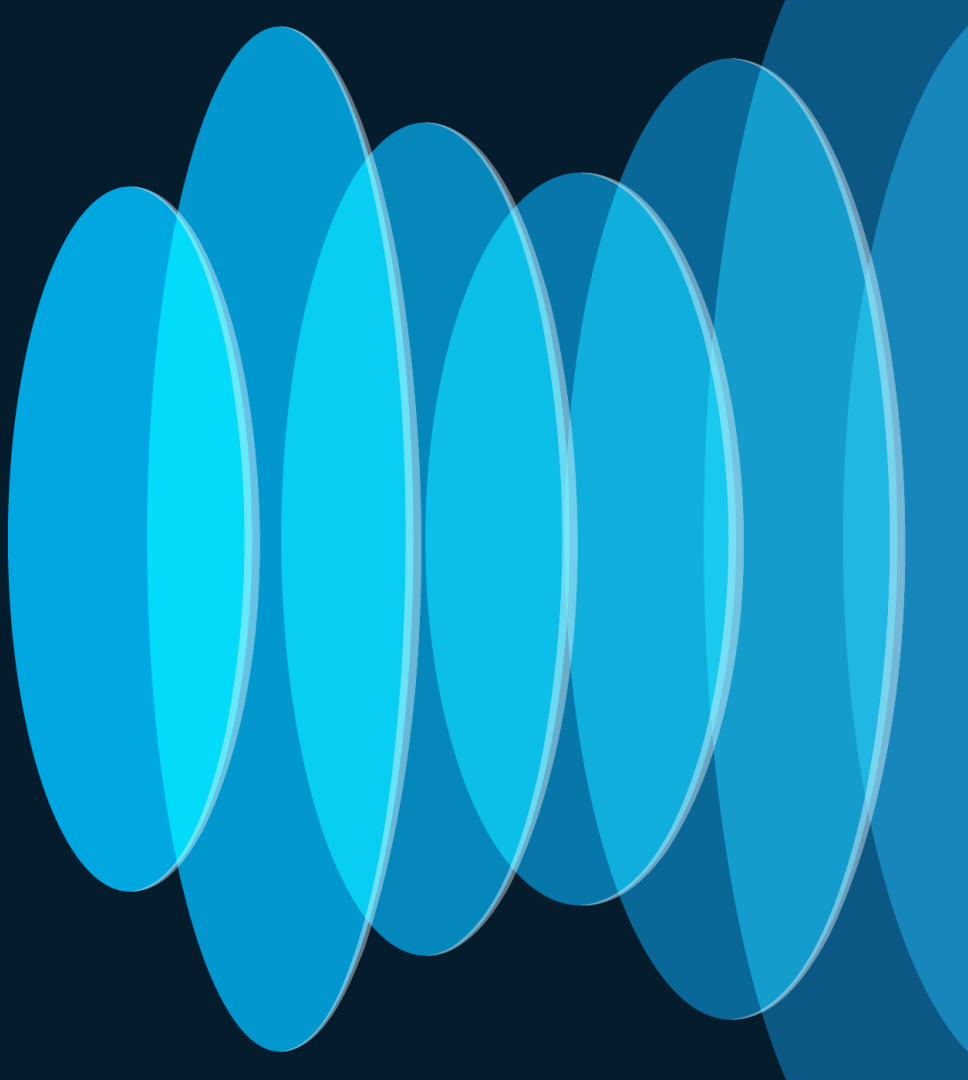
4x 4x 400Gbps = 6.4Tbps Inter-Fabric Bandwidth



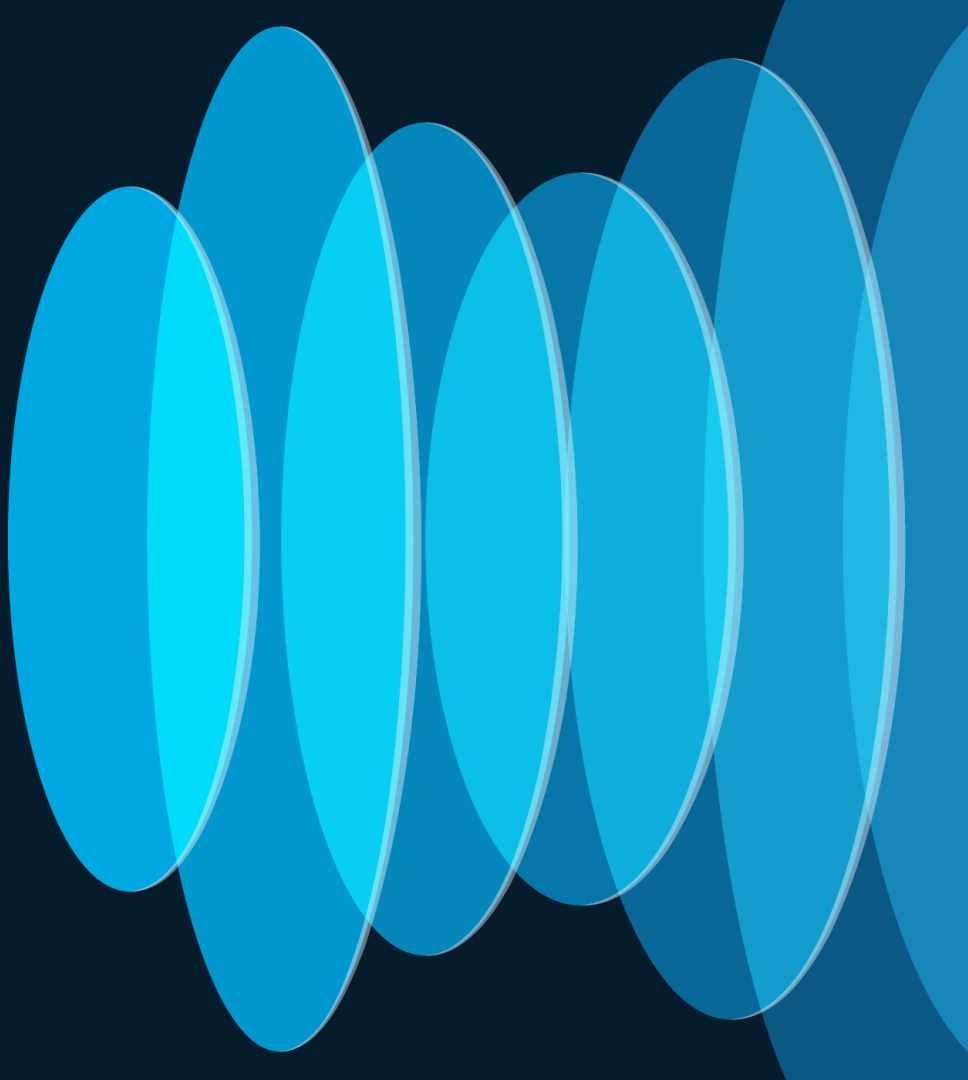
Design Evolution

Mapping to Different Architectural Options

- ACI Fabrics
- VXLAN EVPN Fabrics
- Heterogeneous Fabrics



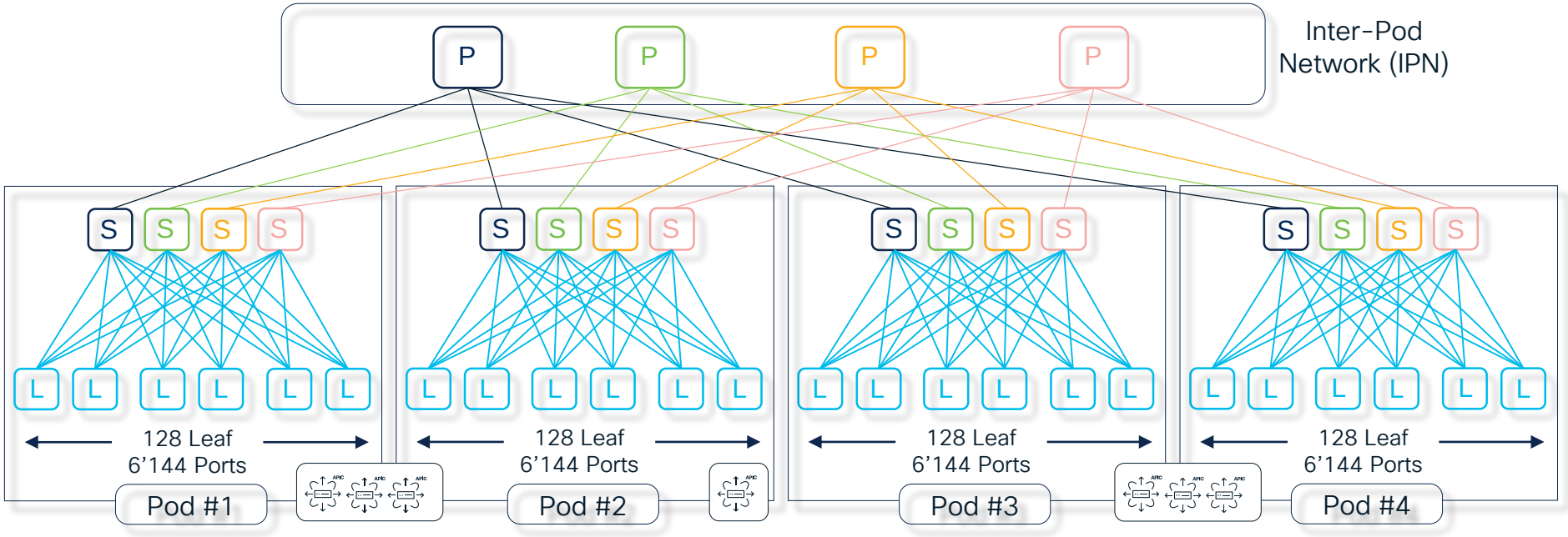
ACI Fabrics



ACI Multi-Pod

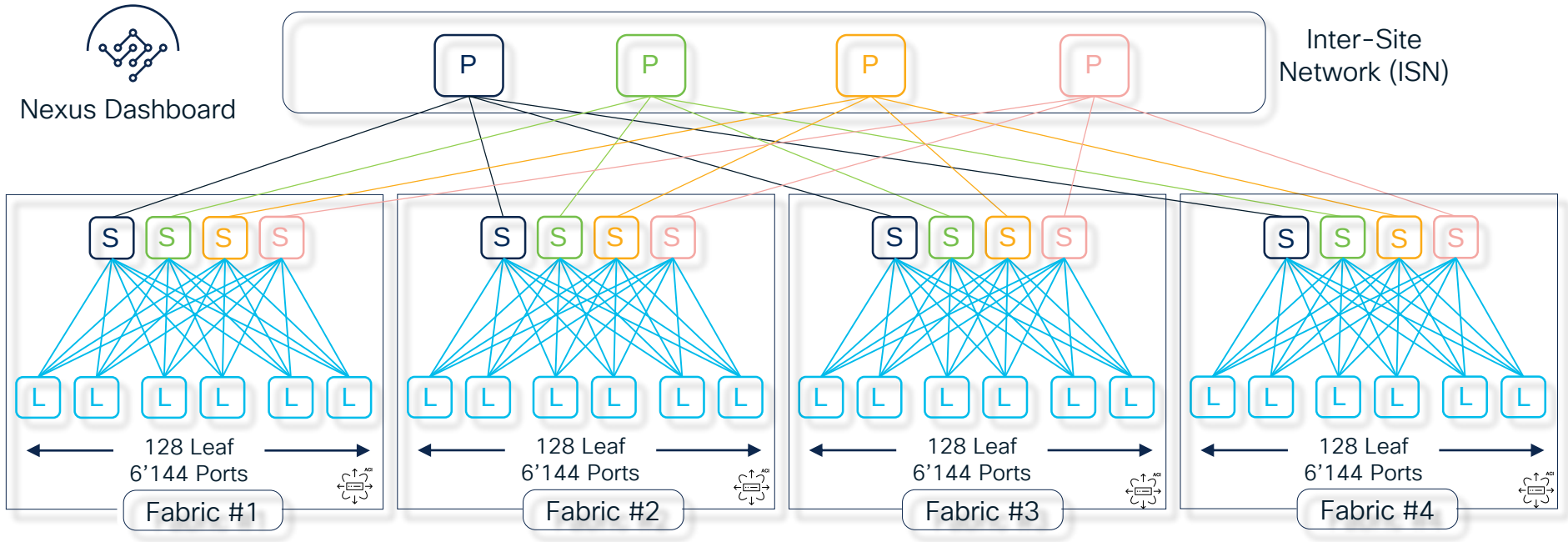
Up to 25 Pods in the Same Fabric

ACI Multi-Pod
BRKDCN-2949



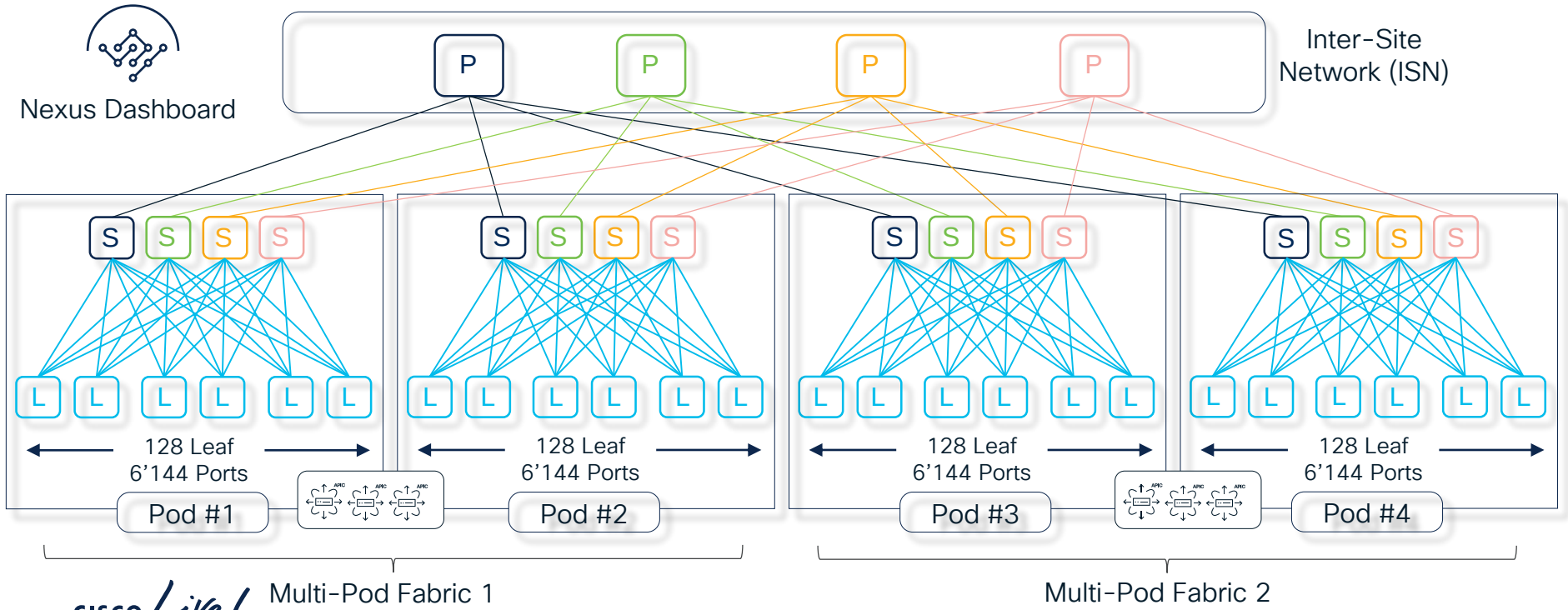
ACI Multi-Site

Up to 14 Fabrics in the same Multi-Site Domain

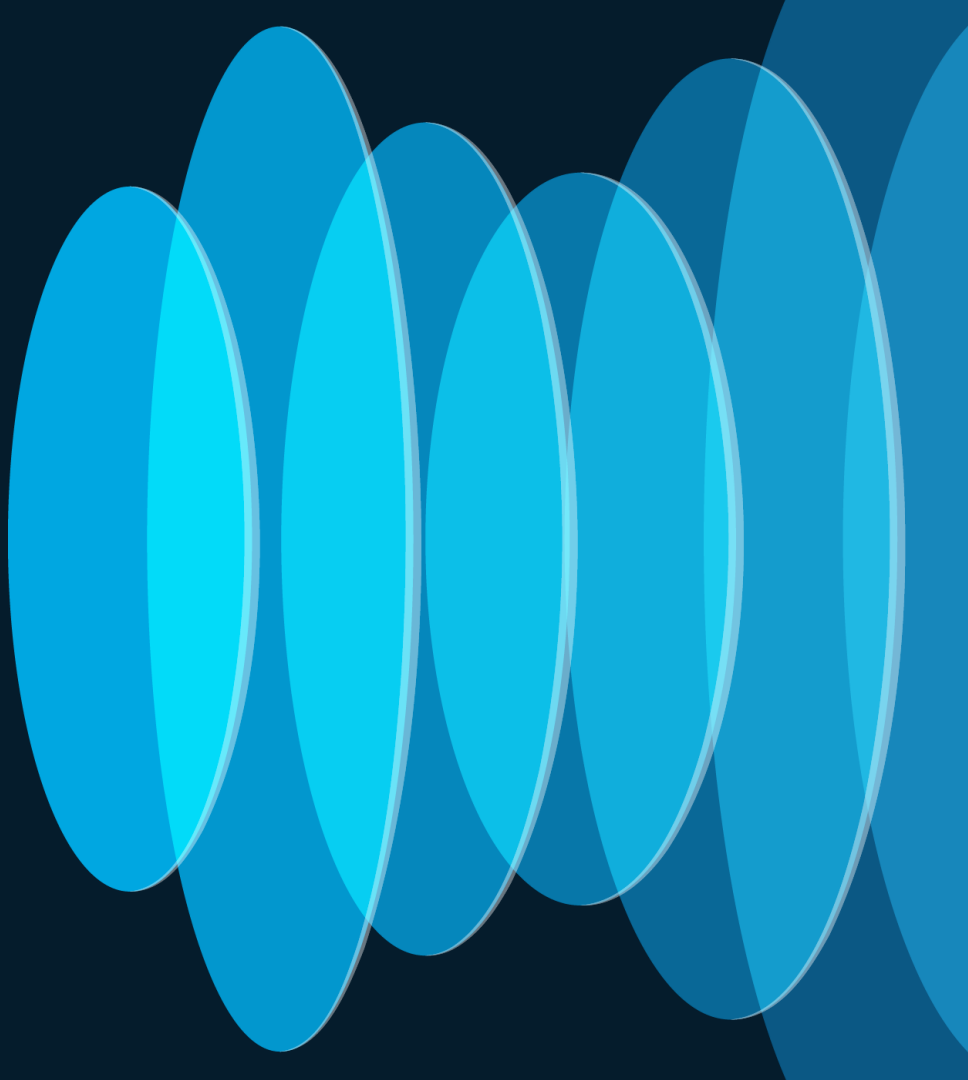


ACI Multi-Pod + Multi-Site

ACI Multi-Site with Single or Multi-Pod
BRKDCN-2919



VXLAN EVPN Fabrics

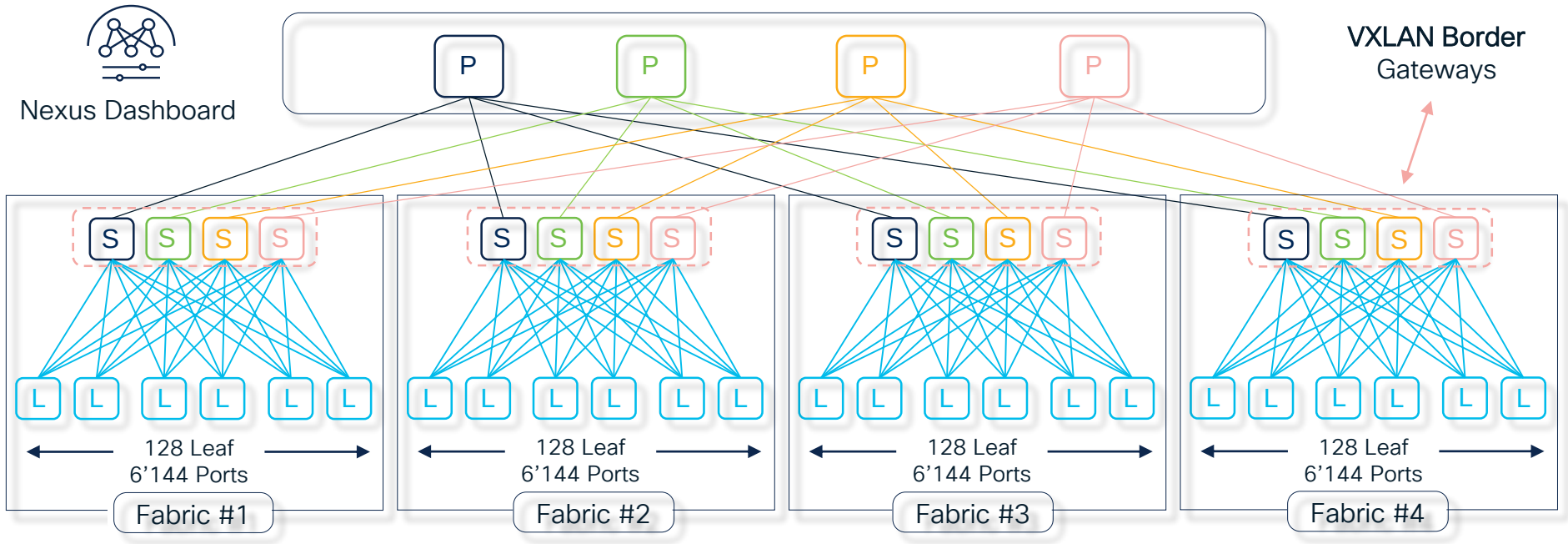


cisco Live!

VXLAN EVPN Fabrics

Multi-Site, Single Domain

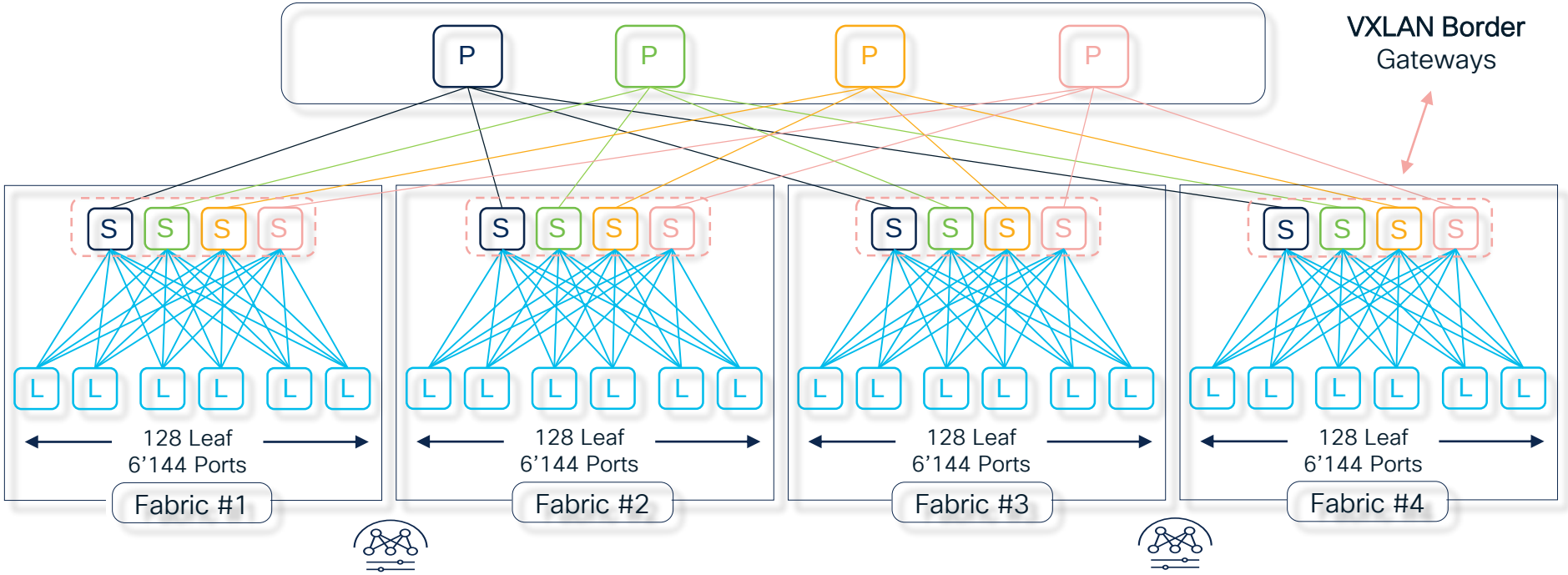
VXLAN EVPN
Multi-Site
BRKDCN-2913



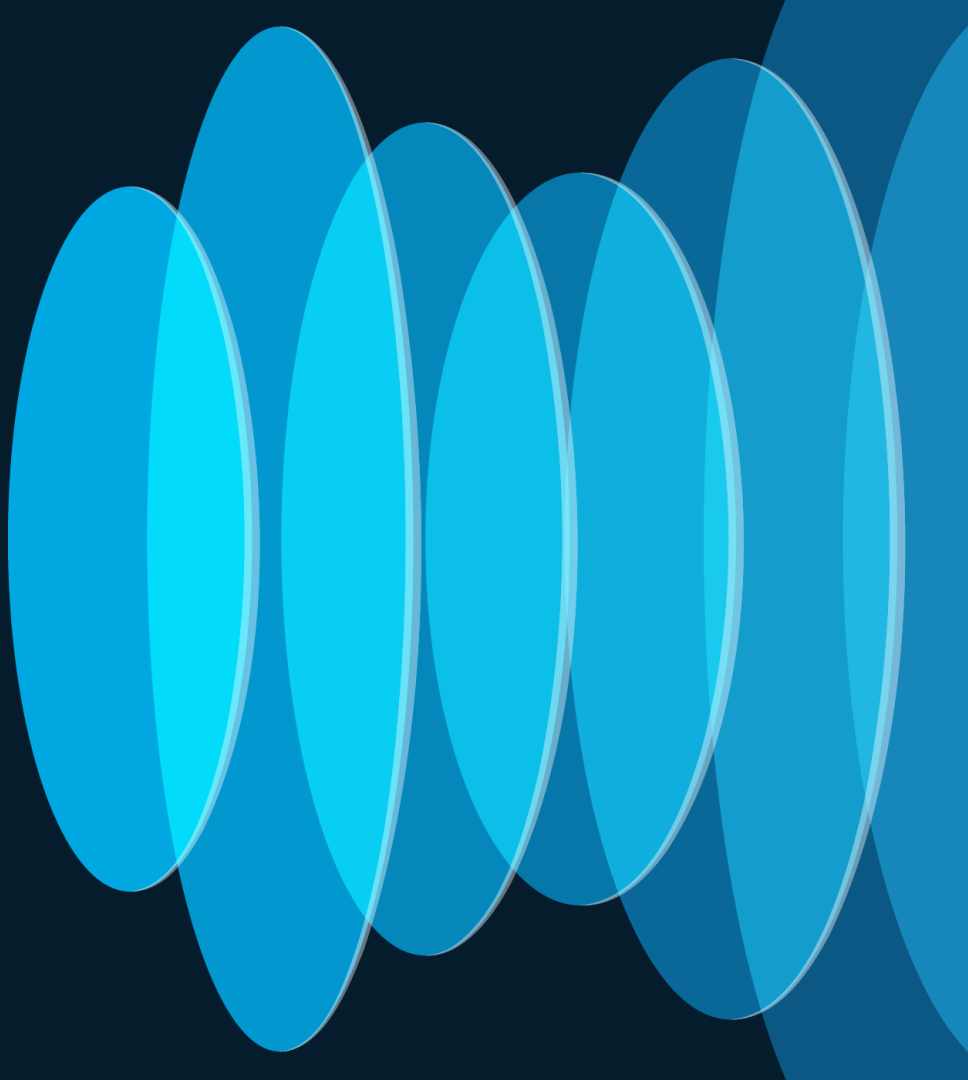
VXLAN EVPN Fabrics

Multi-Site, Multiple Domains

VXLAN EVPN
Multi-Site with NDFC
BRKDCN-2988



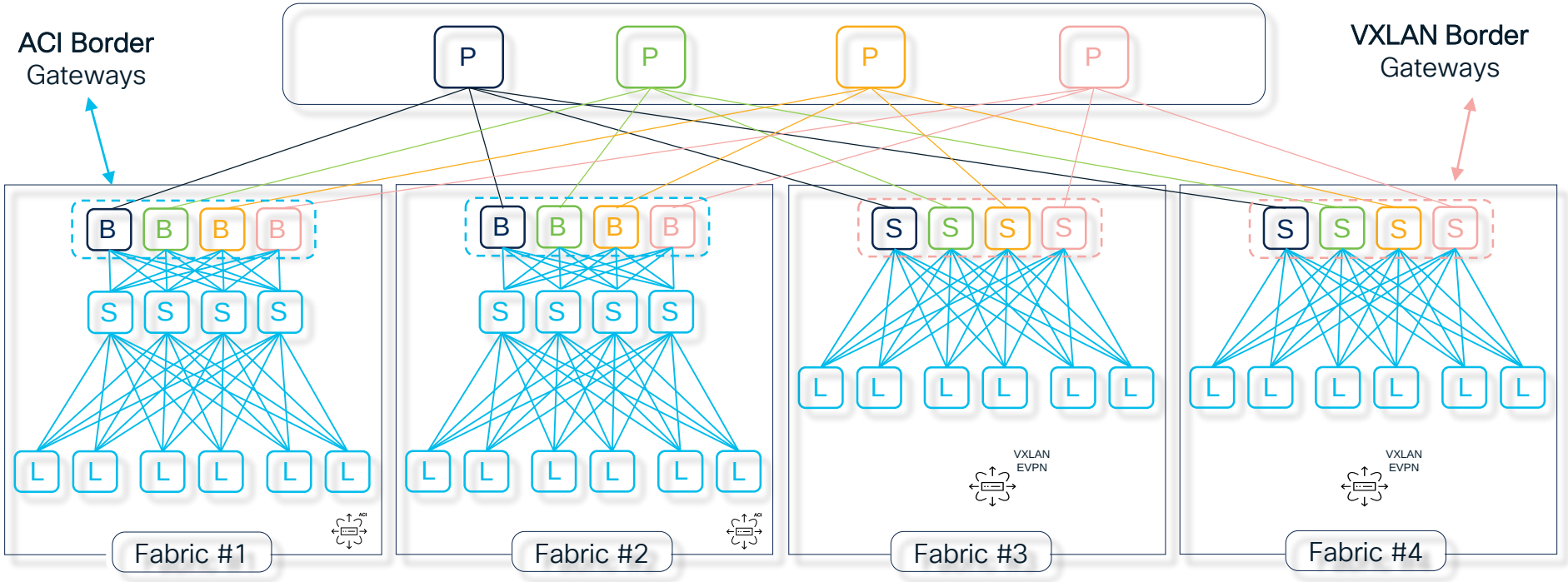
Heterogeneous Fabrics



New Session!
BRKDCN-2934

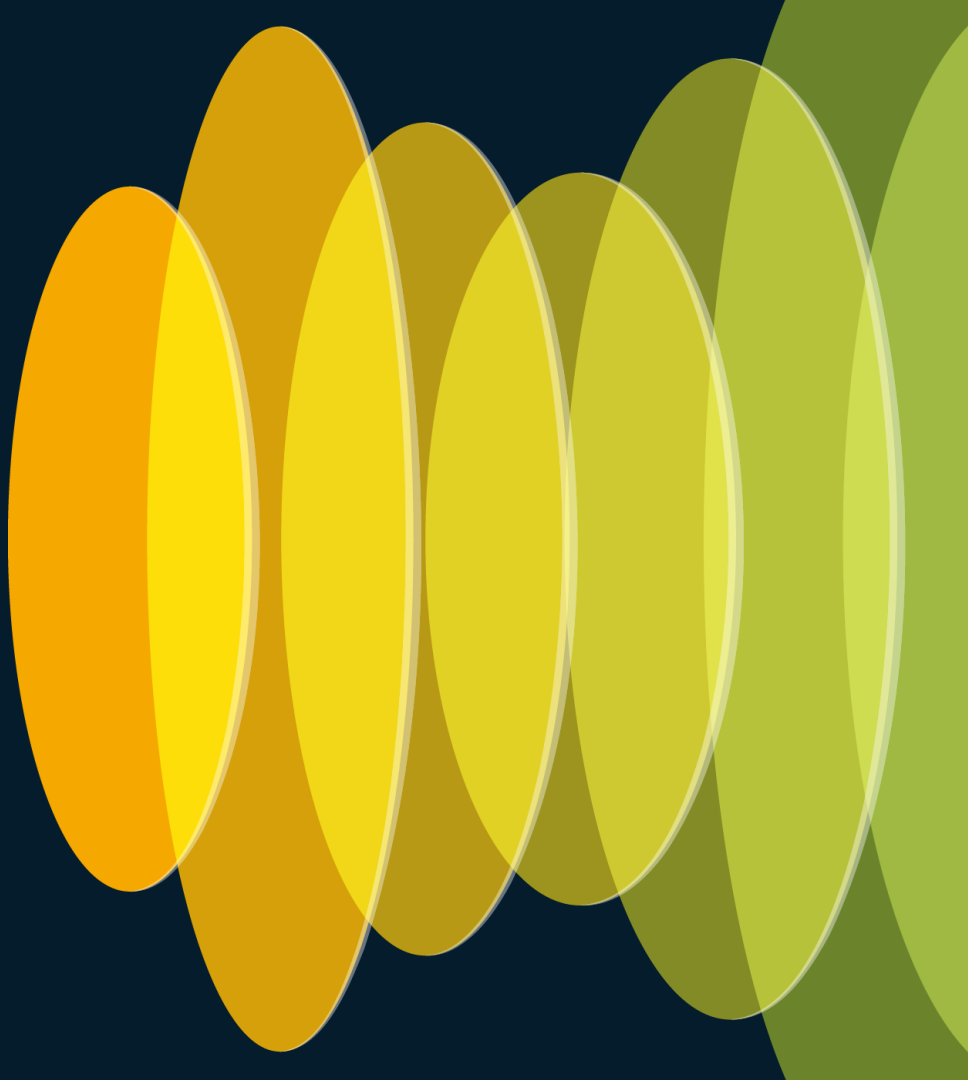
Heterogeneous Fabrics

“Opening Up” L2/L3 Connectivity between ACI and VXLAN EVPN Fabrics



Routed Fabrics

RFC 5549 and RFC 7938



‘Advertising IPv4 Network Layer Reachability Information with an IPv6 Next Hop’

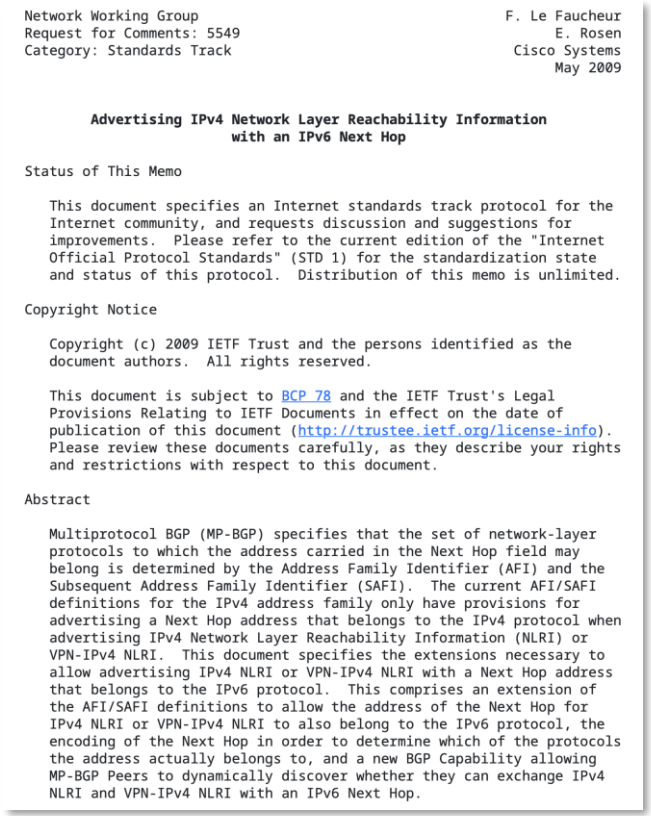
RFC 5549

What is RFC 5549?

By the Standards Body

- Categorized for Standards Track
- Internet Standard since 2009
- Updated by RFC 8950
  - aka RFC 5549bis
- Industry-wide adoption for 10+ years
- Invented and Authored by Cisco

RFC 5549: https://datatracker.ietf.org/doc/html/rfc5549
RFC 8950: https://datatracker.ietf.org/doc/html/rfc8950



Network Working Group
Request for Comments: 5549
Category: Standards Track

F. Le Faucheur
E. Rosen
Cisco Systems
May 2009

Advertising IPv4 Network Layer Reachability Information with an IPv6 Next Hop

Status of This Memo

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

Copyright Notice

Copyright (c) 2009 IETF Trust and the persons identified as the document authors. All rights reserved.

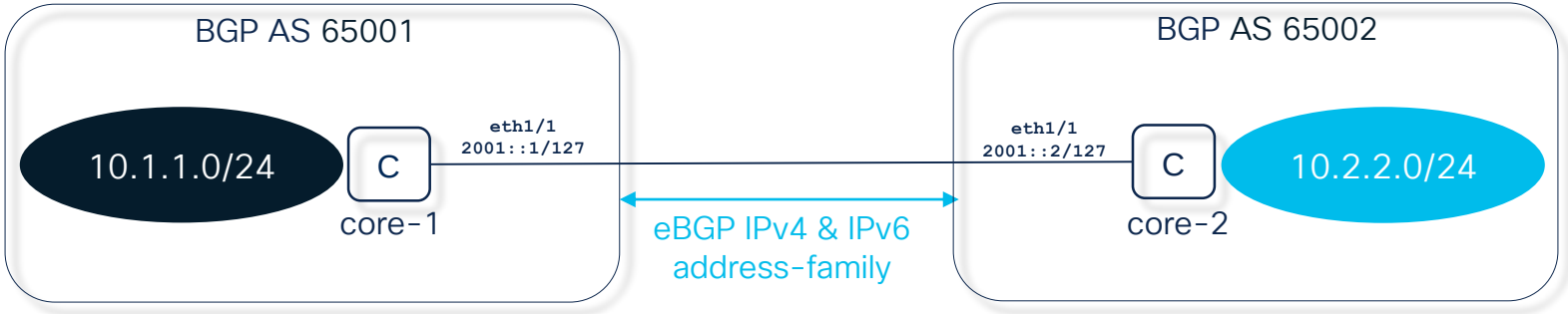
This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents in effect on the date of publication of this document (http://trustee.ietf.org/license-info). Please review these documents carefully, as they describe your rights and restrictions with respect to this document.

Abstract

Multiprotocol BGP (MP-BGP) specifies that the set of network-layer protocols to which the address carried in the Next Hop field may belong is determined by the Address Family Identifier (AFI) and the Subsequent Address Family Identifier (SAFI). The current AFI/SAFI definitions for the IPv4 address family only have provisions for advertising a Next Hop address that belongs to the IPv4 protocol when advertising IPv4 Network Layer Reachability Information (NLRI) or VPN-IPv4 NLRI. This document specifies the extensions necessary to allow advertising IPv4 NLRI or VPN-IPv4 NLRI with a Next Hop address that belongs to the IPv6 protocol. This comprises an extension of the AFI/SAFI definitions to allow the address of the Next Hop for IPv4 NLRI or VPN-IPv4 NLRI to also belong to the IPv6 protocol, the encoding of the Next Hop in order to determine which of the protocols the address actually belongs to, and a new BGP Capability allowing MP-BGP Peers to dynamically discover whether they can exchange IPv4 NLRI and VPN-IPv4 NLRI with an IPv6 Next Hop.

What is RFC 5549 for?

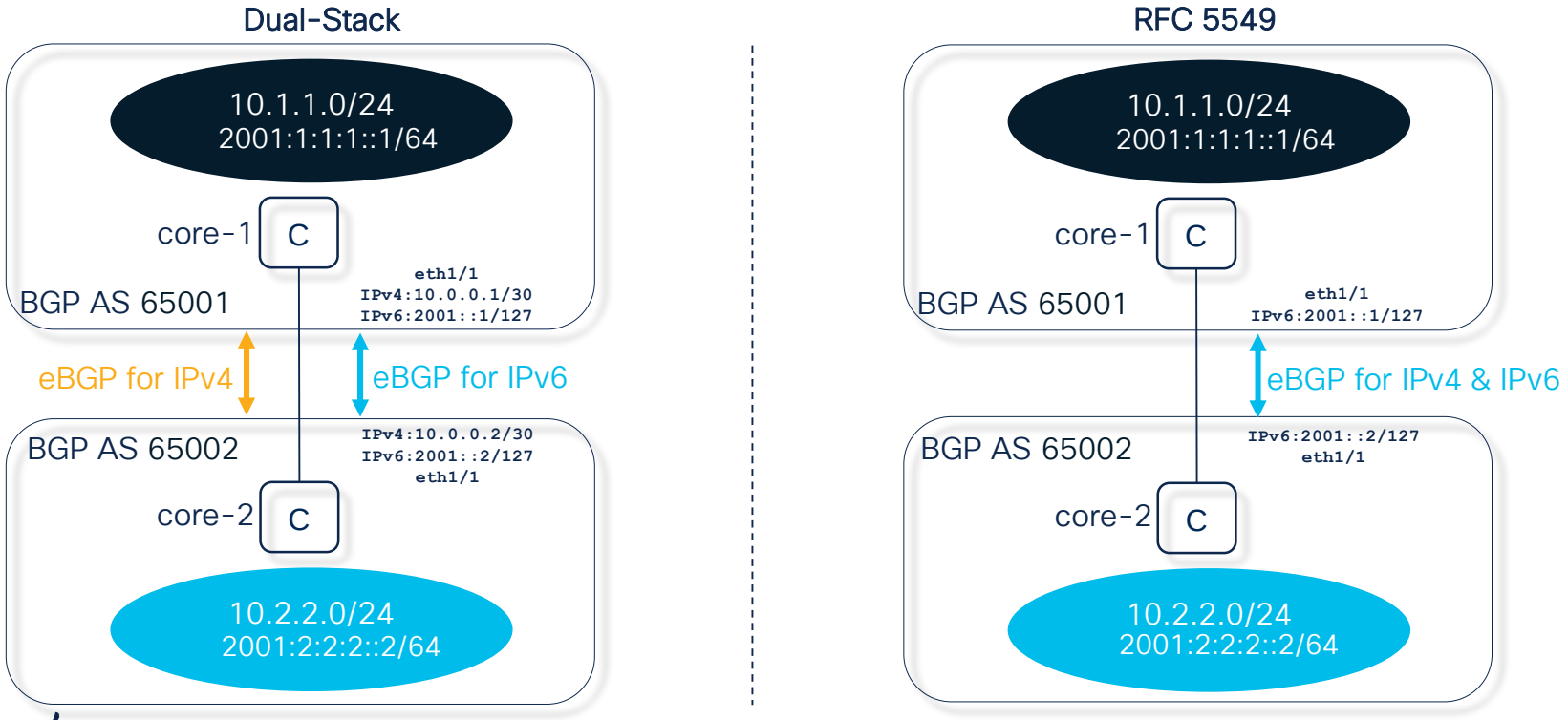
- Defines a specific behavior in BGP
- Allows IPv4 Network Layer Reachability via a IPv6 Next-Hop




```
core-1# show ip bgp
                        <<< output omitted for brevity >>>
   Network              Next Hop            Metric     LocPrf     Weight Path
*>l10.1.1.0/24          0.0.0.0                               100      32768 i
*>e10.2.2.0/24          2001::2                                             0 65002 i
```

Side-by-Side

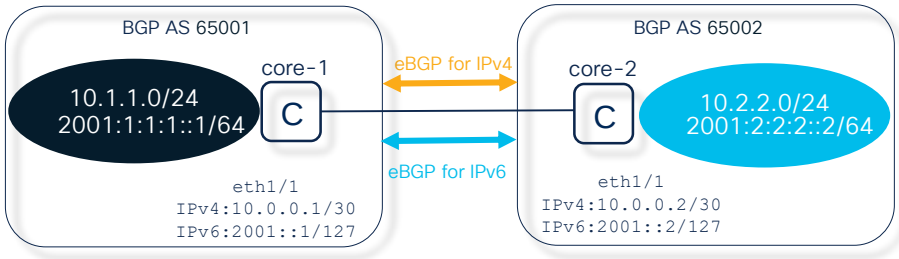
IPv4/IPv6 Dual-Stack and RFC 5549



Side-by-Side – Config with IPv6 Numbered

IPv4/IPv6 Dual-Stack and RFC 5549

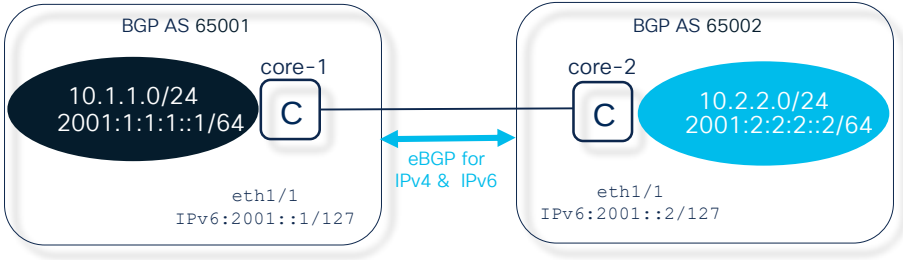
Dual-Stack



Per Address Family Peering

```
router bgp 65001
  neighbor 10.0.0.2
    remote-as 65002
    address-family ipv4 unicast
  neighbor 2001::2
    remote-as 65002
    address-family ipv6 unicast

interface Ethernet1/1
 ipv6 2001::1/127
 ip address 10.0.0.1/30
```

RFC 5549



Per Neighbor Peering

```
router bgp 65001
  neighbor 2001::2
    remote-as 65002
    address-family ipv4 unicast
    address-family ipv6 unicast

interface Ethernet1/1
 ipv6 2001::1/127
 ip forward
```

Side-by-Side – Config with IPv6 Numbered

IPv4/IPv6 Dual-Stack and RFC 5549

Dual-Stack BGP Table

```
core-1# show ip bgp
                        <<< output omitted for brevity >>>
   Network            Next Hop            Metric     LocPrf     Weight Path
*>e10.2.2.0/24        10.0.0.2                                         0 65002 I

core-1# show ipv6 bgp
                        <<< output omitted for brevity >>>

   Network            Next Hop            Metric     LocPrf     Weight Path
*>e2001:2:2:2::/64    2001::2                                          0 65002 i
```

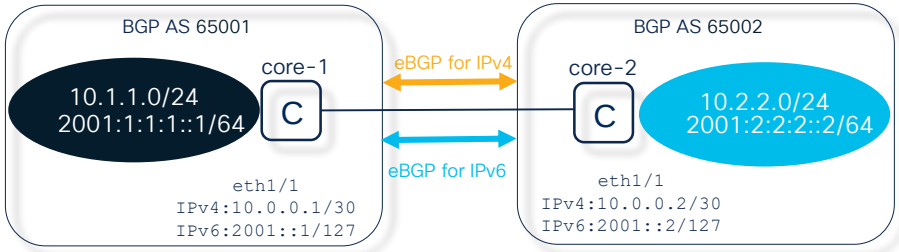
RFC 5549 BGP Table

```
core-1# show ip bgp
                        <<< output omitted for brevity >>>
   Network            Next Hop            Metric     LocPrf     Weight Path
*>e10.2.2.0/24        2001::2                                          0 65002 I

core-1# show ipv6 bgp
                        <<< output omitted for brevity >>>

   Network            Next Hop            Metric     LocPrf     Weight Path
*>e2001:2:2:2::/64    2001::2                                          0 65002 i
```

Side-by-Side – Config with Unnumbered (LLA)

IPv4/IPv6 Dual-Stack and RFC 5549

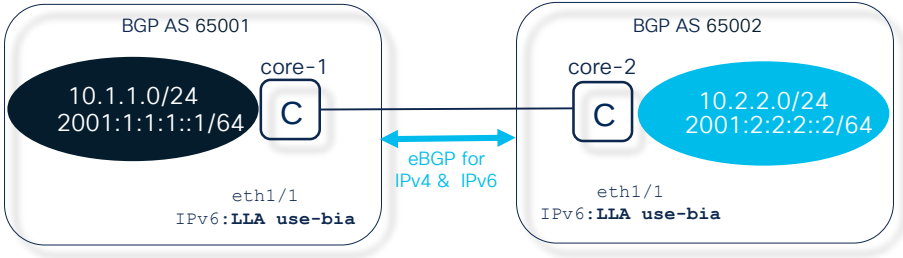
Dual-Stack



Per Address Family Peering

```
router bgp 65001
  neighbor 10.0.0.2
    remote-as 65002
    address-family ipv4 unicast
  neighbor 2001::2
    remote-as 65002
    address-family ipv6 unicast

interface Ethernet1/1
 ipv6 2001::1/127
 ip address 10.0.0.1/30
```

RFC 5549



Per Neighbor Unnumbered Peering

```
router bgp 65001
  neighbor Ethernet1/1
    remote-as 65002
    address-family ipv4 unicast
    address-family ipv6 unicast

interface Ethernet1/1
 ipv6 link-local use-bia
 ip forward
```

Using IPv6 Link-Local Addressing (LLA) and BGP Interface Peering

Side-by-Side – Config with Unnumbered (LLA)

IPv4/IPv6 Dual-Stack and RFC 5549

Dual-Stack BGP Table

```
core-1# show ip bgp
                        <<< output omitted for brevity >>>
   Network            Next Hop            Metric     LocPrf     Weight Path
*>e10.2.2.0/24        10.0.0.2                                         0 65002 I

core-1# show ipv6 bgp
                        <<< output omitted for brevity >>>

   Network            Next Hop            Metric     LocPrf     Weight Path
*>e2001:2:2:2::/64    2001::2                                          0 65002 i
```

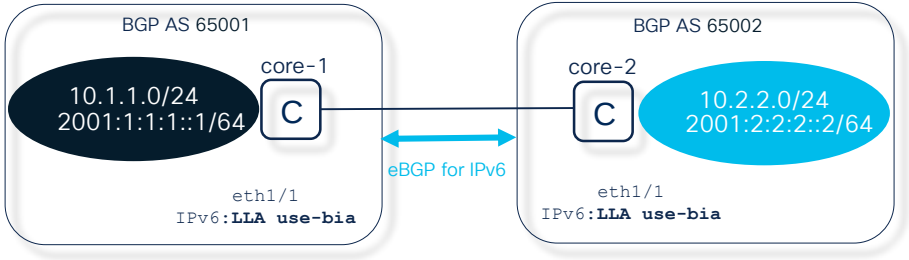
RFC 5549 BGP Table

```
core-1# show ip bgp
                        <<< output omitted for brevity >>>
   Network            Next Hop            Metric     LocPrf     Weight Path
*>e10.2.2.0/24        fe80::500e:45ff:fee4:101                         0 65002 I

core-1# show ipv6 bgp
                        <<< output omitted for brevity >>>

   Network            Next Hop            Metric     LocPrf     Weight Path
*>e2001:2:2:2::/64    fe80::500e:45ff:fee4:101                         0 65002 i
```

Deployment Simplification

IPv6 Link-Local and BGP Interface Peering

RFC 5549



Per Neighbor Unnumbered Peering

```
router bgp 65001
  neighbor Ethernet1/1
    remote-as 65002
    address-family ipv4 unicast
    address-family ipv6 unicast

interface Ethernet1/1
 ipv6 link-local use-bia
 ip forward
```

For the rest of this presentation, IPv6 Link-Local and BGP Interface Peering are used

‘How Does This Fit in the Data Center?’

RFC 5549 Use Case

RFC 5549 Use Cases

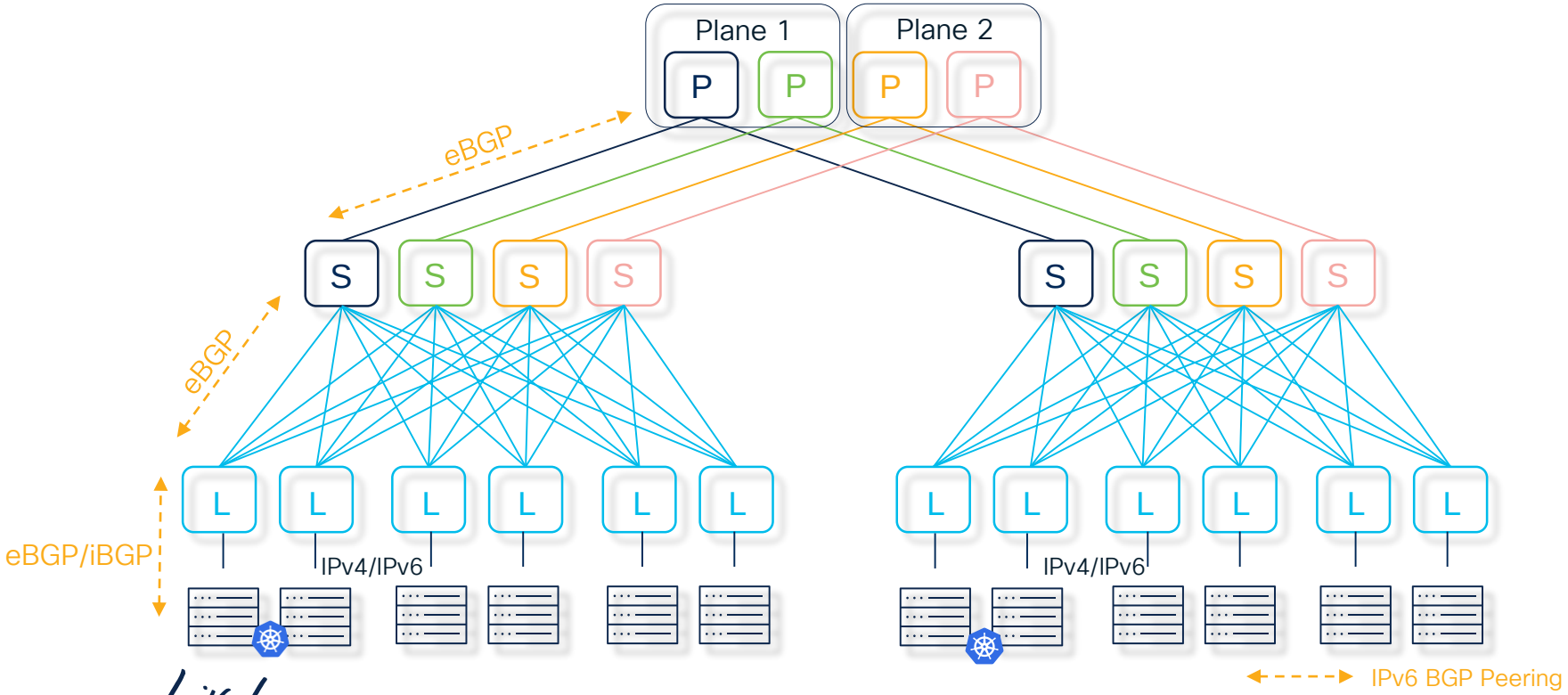
From the Data Center Playbook

- Use of BGP for Routing in Large-Scale Data Centers
  - RFC 7938: https://datatracker.ietf.org/doc/html/rfc7938
- Used as Routing Protocol between Leaf, Spine, and other Tiers (ie.: Super Spines, Spine Planes)
  - IPv4 and IPv6 Prefix with a single Routing Protocol Session – No VRF, VPNs or Overlays
  - Ready for “Cloud Native Applications”* – no need for Layer-2
- Deployment simplification with IPv6 (a way to start, if not already)

*Generally defined as Container or Kubernetes-based Applications

Routed Fabrics

Combining RFC 7938 and RFC 5549



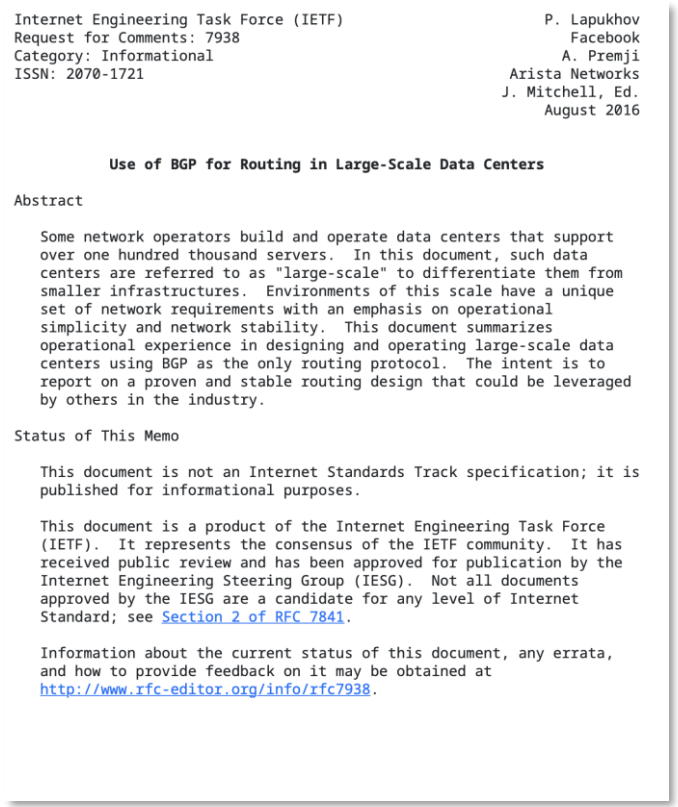
‘Use of BGP for Routing in Large-Scale Data Centers’

RFC 7938

What is RFC 7938?

By the Standards Body

- Categorized as Informational RFC
- Basically, a Design Guide for Leaf/Spine Topologies
- Chooses eBGP as Routing Protocol for the Data Center
  - A flat Layer-3 only approach
  - No Network Overlays considered
- Is RFC 7938 dated?
  - No specific reference to IPv6
  - Only 2-Byte ASN reference
  - Talks about TRILL for Layer-2



Internet Engineering Task Force (IETF)
Request for Comments: 7938
Category: Informational
ISSN: 2070-1721

P. Lapukhov
Facebook
A. Premji
Arista Networks
J. Mitchell, Ed.
August 2016

Use of BGP for Routing in Large-Scale Data Centers

Abstract

Some network operators build and operate data centers that support over one hundred thousand servers. In this document, such data centers are referred to as "large-scale" to differentiate them from smaller infrastructures. Environments of this scale have a unique set of network requirements with an emphasis on operational simplicity and network stability. This document summarizes operational experience in designing and operating large-scale data centers using BGP as the only routing protocol. The intent is to report on a proven and stable routing design that could be leveraged by others in the industry.

Status of This Memo

This document is not an Internet Standards Track specification; it is published for informational purposes.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Not all documents approved by the IESG are a candidate for any level of Internet Standard; see Section 2 of RFC 7841.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at http://www.rfc-editor.org/info/rfc7938.

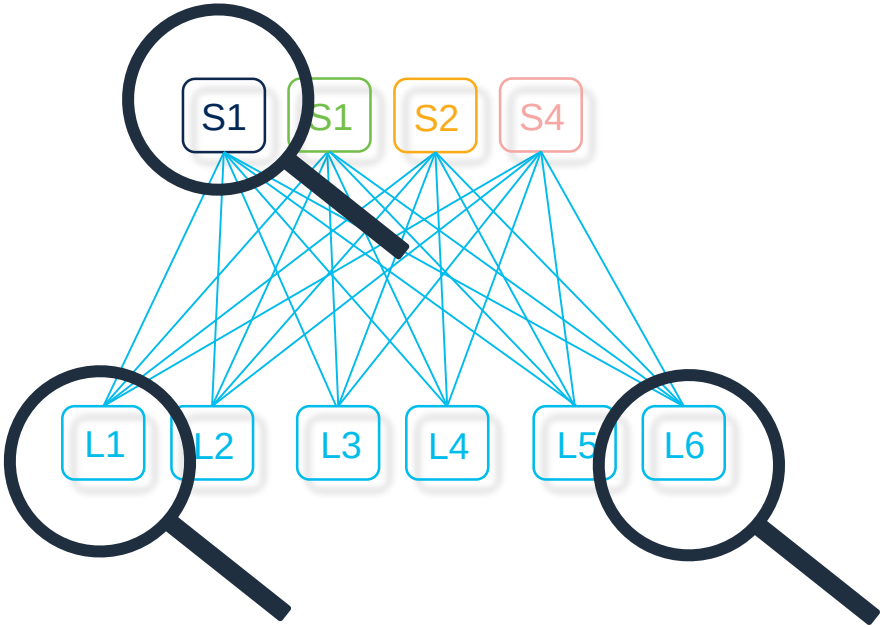
https://datatracker.ietf.org/doc/html/rfc7938

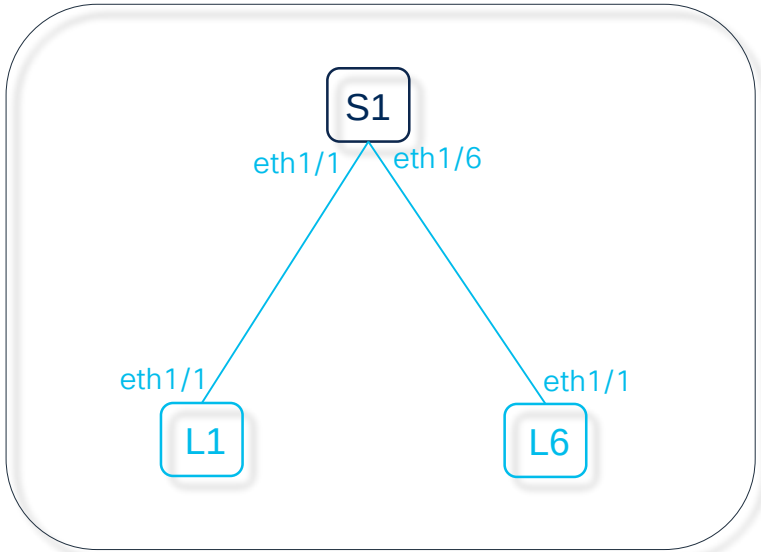
‘Advertising IPv4 & IPv6 Prefix Information with an IPv6 Next Hop enables BGP for Routing in Large-Scale Data Centers to carry IPv4 & IPv6 Address-Family’

RFC 5549 + RFC 7938

Deployments with RFC 5549 at a glance

Magnifying some Nodes

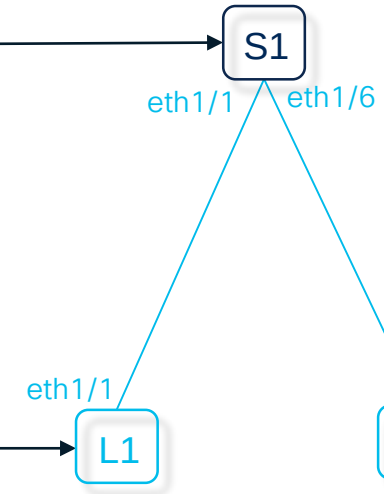




Adding Loopbacks

```
interface loopback0
ip address 10.51.51.51/32 tag 12345
ipv6 address 2001::51/128 tag 12345

router bgp 65111
 address-family ipv4 unicast
  redistribute direct route-map TAG
 address-family ipv6 unicast
  redistribute direct route-map TAG
```

```
[all]
route-map TAG permit 10
  match tag 12345
```

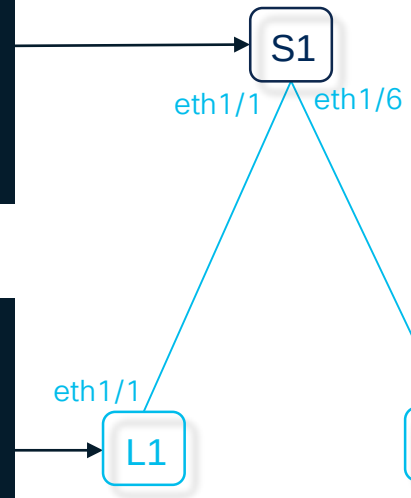



```
interface loopback0
ip address 10.31.31.31/32 tag 12345
ipv6 address 2001::31/128 tag 12345

router bgp 65001
 address-family ipv4 unicast
  redistribute direct route-map TAG
 address-family ipv6 unicast
  redistribute direct route-map TAG
```

```
interface loopback0
ip address 10.36.36.36/32 tag 12345
ipv6 address 2001::36/128 tag 12345

router bgp 65001
 address-family ipv4 unicast
  redistribute direct route-map TAG
 address-family ipv6 unicast
  redistribute direct route-map TAG
```

Config per RFC 7938 (Dual-AS)

```
router bgp 65111
  neighbor Ethernet1/1, 1/6
    remote-as 65001
    address-family ipv4 unicast
    address-family ipv6 unicast

interface Ethernet1/1, 1/6
 ip forward
 ipv6 link-local use-bia
```




```
router bgp 65001
  neighbor Ethernet1/1
    remote-as 65111
    address-family ipv4 unicast
    address-family ipv6 unicast

interface Ethernet1/1
 ip forward
 ipv6 link-local use-bia
```

```
router bgp 65001
  neighbor Ethernet1/1
    remote-as 65111
    address-family ipv4 unicast
    address-family ipv6 unicast

interface Ethernet1/1
 ip forward
 ipv6 link-local use-bia
```

This is NOT going to work (Source AS = Destination AS)

Config per RFC 7938 (Dual-AS)

L1

IPv4 Table

```
leaf-1# show ip route
                      <<< output omitted for brevity >>>

10.31.31.31/32, ubest/mbest: 2/0, attached
    *via 10.31.31.31, Lo0, [0/0], 00:27:53, local, tag 12345
    *via 10.31.31.31, Lo0, [0/0], 00:27:53, direct, tag 12345

leaf-1# show ip bgp
                      <<< output omitted for brevity >>>

   Network            Next Hop            Metric     LocPrf     Weight Path
*>r10.31.31.31/32  0.0.0.0                  0        100         32768 ?
```

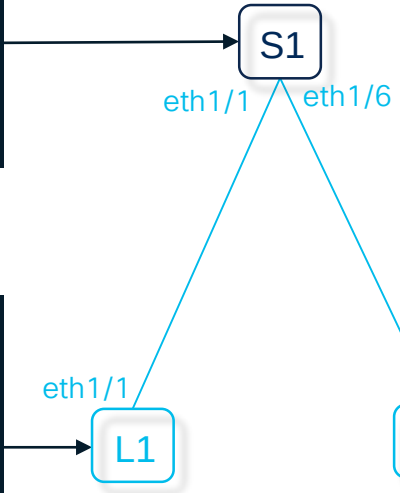
No Routes from the other leaf (same AS)

There are 2 Ways to Remediate

Config per RFC 7938 (Dual-AS with Knobs)

```
router bgp 65111
  neighbor Ethernet1/1, 1/6
    remote-as 65001
    inherit peer DISABLE-AS-PATH-CHECK

interface Ethernet1/1, 1/6
 ip forward
 ipv6 link-local use-bia
```

```
[all]
 template peer DISABLE-AS-PATH-CHECK
    address-family ipv4 unicast
      allowas-in 3
      disable-peer-as-check
    address-family ipv6 unicast
      allowas-in 3
      disable-peer-as-check
```




```
router bgp 65001
  neighbor Ethernet1/1
    remote-as 65111
    inherit peer DISABLE-AS-PATH-CHECK

interface Ethernet1/1
 ip forward
 ipv6 link-local use-bia
```

```
router bgp 65001
  neighbor Ethernet1/1
    remote-as 65111
    inherit peer DISABLE-AS-PATH-CHECK

interface Ethernet1/1
 ip forward
 ipv6 link-local use-bia
```

Config per RFC 7938 (Dual-AS with Knobs)

L1

IPv4 Table

```
leaf-1# show ip route
                                  <<< output omitted for brevity >>>

10.31.31.31/32, ubest/mbest: 2/0, attached
    *via 10.31.31.31, Lo0, [0/0], 00:21:34, local, tag 12345
    *via 10.31.31.31, Lo0, [0/0], 00:21:34, direct, tag 12345
10.36.36.36/32, ubest/mbest: 1/0
    *via fe80::5008:6fff:fe29:101%default, Eth1/1, [20/0], 00:04:18, bgp-65001, external, tag 65111

leaf-1# show ip bgp
                                  <<< output omitted for brevity >>>

   Network            Next Hop            Metric     LocPrf     Weight Path
*>r10.31.31.31/32     0.0.0.0                  0        100      32768 ?
*>e10.36.36.36/32     fe80::5008:6fff:fe29:101                      0 65111 65001 ?
```

Routes Received from the other leaf (same AS)

Option #1: Dual-AS with allow-as-in + disable-peer-as-check knobs

Config per RFC 7938 (Dual-AS with Knobs)

L1

IPv6 Table

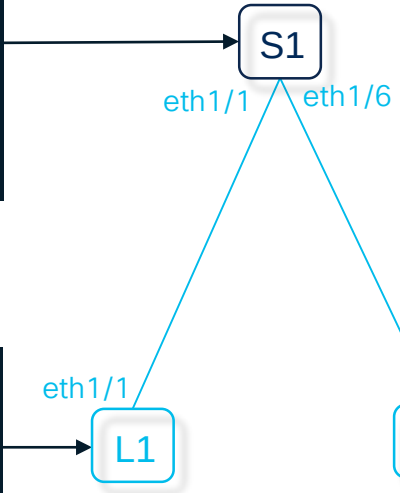
```
leaf-1# show ipv6 route
                                  <<< output omitted for brevity >>>

2001::31/128, ubest/mbest: 2/0, attached
    *via 2001::31, Lo0, [0/0], 00:44:10, direct, , tag 12345
    *via 2001::31, Lo0, [0/0], 00:44:10, local, tag 12345
2001::36/128, ubest/mbest: 1/0
    *via fe80::5008:6fff:fe29:101, Eth1/1, [20/0], 00:03:10, bgp-65001, external, tag 65111

leaf-1# show ipv6 bgp
                                  <<< output omitted for brevity >>>

   Network            Next Hop            Metric     LocPrf     Weight Path
*>r2001::31/128       0::                      0        100      32768 ?
*>e2001::36/128       fe80::5008:6fff:fe29:101                      0 65111 65001 ?
```

Routes Received from the other leaf (same AS)

Option #1: Dual-AS with allow-as-in + disable-peer-as-check knobs

Config of Multi-AS

```
router bgp 65111
  neighbor Ethernet1/1
    remote-as 65001
    address-family ipv4 unicast
    address-family ipv6 unicast
 neighbor Ethernet1/6
    remote-as 65006
    address-family ipv4 unicast
    address-family ipv6 unicast
```




```
router bgp 65001
  neighbor Ethernet1/1
    remote-as 65111
    address-family ipv4 unicast
    address-family ipv6 unicast
```

```
router bgp 65006
  neighbor Ethernet1/1
    remote-as 65111
    address-family ipv4 unicast
    address-family ipv6 unicast
```

Option #2: Multi-AS – Each switch will get its own AS

Config of Multi-AS

L1

IPv4 Table

```
leaf-1# show ip route
                                  <<< output omitted for brevity >>>

10.31.31.31/32, ubest/mbest: 2/0, attached
    *via 10.31.31.31, Lo0, [0/0], 00:24:50, local, tag 12345
    *via 10.31.31.31, Lo0, [0/0], 00:24:50, direct, tag 12345
10.36.36.36/32, ubest/mbest: 1/0
    *via fe80::5016:9cff:fe03:101%default, Eth1/1, [20/0], 00:00:48, bgp-65001, external, tag 65111

Leaf-1# show ip bgp
                                  <<< output omitted for brevity >>>

   Network            Next Hop            Metric     LocPrf     Weight Path
*>r10.31.31.31/32     0.0.0.0                  0        100      32768 ?
*>e10.36.36.36/32     fe80::5016:9cff:fe03:101                      0 65111 65006 ?
```

Option #2: Multi-AS – Each switch will get its own AS

Config of Multi-AS

L1

IPv6 Table

```
leaf-1# show ipv6 route
                                    <<< output omitted for brevity >>>

2001::31/128, ubest/mbest: 2/0, attached
    *via 2001::31, Lo0, [0/0], 00:31:38, direct, , tag 12345
    *via 2001::31, Lo0, [0/0], 00:31:38, local, tag 12345
2001::36/128, ubest/mbest: 1/0
    *via fe80::5016:9cff:fe03:101, Eth1/1, [20/0], 00:07:45, bgp-65001, external, tag 65111

leaf-1# show ipv6 bgp
                                    <<< output omitted for brevity >>>

   Network            Next Hop            Metric     LocPrf     Weight Path
*>r2001::31/128       0::                      0        100      32768 ?
*>e2001::36/128       fe80::5016:9cff:fe03:101                      0 65111 65006 ?
```

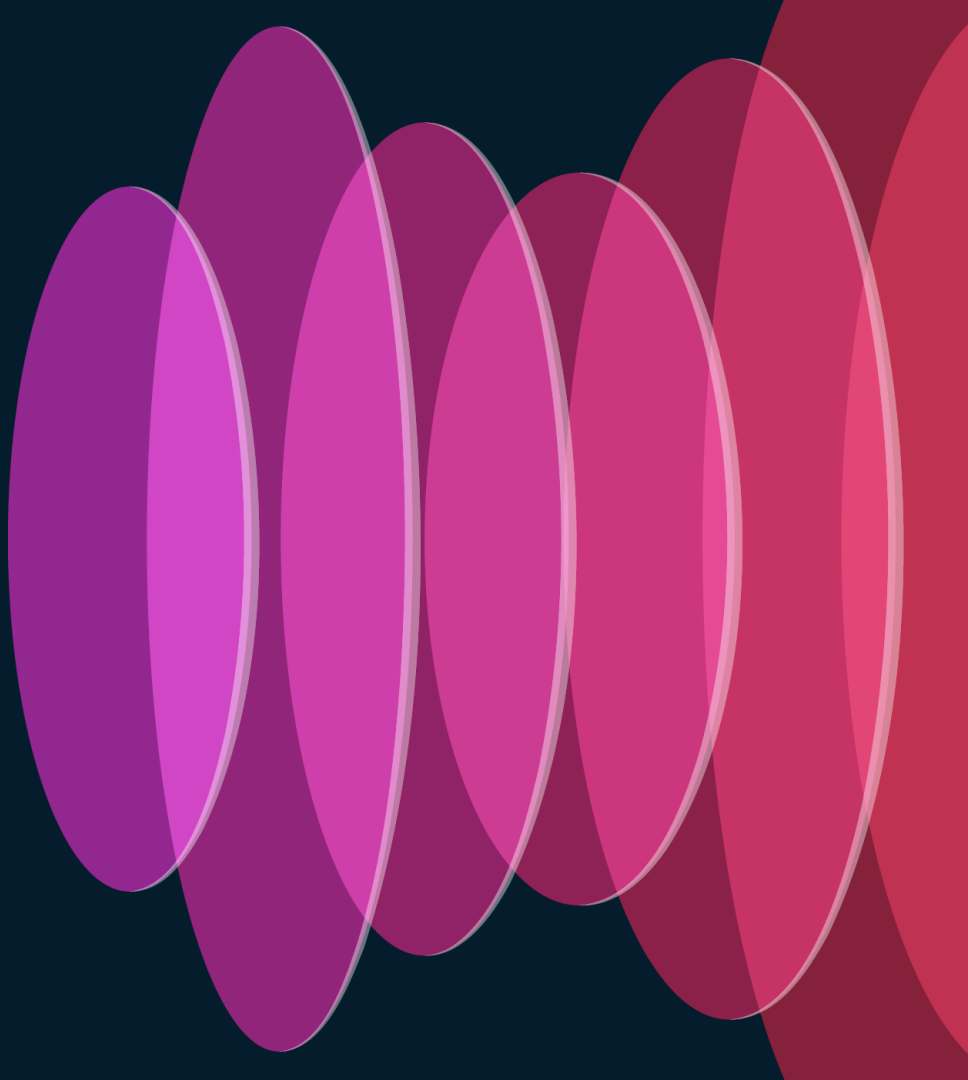
Option #2: Multi-AS – Each switch will get its own AS

Traceroute: Using Loopbacks

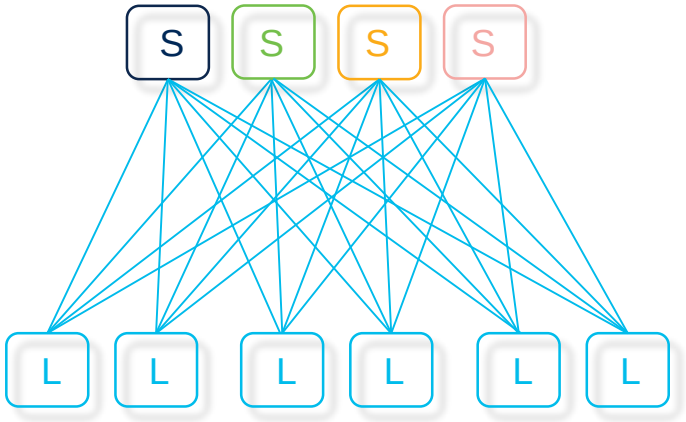
Reference Slides provided at the End

- A Loopback per Switch helps in Operational Tasks
  - For IPv4, add a IPv4 Loopback. For IPv6, add a IPv6 Loopback
- Ping for Connectivity Test
  - Loopback to Loopback
  - Physical Interface to Physical Interface (Link-Local Address)
- Traceroute becomes easy to Read
  - Each Hop clearly identified by the Loopback IP address (IPv4 or IPv6)
  - In Leaf/Spine, Loopback address is sufficient (there is no other path)
- In-band Management (Loopback to Loopback or LLA to LLA)

BGP Auto-Fabric



What is BGP Auto-Fabric?

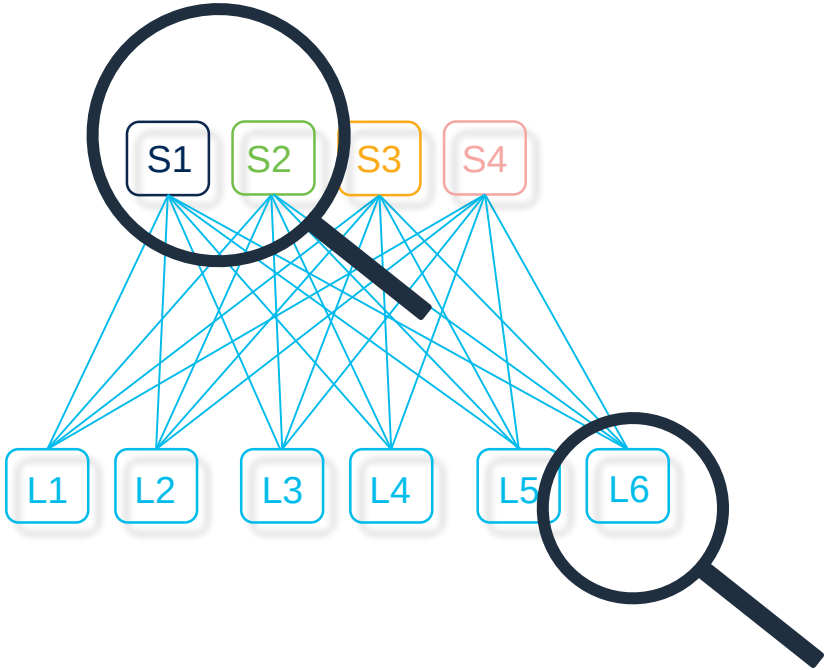


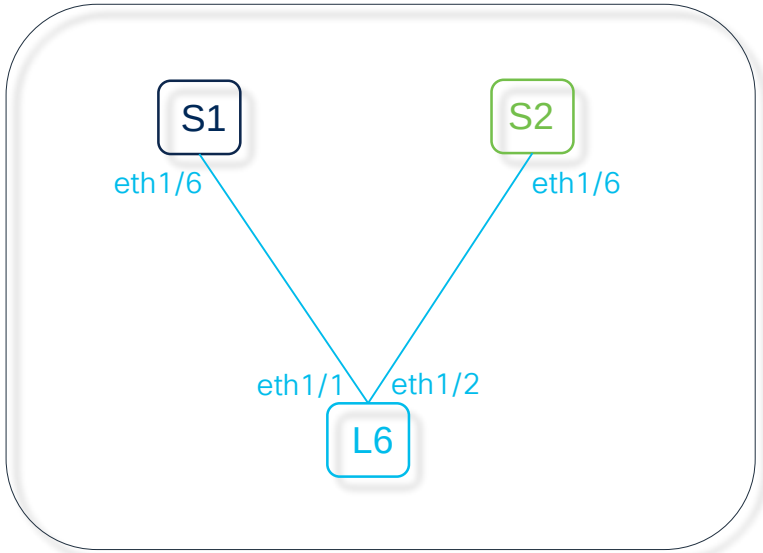
Self-Organized BGP Fabric

- Autonomously Derives Key Values for BGP
- Avoids Per-Interface IP Addressing
- Automates BGP ASN (4-byte) and Router-ID
- Simplifies BGP Peer Configuration

BGP Auto-Fabric at a glance

Magnifying some Nodes

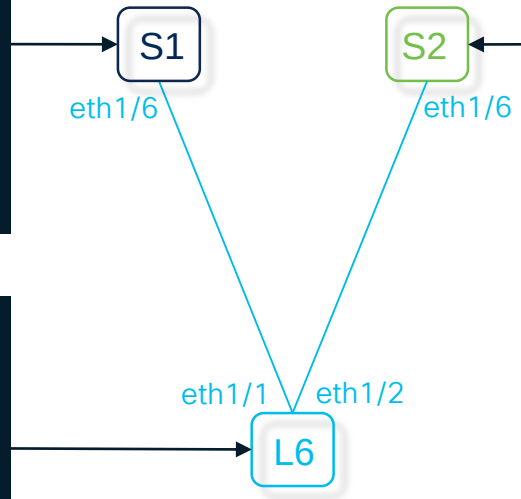




BGP Auto-Fabric Config

```
router bgp auto
  router-id auto
  neighbor Ethernet1/6
    remote-as external
    address-family ipv4 unicast
    address-family ipv6 unicast

interface Ethernet1/6
  ip forward
  ipv6 link-local use-bia
```

```
router bgp auto
  router-id auto
  neighbor Ethernet1/6
    remote-as external
    address-family ipv4 unicast
    address-family ipv6 unicast

interface Ethernet1/6
  ip forward
  ipv6 link-local use-bia
```




```
router bgp auto
  router-id auto
  neighbor Ethernet1/1
    remote-as external
    address-family ipv4 unicast
    address-family ipv6 unicast

interface Ethernet1/1-2
 ip forward
 ipv6 link-local use-bia
```

Config Results – Auto Derived

```
router bgp auto
  router-id auto
  neighbor Ethernet1/6
    remote-as external
    address-family ipv4 unicast
    address-family ipv6 unicast

interface Ethernet1/6
  ip forward
  ipv6 link-local use-bia
```

Auto/Peering Seed Values
Global: [System MAC]
Per-Interface: [Interface MAC]

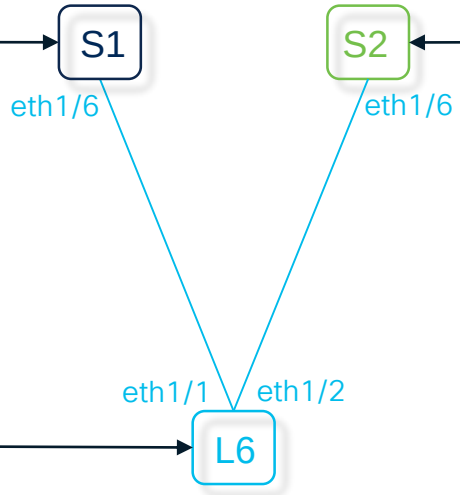
S1

```
spine-1# show bgp sessions
                        <<< output omitted for brevity >>>
ASN 4263297326
VRF default, local ASN 4263297326
peers 1, established peers 1, local router-id 218.223.27.8
State: I-Idle, A-Active, O-Open, E-Established, C-Closing, S-Shutdown

Neighbor        ASN    Flaps LastUpDn|LastRead|LastWrit St Port(L/R)   Notif(S/R)
fe80::5001:60ff:fed8:101%Ethernet1/6
           4215095057 0     00:12:27|00:00:26|00:00:26 E   179/51108        0/0

spine-1# show ipv6 int brief
                        <<< output omitted for brevity >>>

Eth1/6           fe80::501b:93ff:fedf:106                 up/up/up
                 fe80::501b:93ff:fedf:106
spine-1#
spine-1# show ip int brief
                        <<< output omitted for brevity >>>

Eth1/6               forward-enabled protocol-up/link-up/admin-up
```

Same Loopbacks, New BGP Config

```
interface loopback0
ip address 10.51.51.51/32 tag 12345
ipv6 address 2001::51/128 tag 12345

router bgp auto
 address-family ipv4 unicast
  redistribute direct route-map TAG
 address-family ipv6 unicast
  redistribute direct route-map TAG
```

```
interface loopback0
ip address 10.52.52.52/32 tag 12345
ipv6 address 2001::52/128 tag 12345

router bgp auto
 address-family ipv4 unicast
  redistribute direct route-map TAG
 address-family ipv6 unicast
  redistribute direct route-map TAG
```




```
interface loopback0
ip address 10.36.36.36/32 tag 12345
ipv6 address 2001::36/128 tag 12345

router bgp auto
 address-family ipv4 unicast
  redistribute direct route-map TAG
 address-family ipv6 unicast
  redistribute direct route-map TAG
```

```
[all]
route-map TAG permit 10
  match tag 12345
```

IPv4 Routing Output Example

S1

```
spine-1# show ip route
                                  <<< output omitted for brevity >>>
10.36.36.36/32, ubest/mbest: 1/0
    *via fe80::5001:60ff:fed8:101%default, Eth1/6, [20/0], 00:06:57, bgp-auto, external, tag 4215095057
10.51.51.51/32, ubest/mbest: 2/0, attached
    *via 10.51.51.51, Lo0, [0/0], 00:13:38, local, tag 12345
    *via 10.51.51.51, Lo0, [0/0], 00:13:38, direct, tag 12345
10.52.52.52/32, ubest/mbest: 1/0
    *via fe80::5001:60ff:fed8:101%default, Eth1/6, [20/0], 00:11:05, bgp-auto, external, tag 4215095057

spine-1# show ip bgp
BGP table version is 9, Local Router ID is 218.223.27.8
                                  <<< output omitted for brevity >>>
   Network            Next Hop            Metric     LocPrf     Weight Path
*>e10.36.36.36/32     fe80::5001:60ff:fed8:101 0                        0 4215095057 ?
*>r10.51.51.51/32     0.0.0.0                  0        100      32768 ?
*>e10.52.52.52/32     fe80::5001:60ff:fed8:101                          0 4215095057 4231235365 ?
```

IPv6 Routing Output Example

S1

```
spine-1# show ipv6 route
                                  <<< output omitted for brevity >>>
2001::36/128, ubest/mbest: 1/0
    *via fe80::5001:60ff:fed8:101, Eth1/6, [20/0], 00:24:29, bgp-auto, external, tag 4215095057
2001::51/128, ubest/mbest: 2/0, attached
    *via 2001::51, Lo0, [0/0], 00:30:52, direct, , tag 12345
    *via 2001::51, Lo0, [0/0], 00:30:52, local, tag 12345
2001::52/128, ubest/mbest: 1/0
    *via fe80::5001:60ff:fed8:101, Eth1/6, [20/0], 00:27:58, bgp-auto, external, tag 4215095057

spine-1# show ipv6 bgp
BGP table version is 9, Local Router ID is 218.223.27.8
                                  <<< output omitted for brevity >>>
   Network            Next Hop            Metric     LocPrf     Weight Path
*>e2001::36/128       fe80::5001:60ff:fed8:101 0                        0 4215095057 ?
*>r2001::51/128       0::                      0        100      32768 ?
*>e2001::52/128       fe80::5001:60ff:fed8:101                          0 4215095057 4231235365 ?
```

Routed Fabrics

Summary

- A reason to explore RFC 5549
- The benefits of combining it with RFC 7938, if it makes sense
- Additional deployment simplification with BGP auto-fabric

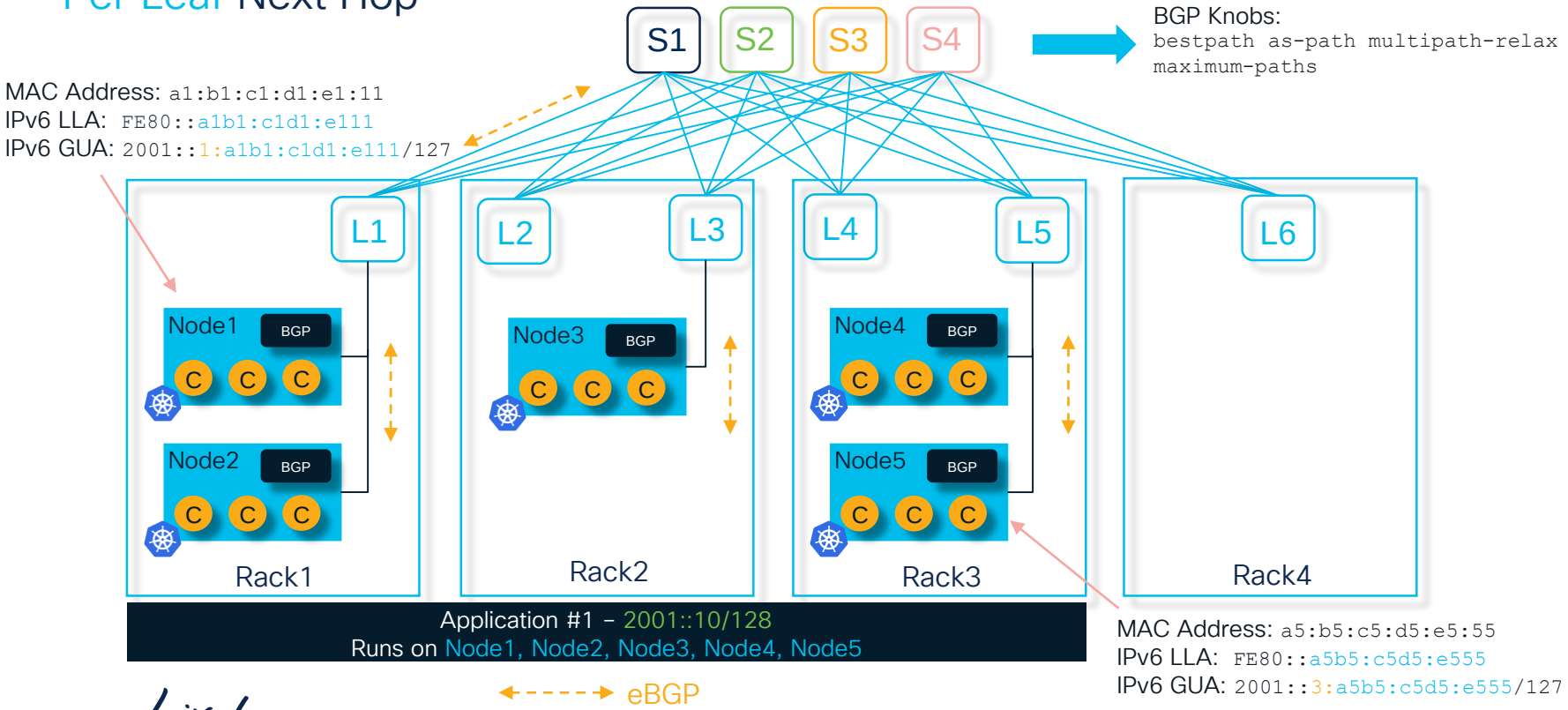
‘Kubernetes (K8s) Infrastructure Connectivity’

Network Designs for the Modern Data Center

Node IP Addressing Example
IPv6 LLA: `FE80::MAC`
IPv6 GUA: `2001::RackID:NodeID`

BGP Load Balancing

Per Leaf Next Hop



BGP Load Balancing

Per Leaf Next Hop

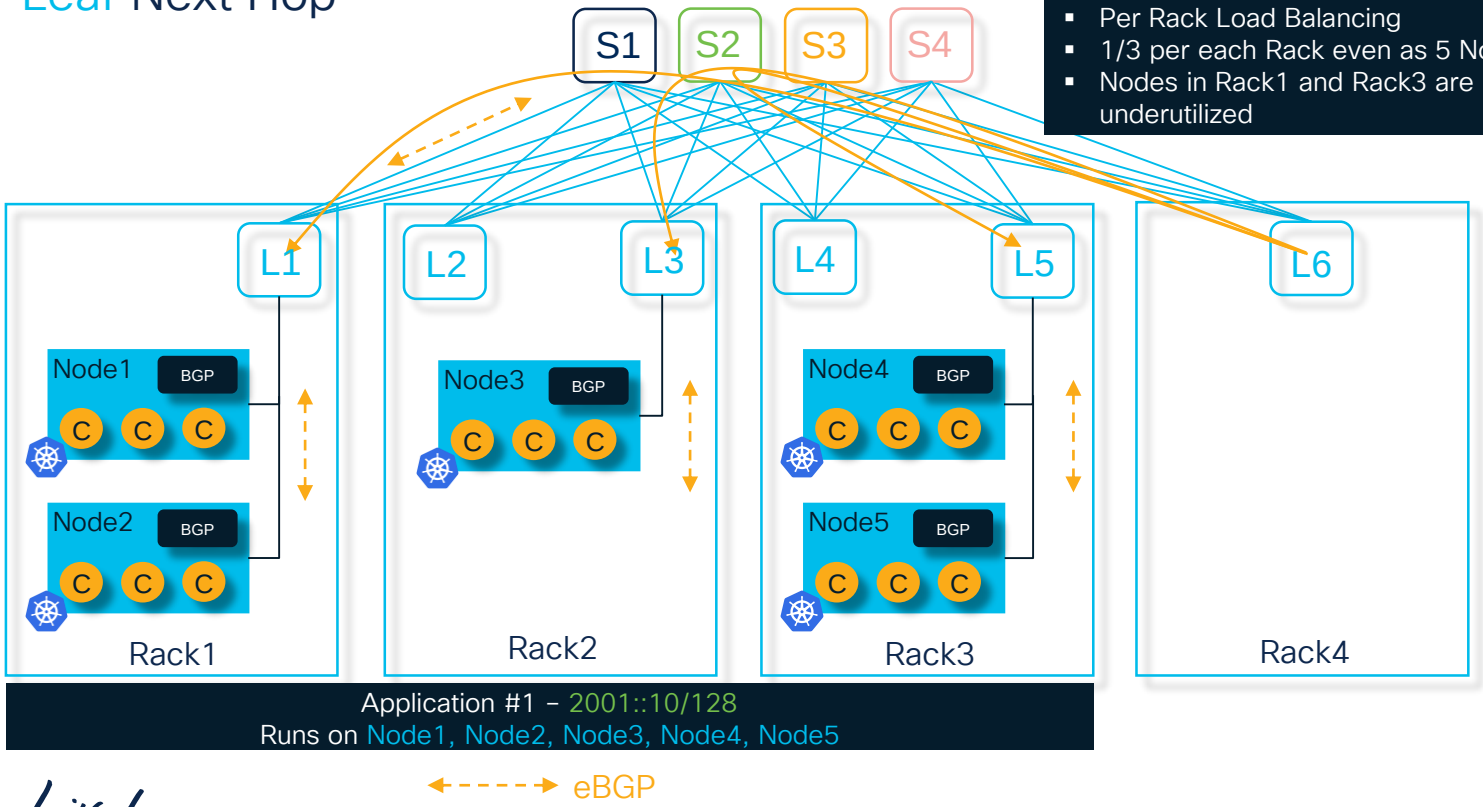
S1

```
spine-1# show ipv6 route
                                <<< output omitted for brevity >>>
2001::10/128, ubest/mbest: 3/0
    *via fe80::5009:beff:fea8:101, Eth1/3, [20/0], 00:00:30, bgp-auto, external, tag 4254307632
    *via fe80::500f:a6ff:fe0f:101, Eth1/1, [20/0], 00:00:30, bgp-auto, external, tag 4233510196
    *via fe80::501f:31ff:fe82:101, Eth1/5, [20/0], 00:00:30, bgp-auto, external, tag 4294588189

2001::1:a1b1:c1d1:e111/128, ubest/mbest: 1/0
    *via fe80::500f:a6ff:fe0f:101, Eth1/1, [20/0], 00:44:40, bgp-auto, external, tag 4233510196
2001::1:a2b2:c2d2:e222/128, ubest/mbest: 1/0
    *via fe80::500f:a6ff:fe0f:101, Eth1/1, [20/0], 00:44:45, bgp-auto, external, tag 4233510196
2001::2:a3b3:c3d3:e333/128, ubest/mbest: 1/0
    *via fe80::5009:beff:fea8:101, Eth1/3, [20/0], 00:08:51, bgp-auto, external, tag 4254307632
2001::3:a4b4:c4d4:e444/128, ubest/mbest: 1/0
    *via fe80::501f:31ff:fe82:101, Eth1/5, [20/0], 00:05:07, bgp-auto, external, tag 4294588189
2001::3:a5b5:c5d5:e555/128, ubest/mbest: 1/0
    *via fe80::501f:31ff:fe82:101, Eth1/5, [20/0], 00:00:22, bgp-auto, external, tag 4294588189
```

Load Balancing to where the Server connects (Leaf)

BGP Load Balancing

Per Leaf Next Hop



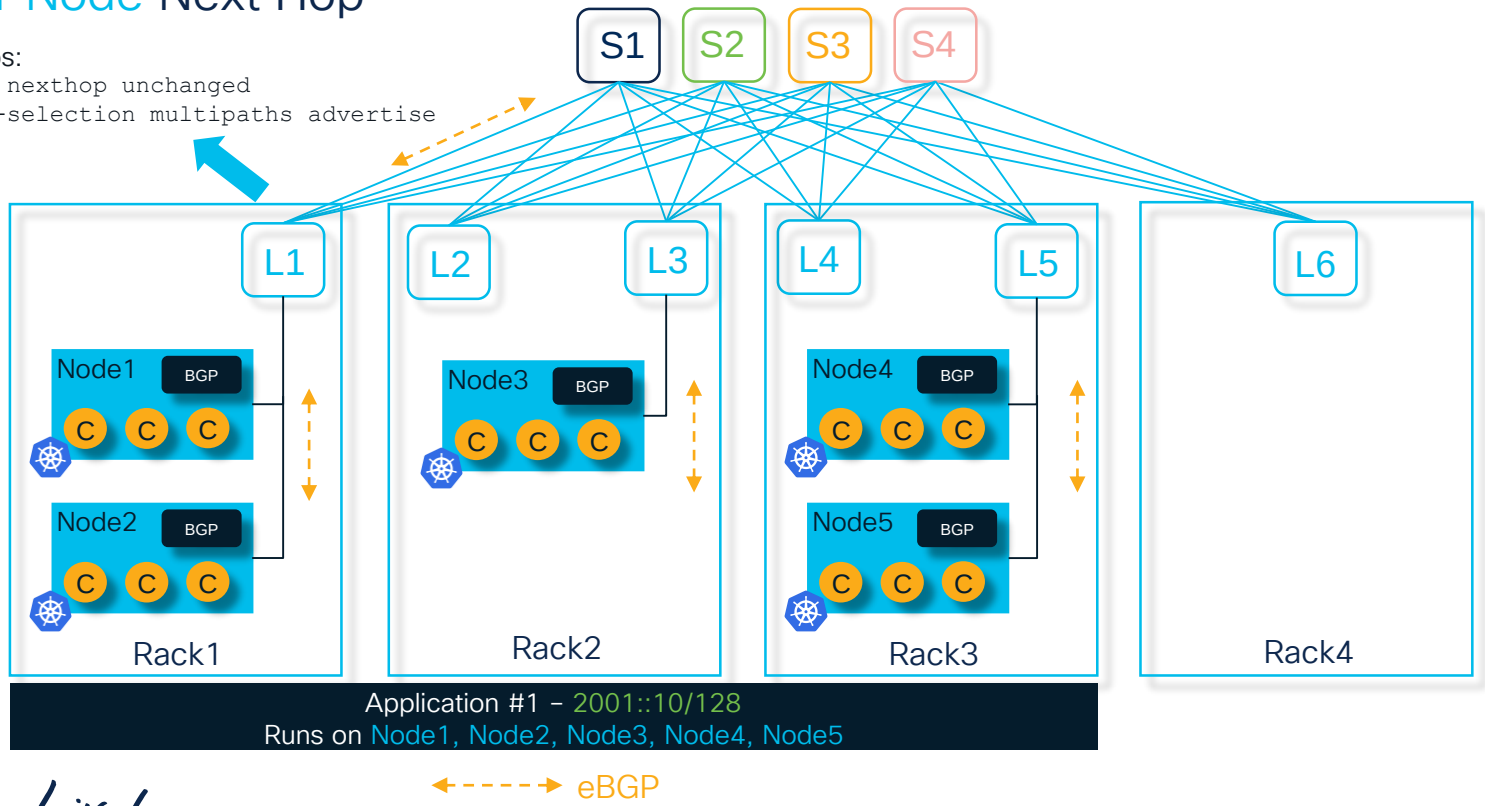
- Per Rack Load Balancing
- 1/3 per each Rack even as 5 Nodes exists
- Nodes in Rack1 and Rack3 are probably underutilized

Optimized BGP Load Balancing

Per Node Next Hop

BGP Knobs:

```
set ipv6 nexthop unchanged
set path-selection multipaths advertise
```



BGP Load Balancing

Per Leaf Next Hop

S1

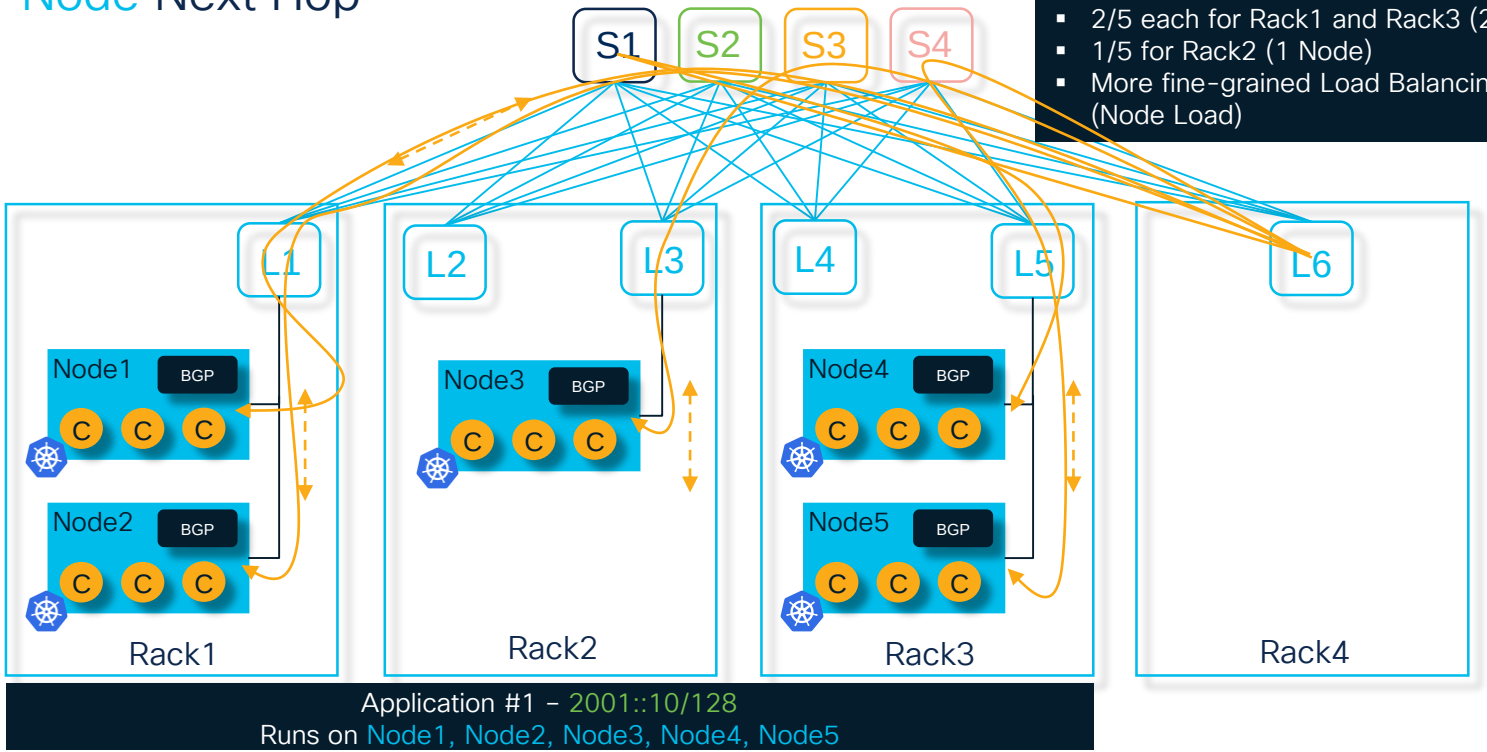
```
spine-1# show ipv6 route
                                <<< output omitted for brevity >>>
2001::10/128, ubest/mbest: 1/0
    *via 2001::1:a1b1:c1d1:e111/128, [20/0], 00:01:11, bgp-auto, external, tag 4233510196
    *via 2001::1:a2b2:c2d2:e222/128, [20/0], 00:01:11, bgp-auto, external, tag 4233510196
    *via 2001::2:a3b3:c3d3:e333/128, [20/0], 00:01:11, bgp-auto, external, tag 4254307632
    *via 2001::3:a4b4:c4d4:e444/128, [20/0], 00:01:11, bgp-auto, external, tag 4294588189
    *via 2001::3:a5b5:c5d5:e555/128, [20/0], 00:01:11, bgp-auto, external, tag 4294588189

2001::1:a1b1:c1d1:e111/128, ubest/mbest: 1/0
    *via fe80::500f:a6ff:fe0f:101, Eth1/1, [20/0], 00:44:40, bgp-auto, external, tag 4233510196
2001::1:a2b2:c2d2:e222/128, ubest/mbest: 1/0
    *via fe80::500f:a6ff:fe0f:101, Eth1/1, [20/0], 00:44:45, bgp-auto, external, tag 4233510196
2001::2:a3b3:c3d3:e333/128, ubest/mbest: 1/0
    *via fe80::5009:beff:fea8:101, Eth1/3, [20/0], 00:08:51, bgp-auto, external, tag 4254307632
2001::3:a4b4:c4d4:e444/128, ubest/mbest: 1/0
    *via fe80::501f:31ff:fe82:101, Eth1/5, [20/0], 00:05:07, bgp-auto, external, tag 4294588189
2001::3:a5b5:c5d5:e555/128, ubest/mbest: 1/0
    *via fe80::501f:31ff:fe82:101, Eth1/5, [20/0], 00:00:22, bgp-auto, external, tag 4294588189
```

Load Balancing to where the Application runs (Node)

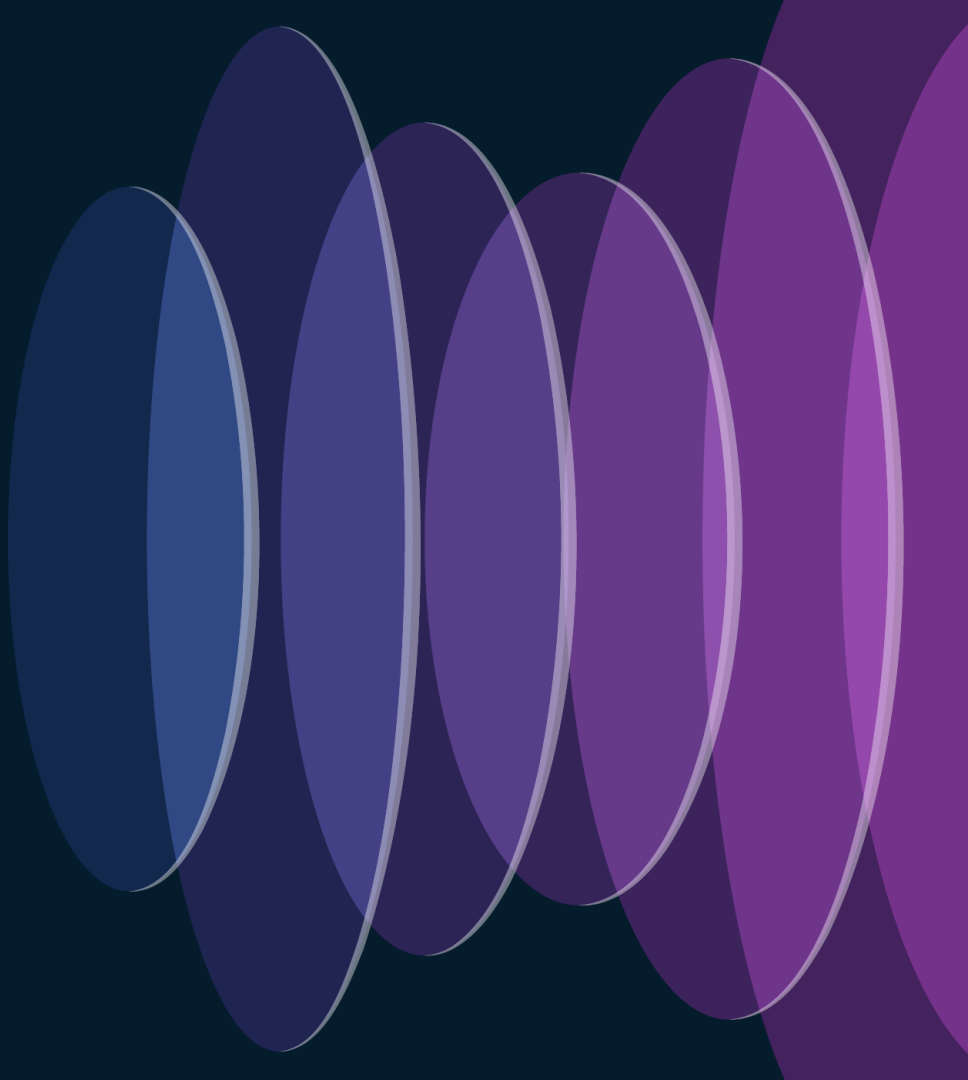
Optimized BGP Load Balancing

Per Node Next Hop

- Per Available Node Load Balancing
- 2/5 each for Rack1 and Rack3 (2 Nodes each)
- 1/5 for Rack2 (1 Node)
- More fine-grained Load Balancing Possible (Node Load)



Conclusion



‘We can scale from Small to Very Large – Don’t be shy starting with a Small Setup; we can Evolve!’

Key Takeaway #1

‘Bigger is not Always Better; Using Fixed-Form factor Switches is a Modern Practice’

Key Takeaway #2

‘More Switches != Higher Cost’

Key Takeaway #3

‘Routed Fabrics is a Real Thing ’ 

Key Takeaway #4

‘If you are still not convinced on IPv6, or want to learn more; I recommend you watch Nicole’s session BRKENT-2109’

Key Takeaway #5

Additional Resources

- RFC 5549
  - See “Nexus 9000 Series NX-OS Unicast Routing Configuration Guide” – Advanced BGP section
- BGP Auto-Fabric
  - See “Nexus 9000 Series NX-OS Unicast Routing Configuration Guide” – Basic BGP section
  - Supported starting NX-OS 10.2(3)F

Complete Your Session Evaluations

Complete a minimum of 4 session surveys and the Overall Event Survey to be entered in a drawing to **win 1 of 5 full conference passes** to Cisco Live 2025.

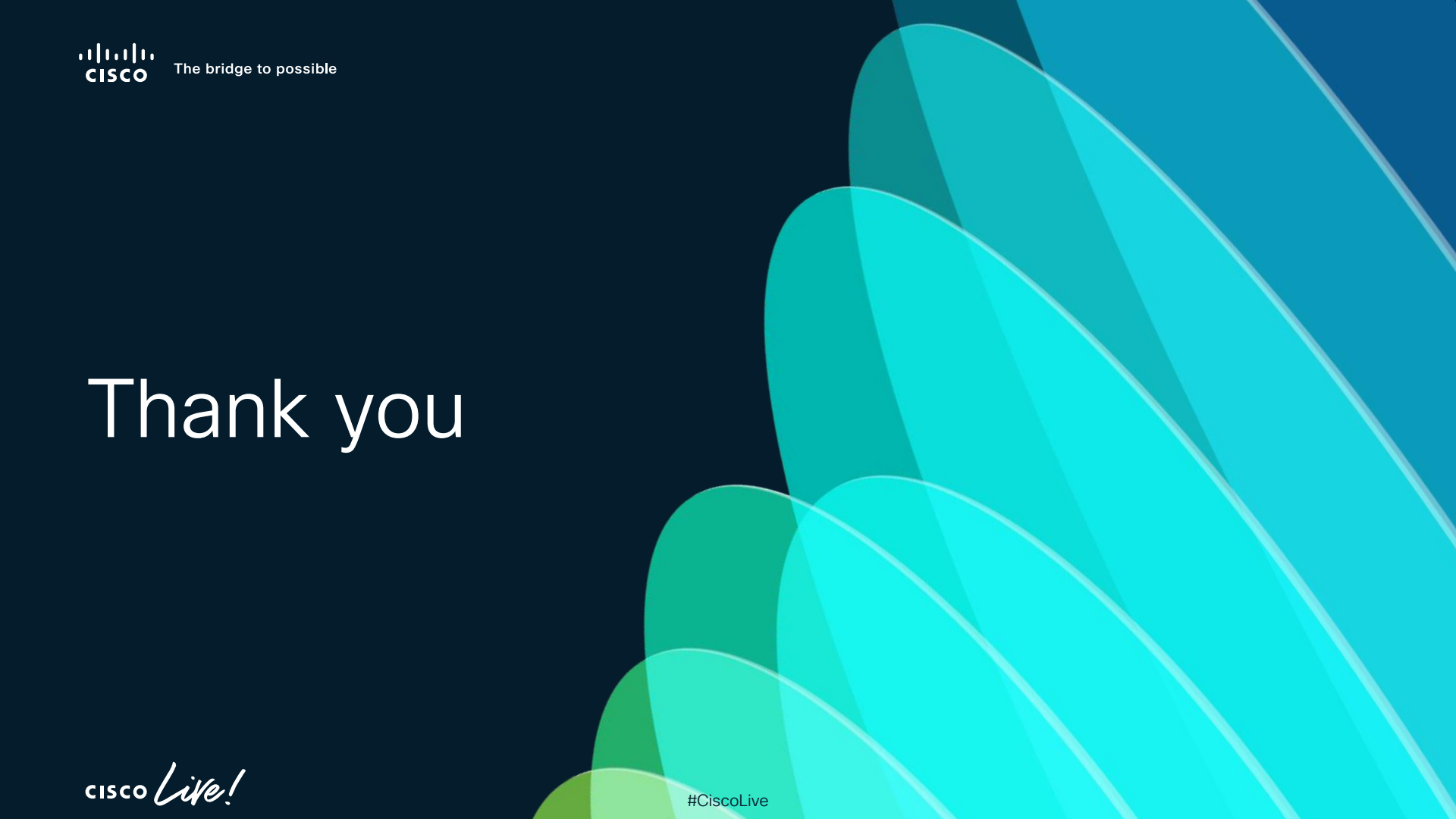
Earn 100 points per survey completed and compete on the Cisco Live Challenge leaderboard.

Level up and earn **exclusive prizes!**

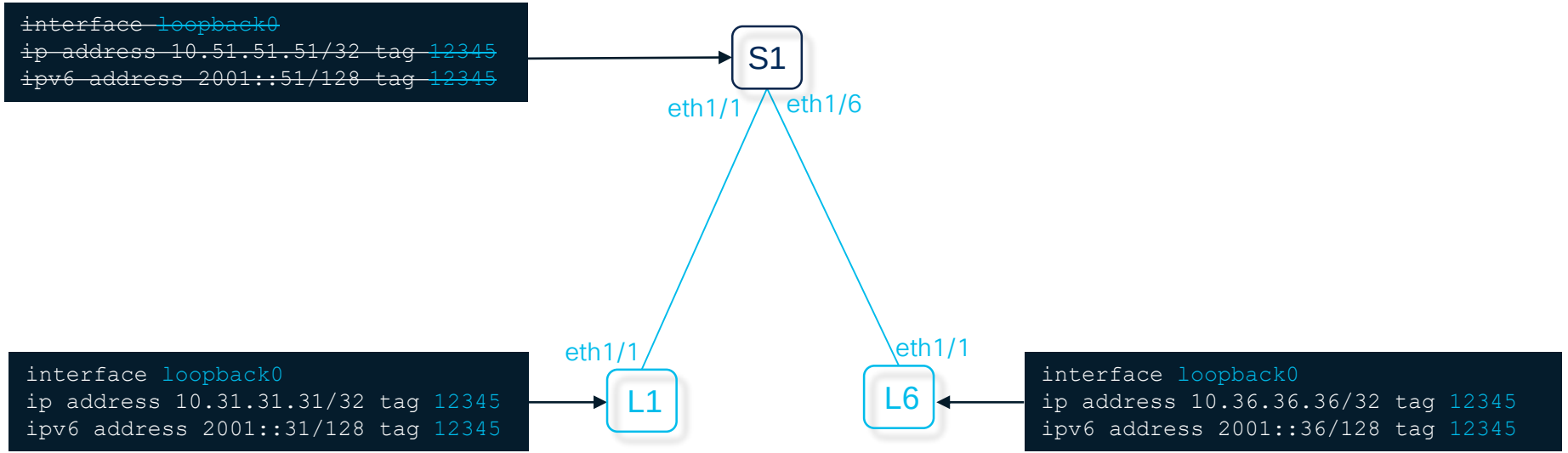
Complete your surveys in the **Cisco Live mobile app.**

Continue your education

- Visit the Cisco Showcase for related demos
- Book your one-on-one Meet the Engineer meeting
- Attend the interactive education with DevNet, Capture the Flag, and Walk-in Labs
- Visit the On-Demand Library for more sessions at www.CiscoLive.com/on-demand


The bridge to possible
Thank you
cisco Live!
#CiscoLive

Traceroute: Using Loopbacks



Traceroute: Using Loopbacks


```
leaf-1# traceroute 10.36.36.36 source 10.31.31.31
traceroute to 10.36.36.36 (10.36.36.36) from 10.31.31.31 (10.31.31.31), 30 hops max, 48 byte packets
 1   2.402 ms  1.704 ms  1.354 ms
 2  10.36.36.36 (10.36.36.36) (AS 65006)  3.376 ms  2.892 ms  2.615 ms
leaf-1#
```

IPv4 Traceroute with No Loopback on Spine

Something is Missing!?

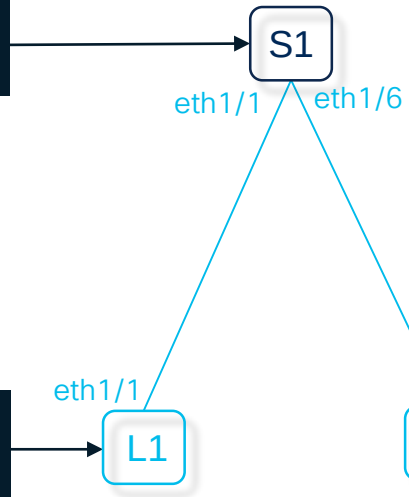
```
leaf-1# traceroute6 2001::36 source 2001::31
traceroute to 2001::36 (2001::36) from 2001::31, 30 hops max, 24 byte packets
 1  fe80::5016:9cff:fe03:101  16.323 ms  3.117 ms  2.351 ms
 2  2001::36  22.947 ms  4.213 ms  3.428 ms
leaf-1#
```

IPv6 Traceroute with No Loopback on Spine

Complete, but Difficult to Read

Traceroute: Using Loopbacks

```
interface loopback0
ip address 10.51.51.51/32 tag 12345
ipv6 address 2001::51/128 tag 12345
```




```
interface loopback0
ip address 10.31.31.31/32 tag 12345
ipv6 address 2001::31/128 tag 12345
```

```
interface loopback0
ip address 10.36.36.36/32 tag 12345
ipv6 address 2001::36/128 tag 12345
```

Traceroute: Using Loopbacks

L1

```
leaf-1# traceroute 10.36.36.36 source 10.31.31.31
traceroute to 10.36.36.36 (10.36.36.36) from 10.31.31.31 (10.31.31.31), 30 hops max, 48 byte packets
 1  10.51.51.51 (10.51.51.51) (AS 65111)  2.474 ms  2.076 ms  1.504 ms
 2  10.36.36.36 (10.36.36.36) (AS 65006)  4.077 ms  3.797 ms  3.917 ms
leaf-1#
```

IPv4 Traceroute with IPv4 Loopback on Spine

```
leaf-1# traceroute6 2001::36 source 2001::31
traceroute to 2001::36 (2001::36) from 2001::31, 30 hops max, 24 byte packets
 1  2001::51  4.81 ms  2.703 ms  2.452 ms
 2  2001::36  4.051 ms  3.682 ms  4.701 ms
leaf-1#
```

IPv6 Traceroute with IPv6 Loopback on Spine

Complete and Easy to Read