



# Enabling Topology Visibility in SONiC with BGP-LS

Ahmed Abdelsalam (Cisco) & Yijiao Qin (Alibaba)

On behalf of Cisco and Alibaba teams



# AI DCI/WAN

- Why? Lack of power density

# AI DCI/WAN

- Why? Lack of power density

## Inside the world's most powerful AI datacenter

Sep 18, 2025 | [Scott Guthrie - Executive Vice President, Cloud + AI](#)

### **AI WAN: Connecting multiple datacenters for an even larger AI supercomputer**

These new AI datacenters are part of a global network of Azure AI datacenters, interconnected via our Wide Area Network (WAN). This isn't just about one building, it's about a distributed, resilient and scalable system that operates as a single, powerful AI machine. Our AI WAN is built with growth capabilities in AI-native bandwidth scales to enable large-scale distributed training across multiple, geographically diverse Azure regions, thus allowing customers to harness the power of a giant AI supercomputer.

This is a fundamental shift in how we think about AI supercomputers. Instead of being limited by the walls of a single facility, we're building a distributed system where compute, storage and networking resources are seamlessly pooled and orchestrated across datacenter regions. This means greater resiliency, scalability and flexibility for customers.

<https://blogs.microsoft.com/blog/2025/09/18/inside-the-worlds-most-powerful-ai-datacenter/>

# AI DCI/WAN

- Why? Lack of power density

## Inside the world's most powerful AI datacenter

Sep 18, 2025 | [Scott Guthrie - Executive Vice President, Cloud + AI](#)

### AI WAN: Connecting multiple datacenters for an even larger AI supercomputer

These new AI datacenters are part of a global network of Azure AI datacenters, interconnected via our Wide Area Network (WAN). This isn't just about one building, it's about a distributed, resilient and scalable system that operates as a single, powerful AI machine. Our AI WAN is built with growth capabilities in AI-native bandwidth scales to enable large-scale distributed training across multiple, geographically diverse Azure regions, thus allowing customers to harness the power of a giant AI supercomputer.

This is a fundamental shift in how we think about AI supercomputers. Instead of being limited by the walls of a single facility, we're building a distributed system where compute, storage and networking resources are seamlessly pooled and orchestrated across datacenter regions. This means greater resiliency, scalability and flexibility for customers.

<https://blogs.microsoft.com/blog/2025/09/18/inside-the-worlds-most-powerful-ai-datacenter/>

## Introducing Virgo Network, Google's scale-out AI data center fabric

April 22, 2026

The AI era requires a fundamental rethink of physical cloud architecture — networking, in particular. With foundational model parameters growing exponentially, traditional general-purpose networks are reaching their breaking points. To fuel the next decade of machine learning, Google designed Virgo Network, a new megascale AI data center fabric that embraces a "campus-as-a-computer" philosophy, and that underpins our [AI Hypercomputer](#).

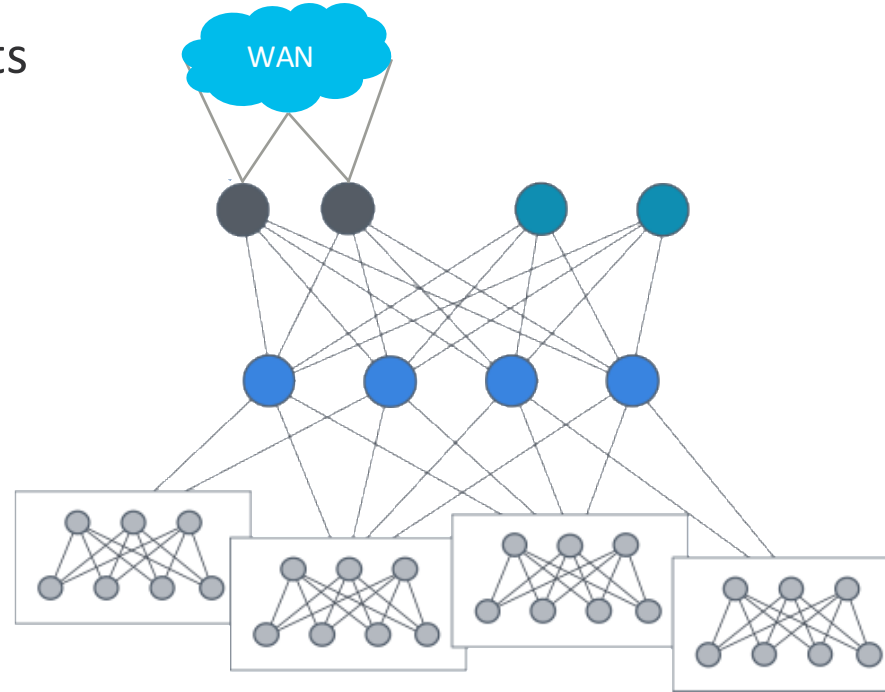
Legacy network designs simply cannot handle some of the constraints of modern AI:

1. **Massive scale:** Training demands now exceed the power and space of a single data center, requiring unified, multi-data-center domains.
2. **Explosive bandwidth growth:** Because foundational model training is heavily network-bound, the required bandwidth per accelerator has surged significantly over the last few years, creating throughput and congestion bottlenecks for older architectures.
3. **Synchronized bursts:** Intense, millisecond-level traffic spikes (figure 1) put immense pressure on network buffers. The outcome is that even a single "straggler" node can throttle the entire cluster's performance.
4. **Low latency:** ML serving requires fast, consistent response times to deliver real-time inference, making strict latency control a critical architectural constraint.

<https://cloud.google.com/blog/products/networking/introducing-virgo-megascale-data-center-fabric>

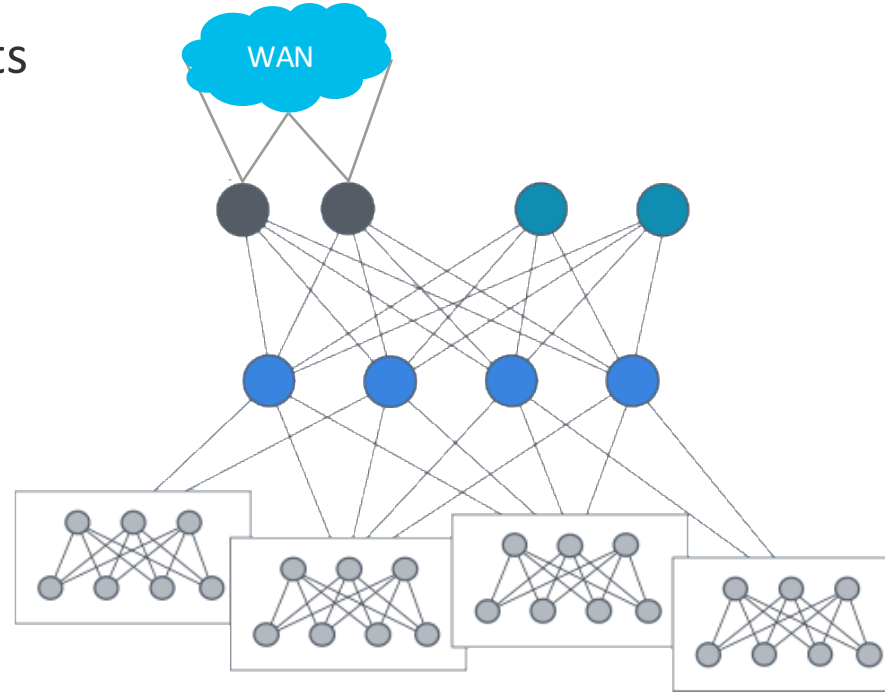
# AI DCI/WAN

- Different traffic patterns/requirements
- The topology is less symmetric
- The capacity is more scarce
- Latency matters more
- Weirdness:
  - SRLG's
  - Shortcuts between clusters



# AI DCI/WAN

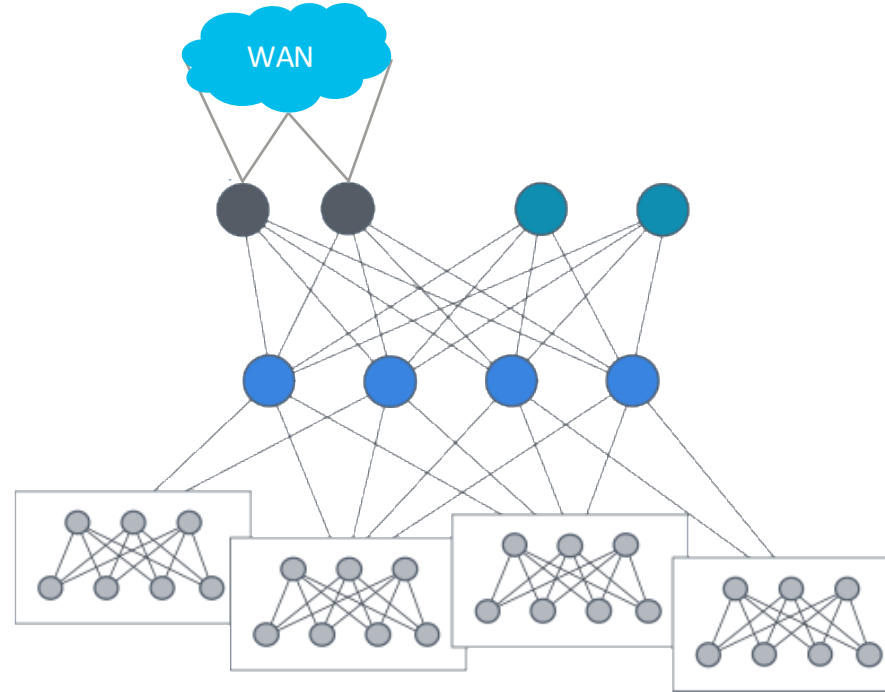
- Different traffic patterns/requirements
- The topology is less symmetric
- The capacity is more scarce
- Latency matters more
- Weirdness:
  - SRLG's
  - Shortcuts between clusters



Traffic Engineering and IP Fast Reroute – MUST!!

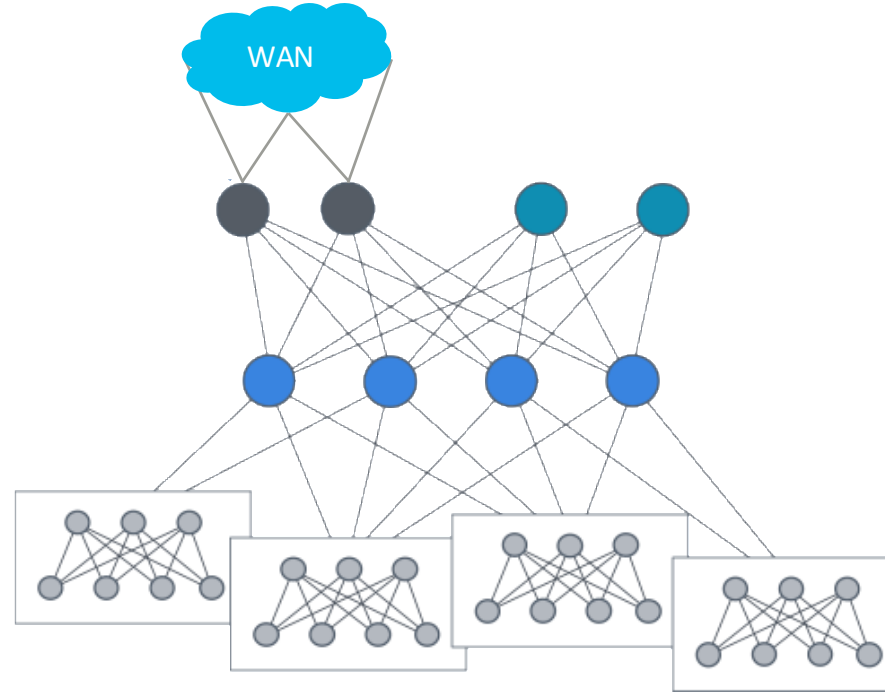
# AI DCI/WAN – Traffic Engineering and IP Fast Reroute

- TE path computation
  - Low latency, disjointness, BW
- IP Fast Reroute
  - LFA, Ti-LFA, uloop avoidance



# AI DCI/WAN – Traffic Engineering and IP Fast Reroute

- TE path computation
  - Low latency, disjointness, BW
- IP Fast Reroute
  - LFA, Ti-LFA, uloop avoidance



Topology View – Essential!!

# Topology

- Problem:
  - Existing approaches: *non-standard, fragmented, static, and difficult to scale across heterogeneous environments*
  - SONiC eBGP Fabric: each node has a visibility only of its direct neighbors

# Topology

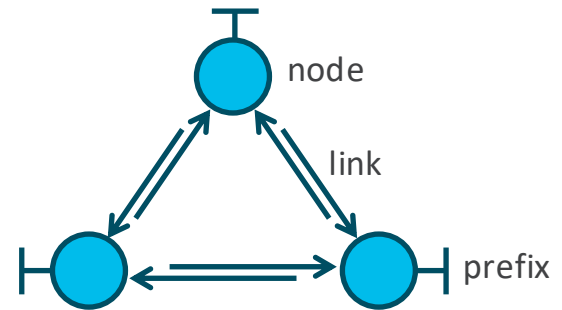
- Problem:
  - Existing approaches: *non-standard, fragmented, static, and difficult to scale across heterogeneous environments*
  - SONiC eBGP Fabric: each node has a visibility only of its direct neighbors
- Goal:
  - Standard, Open, Multi-vendor, and Scalable way.

# Topology

- Problem:
  - Existing approaches: *non-standard, fragmented, static, and difficult to scale across heterogeneous environments*
  - SONiC eBGP Fabric: each node has a visibility only of its direct neighbors
- Goal:
  - Standard, Open, Multi-vendor, and Scalable way.
  - **Unified solution for IGP and eBGP Fabric**

# BGP-LS: the right tool

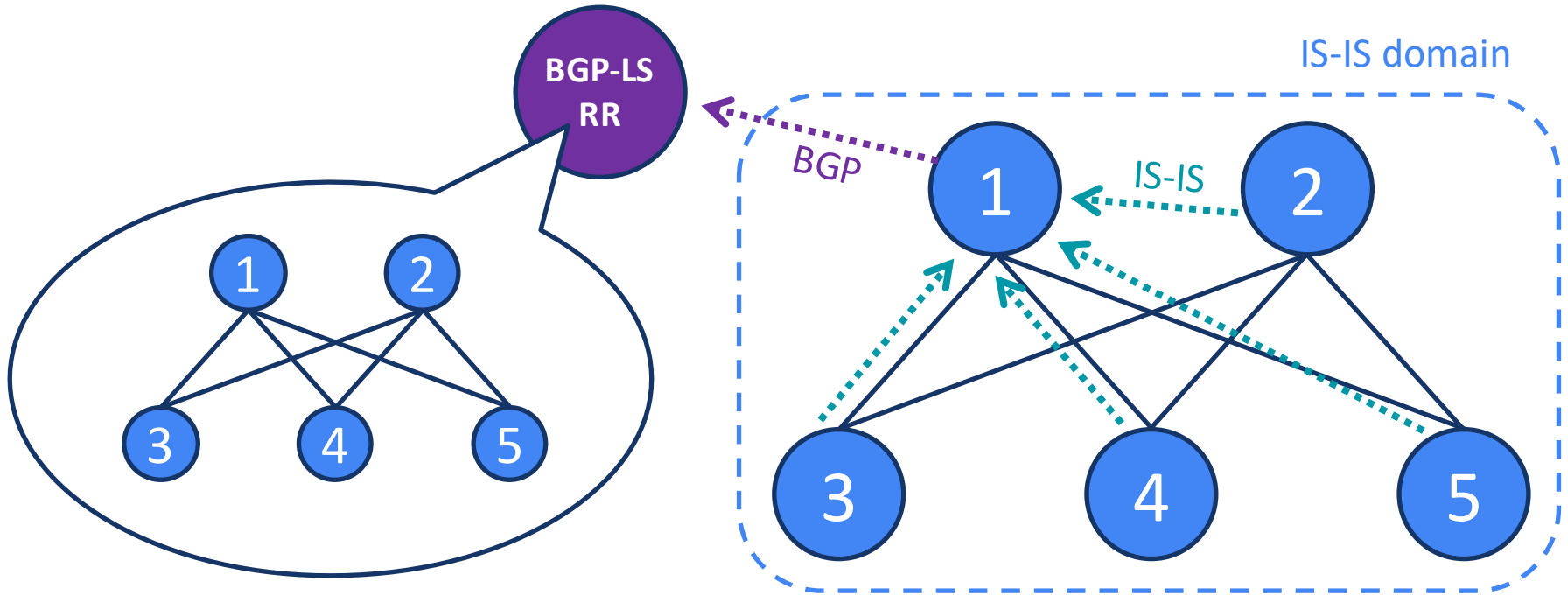
- Initially only for LS-IGPs...today much more...
- IETF RFC 9552 – a BGP extension to:
  - **Collect** topology info from the network protocol (IGP, BGP)
  - **Normalize** protocol information into a single topology model
  - **Distribute** it to any app or process that requires the topology
- BGP-LS topology model, three base object types:
  - **Node** – router, switch, protocol instance
  - **Link** – directed link anchored to two nodes
  - **Prefix** – anchored to a node
- Different BGP NLRI type per object-type
- More object/NLRI types exist (SID, SR Policy, ...)



# Why BGP-LS?

- **Standard-based** – IETF RFC 9552
- **Multi-vendor** – Supported by majority of vendors
- **Open** – Supported by open-source stacks (e.g., FRR)
- **Scalable** – leverage mature BGP infrastructure (RRs, policies, filtering, update-groups)
- **Multi-domain** – consolidates topology across IGPs, BGP ASes
- **Rich** – carries topology graph, node and link attributes, SIDs, ...
- **Application-friendly** – one session to an RR yields the full topology graph
- **Control-plane only** – no data-plane impact

# BGP-LS in IGP network (IS-IS) – deployment model



# BGP-LS in IGP (IS-IS) – FRR support

- Available in FRR mainline - <https://github.com/FRRouting/frr/pull/20470>

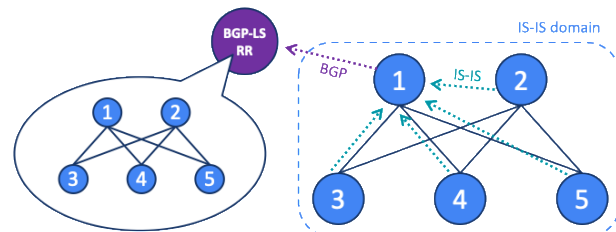
## RR

```
router bgp 65000
 neighbor 10.0.1.1 remote-as 65000
 !
 address-family link-state link-state
  neighbor 10.0.1.1 activate
 exit-address-family
```

## R1

```
router bgp 65001
 neighbor 10.0.0.1 remote-as 65000
 !
 address-family link-state link-state
  neighbor 10.0.0.1 activate
 exit-address-family

router isis SONIC
 net 49.0001.0000.0000.0001.00
 distribute link-state
 !
```



# BGP-LS Topology – NLRI's & Attributes

## Node NLRI:

Protocol-ID:  
Identifier:  
Local Node:  
AS:  
BGP-LS ID:  
Router-ID:  
Area-ID:

## Attributes:

Name:  
Flags:  
SRGB:  
SRLB:  
MSD:  
IPv6-ID:

## Link NLRI:

Protocol-ID:  
Identifier:  
Local Node:  
AS:  
BGP-LS ID:  
Router-ID:  
Area-ID:  
Remote Node:  
AS:  
BGP-LS ID:  
Router-ID:  
Area-ID:  
Link Descriptor:  
Link Local ID:  
Link Remote ID:  
IPv4 Interface Address:  
IPv4 Neighbor Address:  
IPv6 Interface Address:  
IPv6 Neighbor Address:

## Attributes:

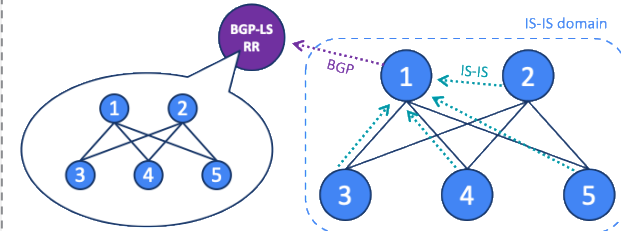
Metric:  
TE Metric:  
SRLG:  
IGP Flags:  
Delay:  
.....

## Prefix NLRI:

Protocol-ID:  
Identifier:  
Local Node:  
AS:  
BGP-LS ID:  
Router-ID:  
Area-ID:  
Prefix Descriptor:  
IP Reachability Info:  
Prefix Length:  
OSPF Route Type:  
IGP Flags:

## Attributes:

Metric:  
Prefix Metric:  
OSPF Forwarding Address:  
Route Tag:  
Extended Route Tag:  
Prefix SID:  
SR Algorithm:  
SR Local Block:  
SR Mapping Server:  
IGP Flags:  
Route Type:



# BGP-LS – FRR Show CLI

- Display all NLRIs

```
show bgp link-state link-state
```

- Show a specific Node NLRI

```
show bgp link-state link-state [NODE NLRI]
```

- Show a specific Link NLRI

```
show bgp link-state link-state [LINK NLRI]
```

- Show a specific Prefix NLRI

```
show bgp link-state link-state [Prefix NLRI]
```

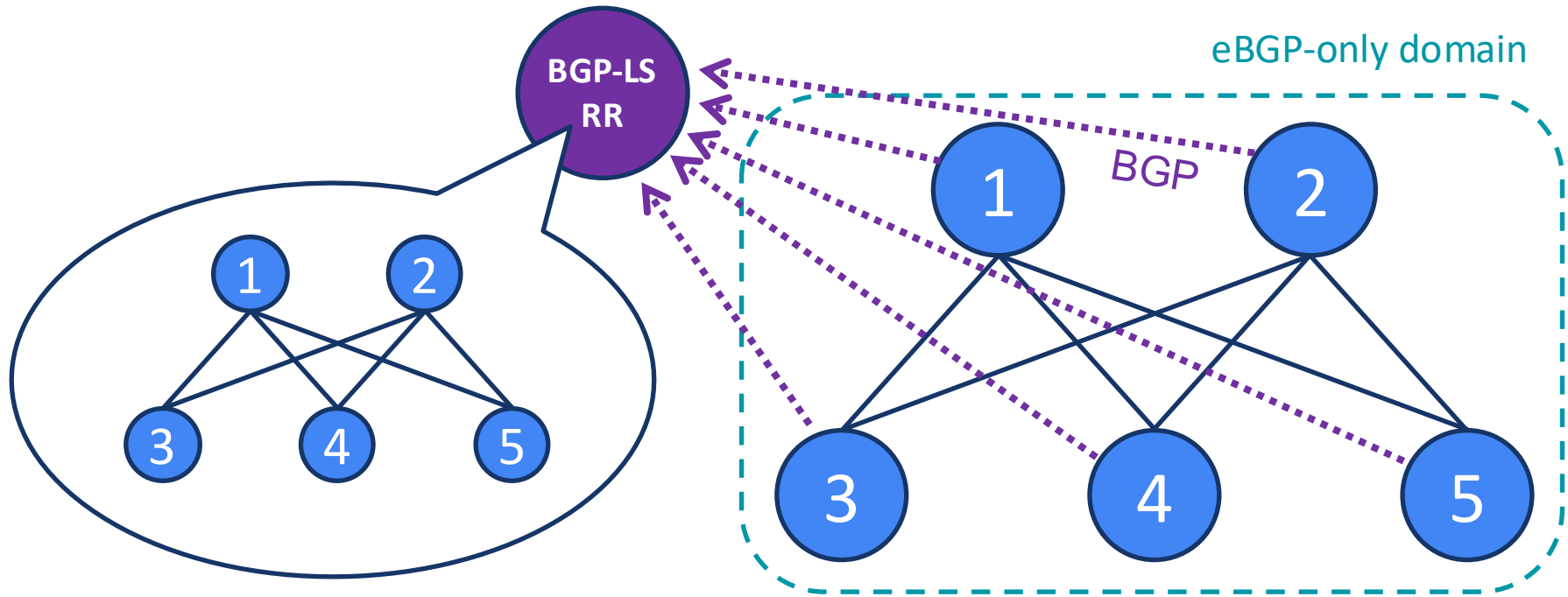
# SONiC eBGP Fabric

- eBGP is the norm in both DC and DCI
- Data Center Fabrics
  - eBGP-only Clos designs
  - No IGP required — BGP handles both underlay and overlay reachability
- Data Center Interconnect
  - eBGP extends naturally beyond the fabric boundary, across DC and DCI domains
  - Simplified operations: one protocol, one toolchain

# BGP-LS for eBGP

- No need for IGP
  - BGP-LS exports topology of eBGP fabrics
- Same BGP-LS topology model objects used in IGP
  - Node, Link, Prefix, SID, SR Policy, ....
- Standard
  - draft-ietf-idr-bgp-ls-bgp-only-fabric
  - specifies how to encode and advertise an eBGP-only network topology in BGP-LS
- Supported in FRR mainline
- Fully compatible with existing BGP-LS consumers
  - No changes to BGP-LS RR, Controller, or Apps

# BGP-LS for eBGP – deployment model



# BGP-LS for eBGP – FRR support

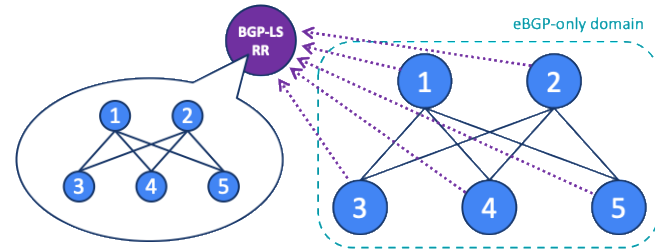
- Available in FRR mainline - <https://github.com/FRRouting/frr/pull/20726>

## RR

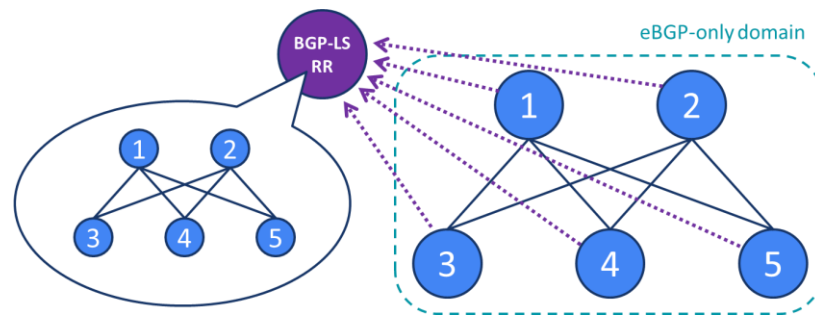
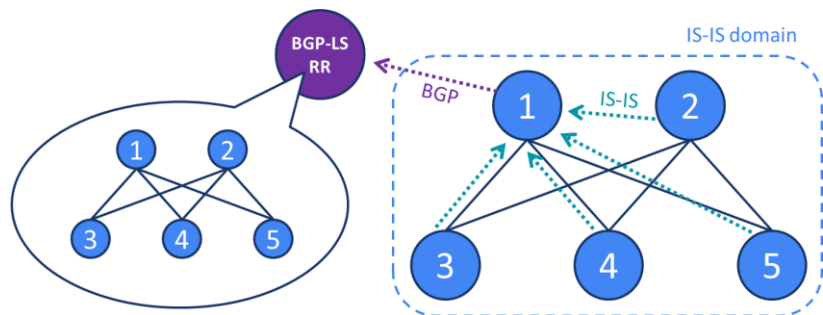
```
router bgp 65000
 neighbor 10.0.1.1 remote-as 65000
 neighbor 10.0.2.1 remote-as 65000
 neighbor 10.0.3.1 remote-as 65000
 neighbor 10.0.4.1 remote-as 65000
 neighbor 10.0.5.1 remote-as 65000
!
address-family link-state link-state
 neighbor 10.0.1.1 activate
 neighbor 10.0.2.1 activate
 neighbor 10.0.3.1 activate
 neighbor 10.0.4.1 activate
 neighbor 10.0.5.1 activate
exit-address-family
```

## R1 – R5

```
router bgp 65000
 neighbor 10.0.0.1 remote-as 65000
!
address-family link-state link-state
 distribute bgp-fabric-link-state
 neighbor 10.0.0.1 activate
exit-address-family
```



# BGP-LS - Unified solution for topology view in IGP & BGP Fabrics



**Node NLRI:**  
 Protocol-ID:  
 Identifier:  
 Local Node:  
 AS:  
 BGP-LS ID:  
 Router-ID:  
 Area-ID:

**Attributes:**  
 Name:  
 Flags:  
 SRGB:  
 SRLB:  
 MSD:  
 IPv6-ID:

**Link NLRI:**  
 Protocol-ID:  
 Identifier:  
 Local Node:  
 AS:  
 BGP-LS ID:  
 Router-ID:  
 Area-ID:  
 Remote Node:  
 AS:  
 BGP-LS ID:  
 Router-ID:  
 Area-ID:  
 Link Descriptor:  
 Link Local ID:  
 Link Remote ID:  
 IPv4 Interface Address:  
 IPv4 Neighbor Address:  
 IPv6 Interface Address:  
 IPv6 Neighbor Address:

**Attributes:**  
 Metric:  
 TE Metric:  
 SRLG:  
 IGP Flags:  
 Delay:  
 .....

**Prefix NLRI:**  
 Protocol-ID:  
 Identifier:  
 Local Node:  
 AS:  
 BGP-LS ID:  
 Router-ID:  
 Area-ID:

Prefix Descriptor:  
 IP Reachability Info:  
 Prefix Length:  
 OSPF Route Type:  
 IGP Flags:

**Attributes:**  
 Metric:  
 Prefix Metric:  
 OSPF Forwarding Address:  
 Route Tag:  
 Extended Route Tag:  
 Prefix SID:  
 SR Algorithm:  
 SR Local Block:  
 SR Mapping Server:  
 IGP Flags:  
 Route Type:

**Node NLRI:**  
 Protocol-ID:  
 Identifier:  
 Local Node:  
 AS:  
 BGP-LS ID:  
 Router-ID:  
 Area-ID:

**Attributes:**  
 Name:  
 Flags:  
 SRGB:  
 SRLB:  
 MSD:  
 IPv6-ID:

**Link NLRI:**  
 Protocol-ID:  
 Identifier:  
 Local Node:  
 AS:  
 BGP-LS ID:  
 Router-ID:  
 Area-ID:  
 Remote Node:  
 AS:  
 BGP-LS ID:  
 Router-ID:  
 Area-ID:  
 Link Descriptor:  
 Link Local ID:  
 Link Remote ID:  
 IPv4 Interface Address:  
 IPv4 Neighbor Address:  
 IPv6 Interface Address:  
 IPv6 Neighbor Address:

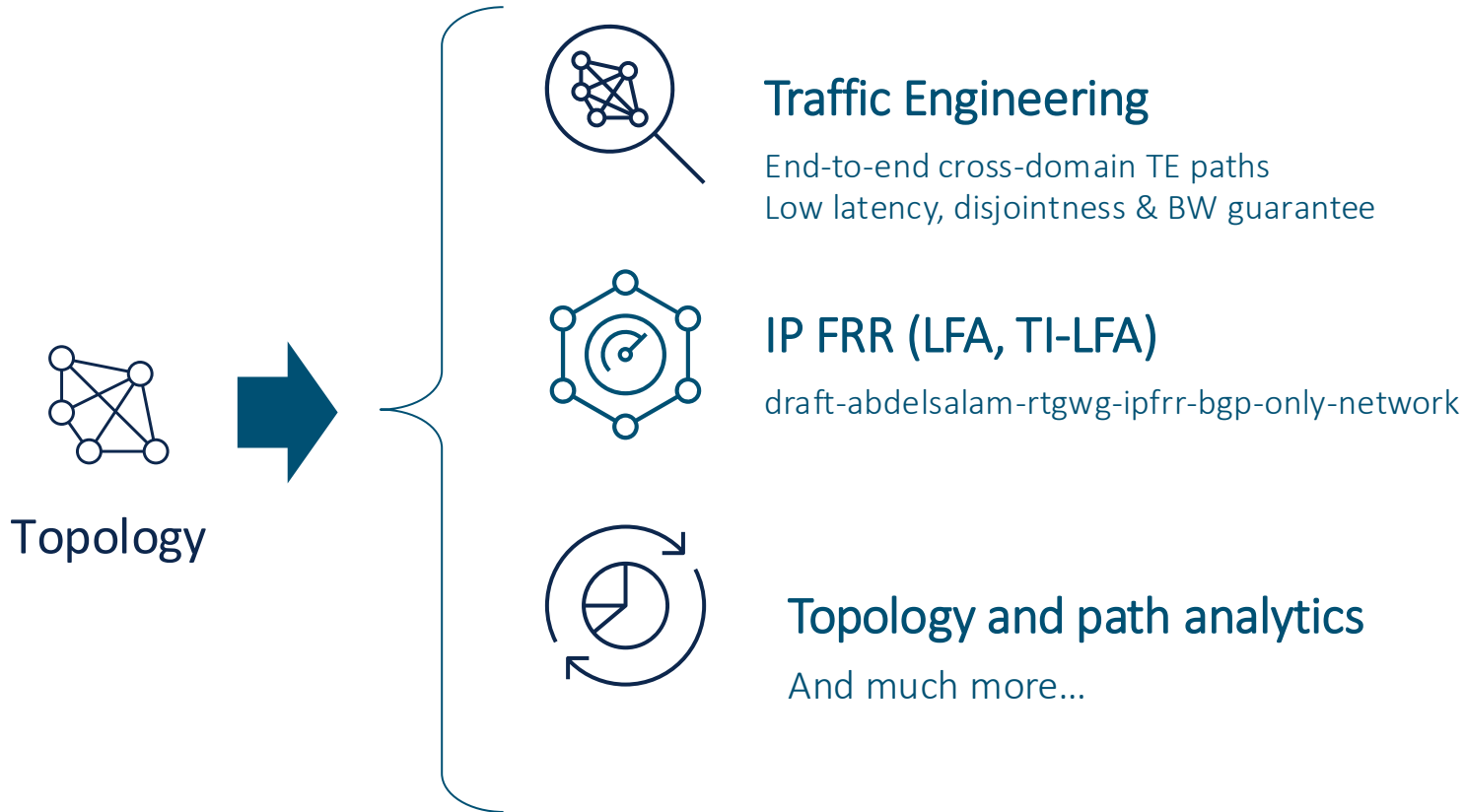
**Attributes:**  
 Metric:  
 TE Metric:  
 SRLG:  
 IGP Flags:  
 Delay:  
 .....

**Prefix NLRI:**  
 Protocol-ID:  
 Identifier:  
 Local Node:  
 AS:  
 BGP-LS ID:  
 Router-ID:  
 Area-ID:

Prefix Descriptor:  
 IP Reachability Info:  
 Prefix Length:  
 OSPF Route Type:  
 IGP Flags:

**Attributes:**  
 Metric:  
 Prefix Metric:  
 OSPF Forwarding Address:  
 Route Tag:  
 Extended Route Tag:  
 Prefix SID:  
 SR Algorithm:  
 SR Local Block:  
 SR Mapping Server:  
 IGP Flags:  
 Route Type:

# Foundational Building Block



# Conclusion

- AI DCI/WAN
  - Training demands exceed the power and space of a single datacenter
  - Connecting multiple datacenters for a larger AI supercomputer
- Traffic Engineering and IP Fast Reroute
  - Must in AI/DCI/WAN
  - Topology is an essential foundational block
- BGP-LS
  - Unified solution for topology view in IGP & BGP Fabrics
  - Standard, Multi-vendor, Open, Scalable, Multi-domain, Rich, Application-friendly.
  - Available in FRR mainline for both IGP & BGP Fabrics



Thank you!

