Multi-Path Traffic Engineering (MPTE) *for WAN & AIDC*

Kireeti Kompella, Jeffrey Zhang

4 February 2026, NANOG #96

Agenda

- To TE or not to TE
- Existing TE Toolkit
- Multipath Traffic Engineering (MPTE)
  - Adding Multipath to TE
  - Adding TE to Multipath
  - MPTE DAG (MPTED) – Key Concepts
- Takeaways and References

Network Optimization: Goals & Tools

Goals:

- High utilization, congestion-free, low latency/loss, resilient
- Same for WAN and AIDC – even more so for AIDC!
  - AI workloads are sensitive to congestion & latency. GPU pipelines stall when even a single packet is delayed or dropped.

Tools:

- Non-shortest path, bandwidth reservation, alternative path

Traditionally, TE with BW reservation is the de facto tool for achieving the goals in the WAN

- But it does not support multipath, and is not adopted in DCs

MPLS and TE

MPLS was introduced in 1998 with a goal of providing TE

- RFC 2702: Traffic Engineering over MPLS (UUnet) (1998).
- Productizing MPLS was Kireeti's first task in routing protocols.

Yes, a few things changed, but the fundamentals remain the same

- "Auto-bandwidth" is a big step forward, making BW specs easier & more useful
- "forwarding adjacencies" allow us to build hierarchical TE more easily

Today's networks offer fresh perspectives on the need for load balancing & TE

- … perhaps MPLS could bring something novel to the table and change how we engineer networks for AI workloads - MPTE for WAN & AIDC

Why Engineer Traffic?		Why Not?
To better utilize network capacity		Throw bandwidth at the problem!
To handle faults and network outages		Use **ECMP** and cross your fingers!
To improve (end) customer experience		They don't pay us differentially!
To get more insights into traffic patterns		Patterns? What patterns?

Primary concerns about TE:
Scaling & OPEX overhead of managing TE tunnels/paths/policies

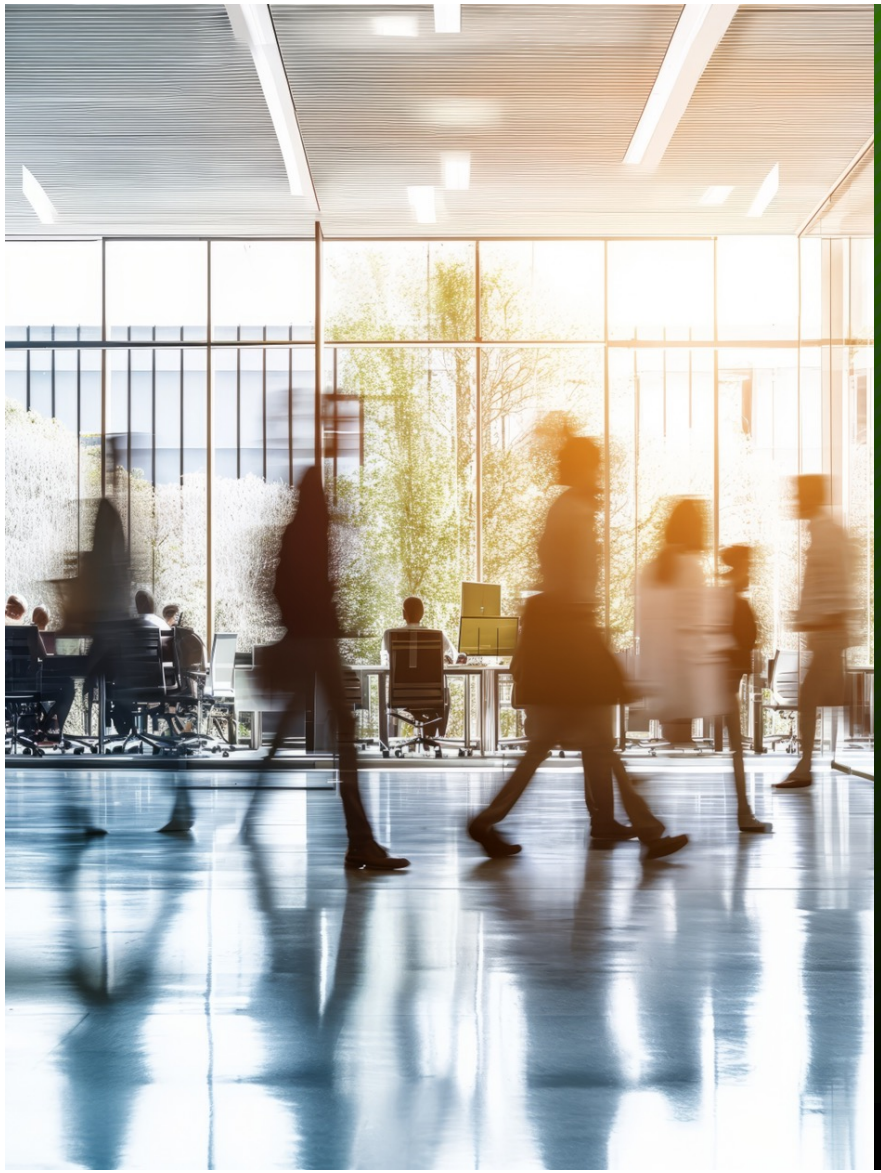
TE Considerations Today

Control/data planes handle scale & churn much better

Management and automation tools have vastly improved
Telemetry, analysis, insights are HUGE today necessary for AIOPs)

Network usage is changing → AI workloads benefit from TE as well

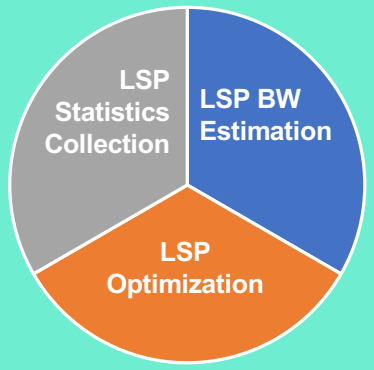
To *TE* or not to *TE*, that is ***no longer*** the question



Traffic Engineering Toolkit

Auto-Bandwidth Enabled LSPs

Auto-bandwidth automatically adjusts BW allocation based on real-time traffic



LSP Statistics Collection

Customized statistics polling profiles

LSP BW Estimation

cBPS (Containerized Bandwidth Prediction Service)

1st AI/ML enabled Auto BW solution

LSP Optimization

LSP Bandwidth Update, In-Place

- Difficulty in placing fat LSPs in the network (bin packing problems)
- Needs manual provisioning of additional LSPs between the same end-points based on traffic demand

Container LSPs [TE++]

Improved “bin packing” without the need for additional provisioning efforts

- Consists of multiple dynamically created member LSPs
- Allows for elastic sizing of member LSPs AND the creation/removal of member LSPs based on actual traffic patterns

Traffic LB only at the ingress node

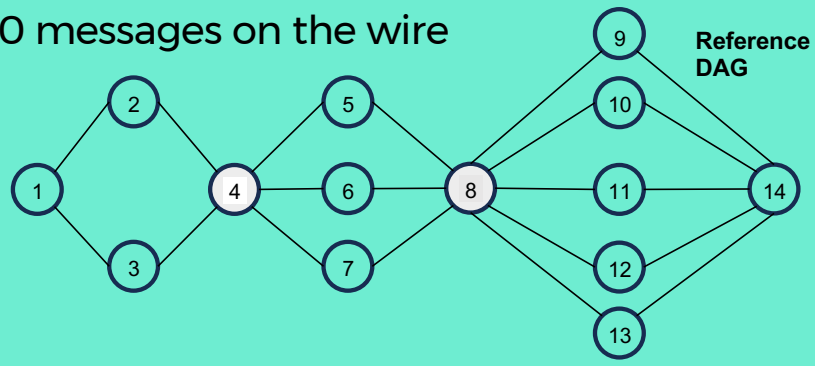
- Currently, supports only equally weighted load balancing

No intent to maximize multipath spread

Individually signaling/maintenance of member LSP state can be a deterrent in scaled deployments

Requires 30 member LSPs to cover all paths

360 messages on the wire



R4/R8 maintain 30 PSBs and 30 RSBs

Other Notable TE Tools

Optimizing Bypass LSP using Unreserved Bandwidth

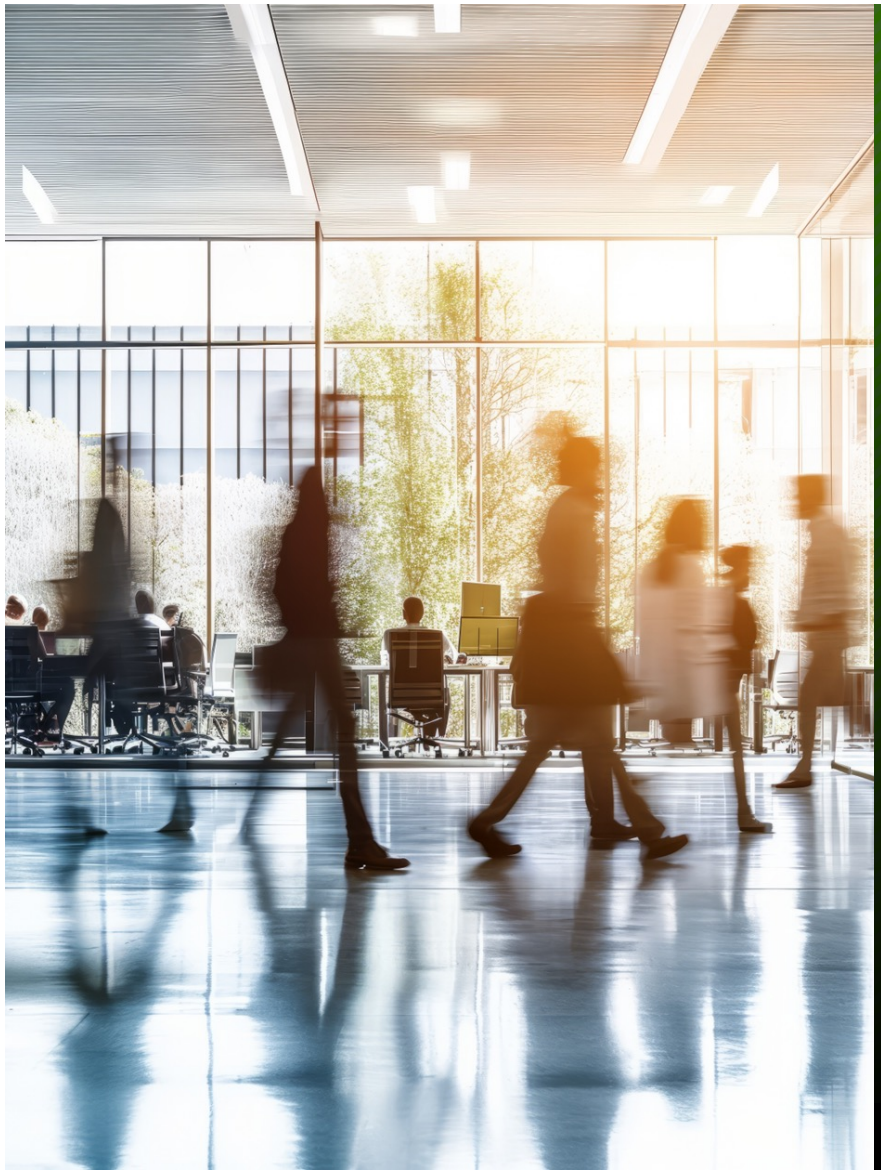
- Compute bypass paths with “unreserved TE link bandwidth” as the optimization objective
- With this functionality, the bypass path computation will be greedy for unreserved-bandwidth on a graph that is pruned of all TE links that do not satisfy any constraint

RSVP Tactical TE (TTE)*

- A congestion point triggers ingress routers to evaluate affected LSPs and take action

Lightweight DiffServ TE*

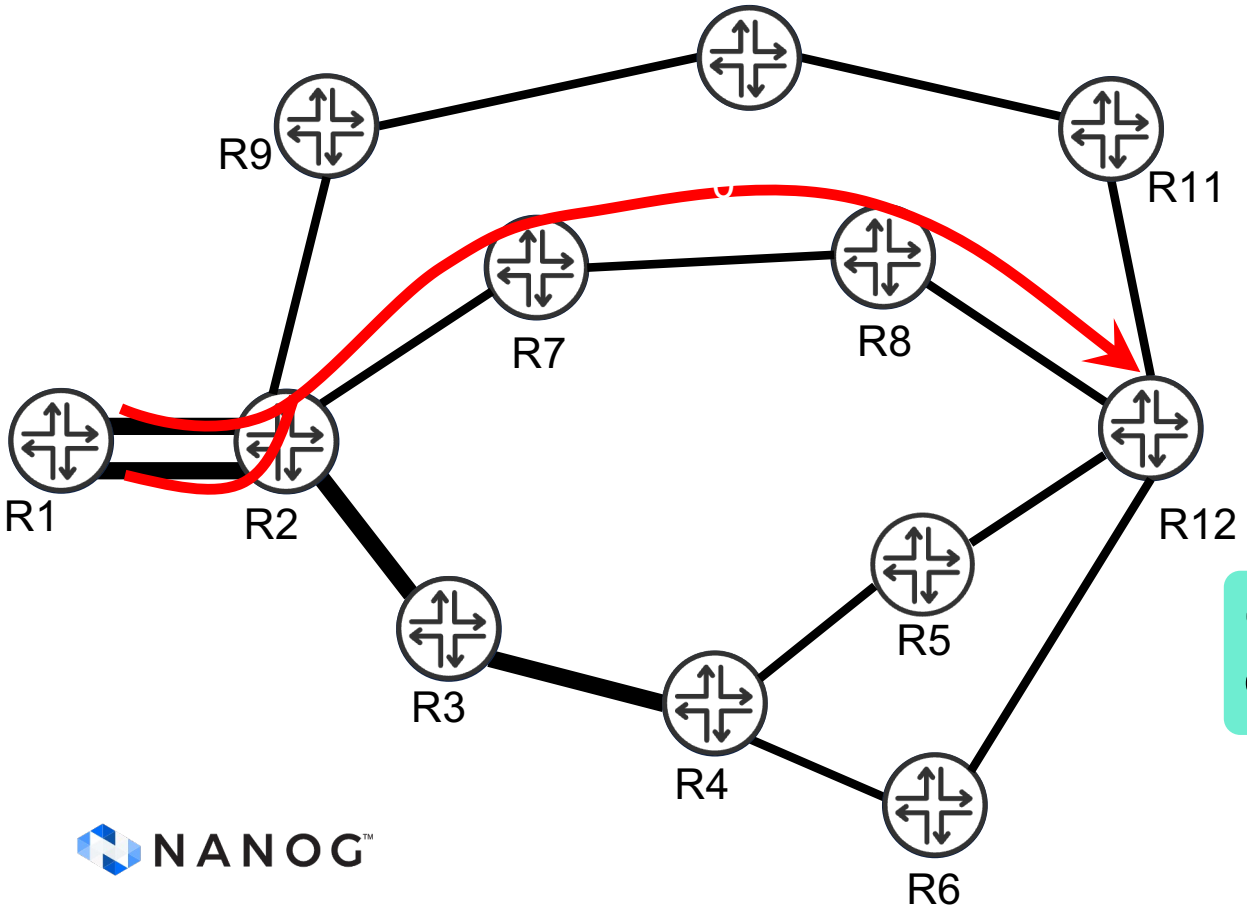
- Relaxed per priority subscription rules facilitate the creation of customized control-plane and forwarding-plane resource allocation policies with resource sharing



Multipath Traffic Engineering MPTE

- Adding Multipath to TE

Shortest Path Routing



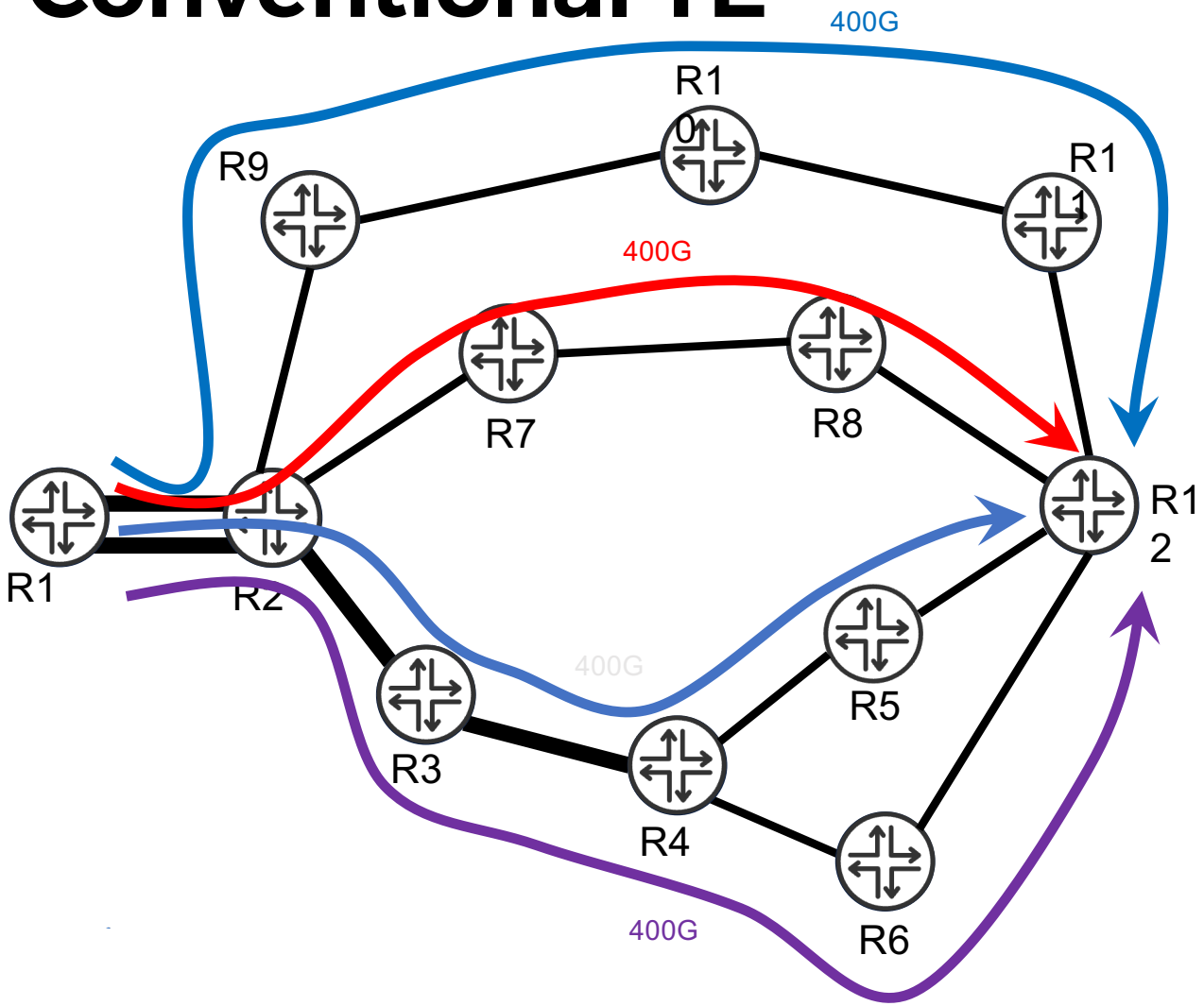
400G link

800G link

All link metrics are equal

Can only send a maximum of 400 Gbps from R1 to R12

Conventional TE



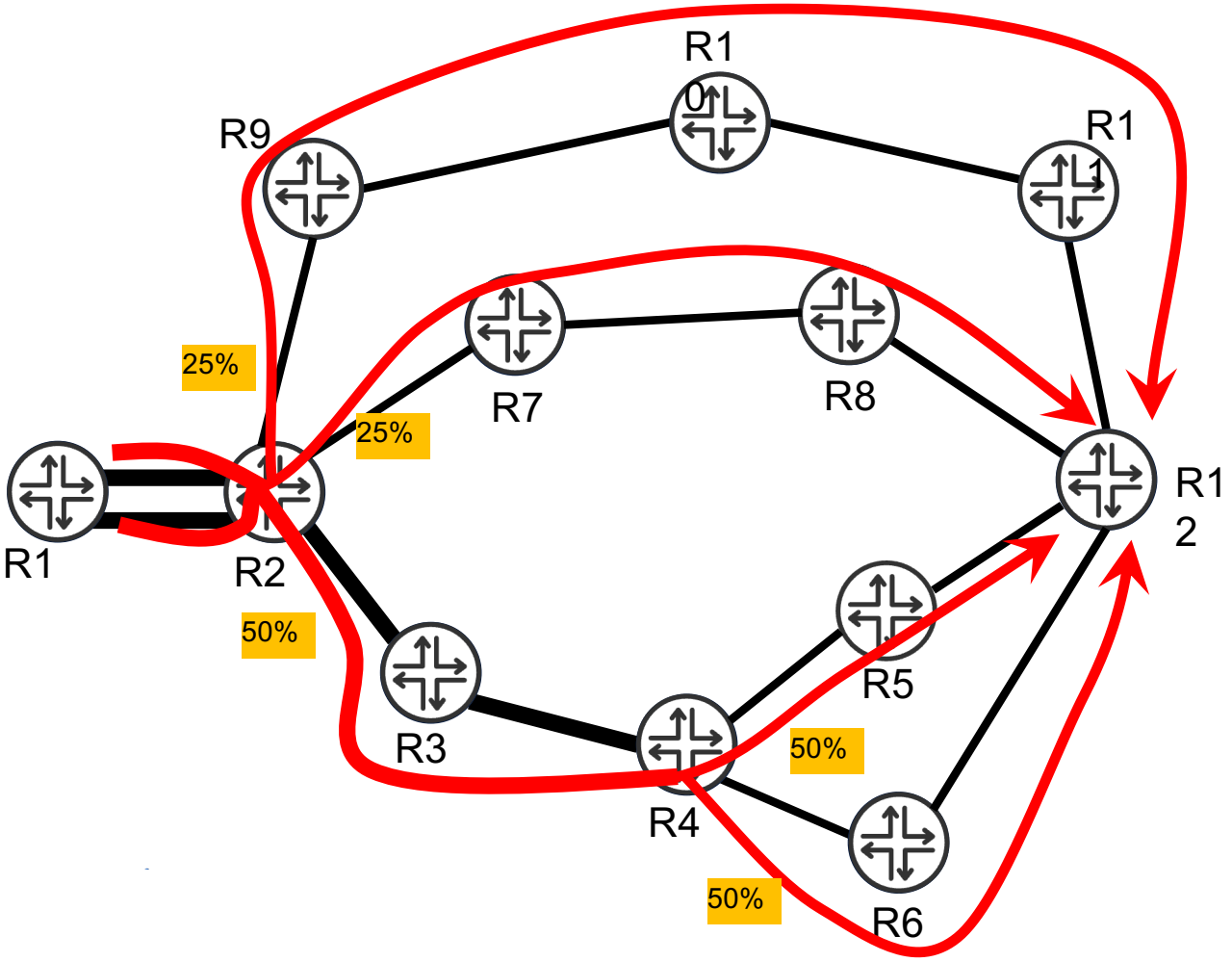
400G link

800G link

Can achieve 1600 Gbps throughput from R1 to R12.

But require four LSP constructs (created via TE++ or manually) to do so

Multipath TE (MPTE)



400G link

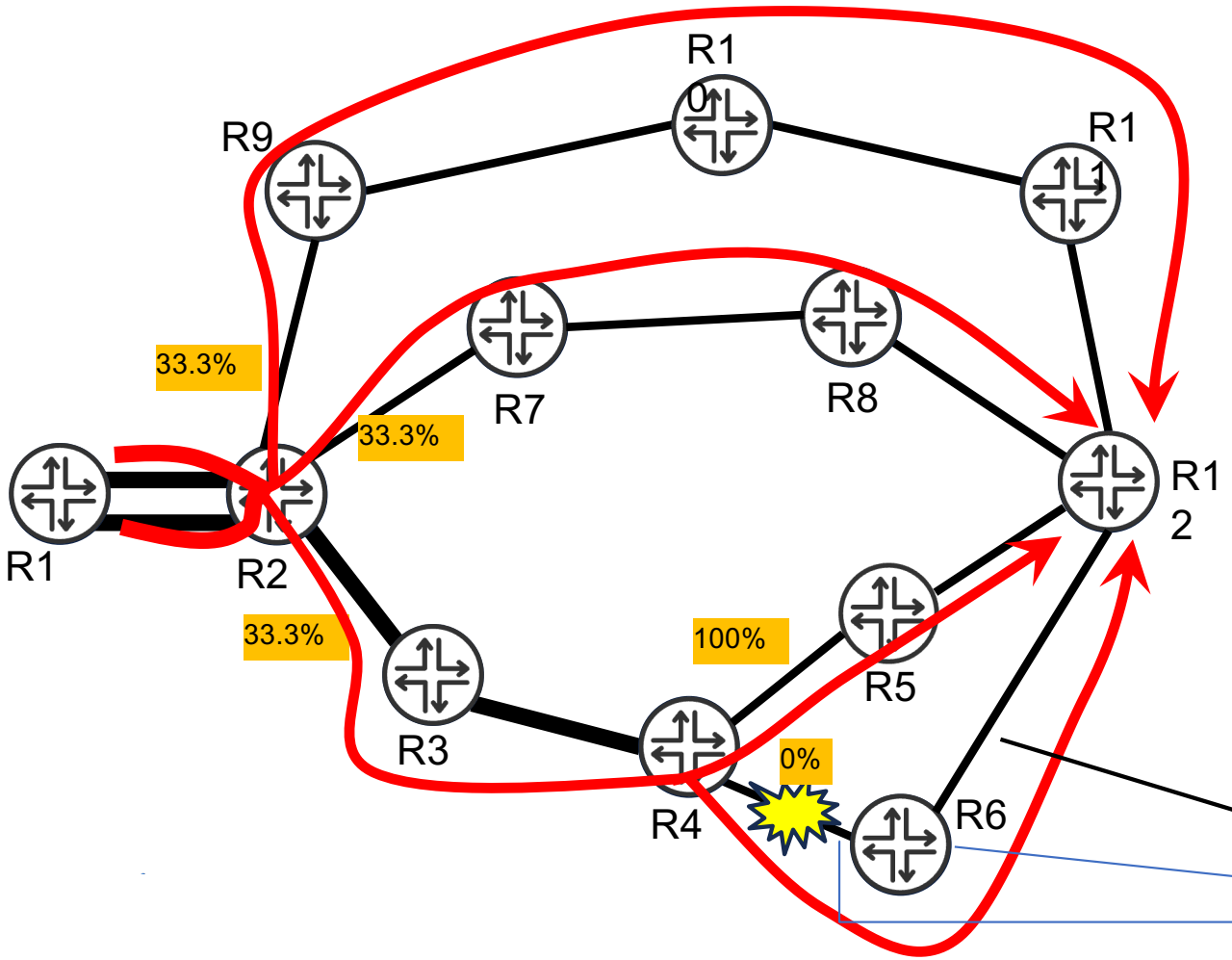
800G link

1600 Gbps throughput is achieved with just one MPTE DAG construct

MPTE DAG can branch (with an "optimal" split ratio at each junction) and recombine as needed

X% Local split percentage (load share) at a junction node

MPTE and Link Failure



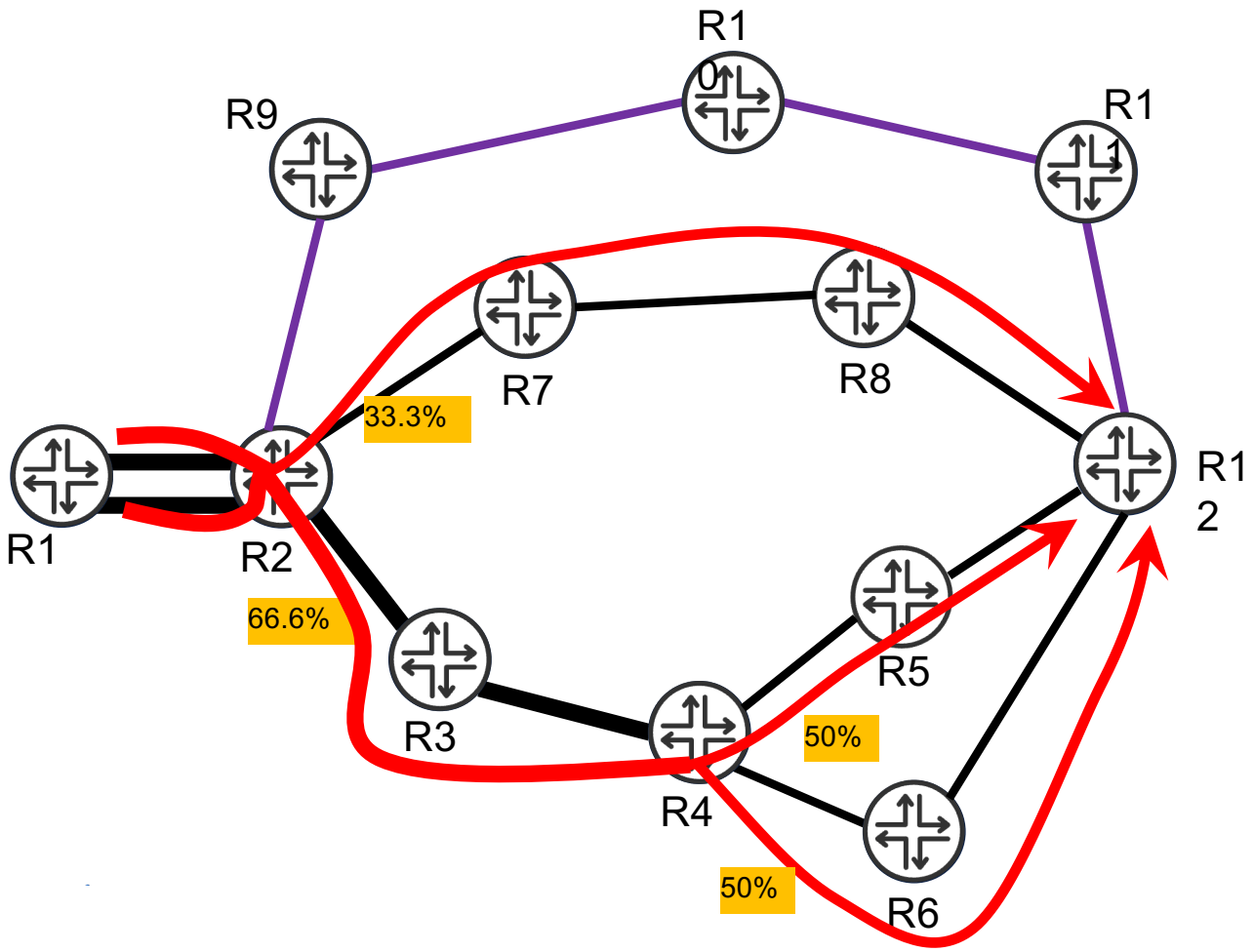
400G link

800G link

Automatic adjustment of the MPTE DAG and its branching ratios when a link on the DAG goes down temporarily.

Shape of the DAG remains unchanged unless there is a permanent change to the topology

MPTE with Constraints



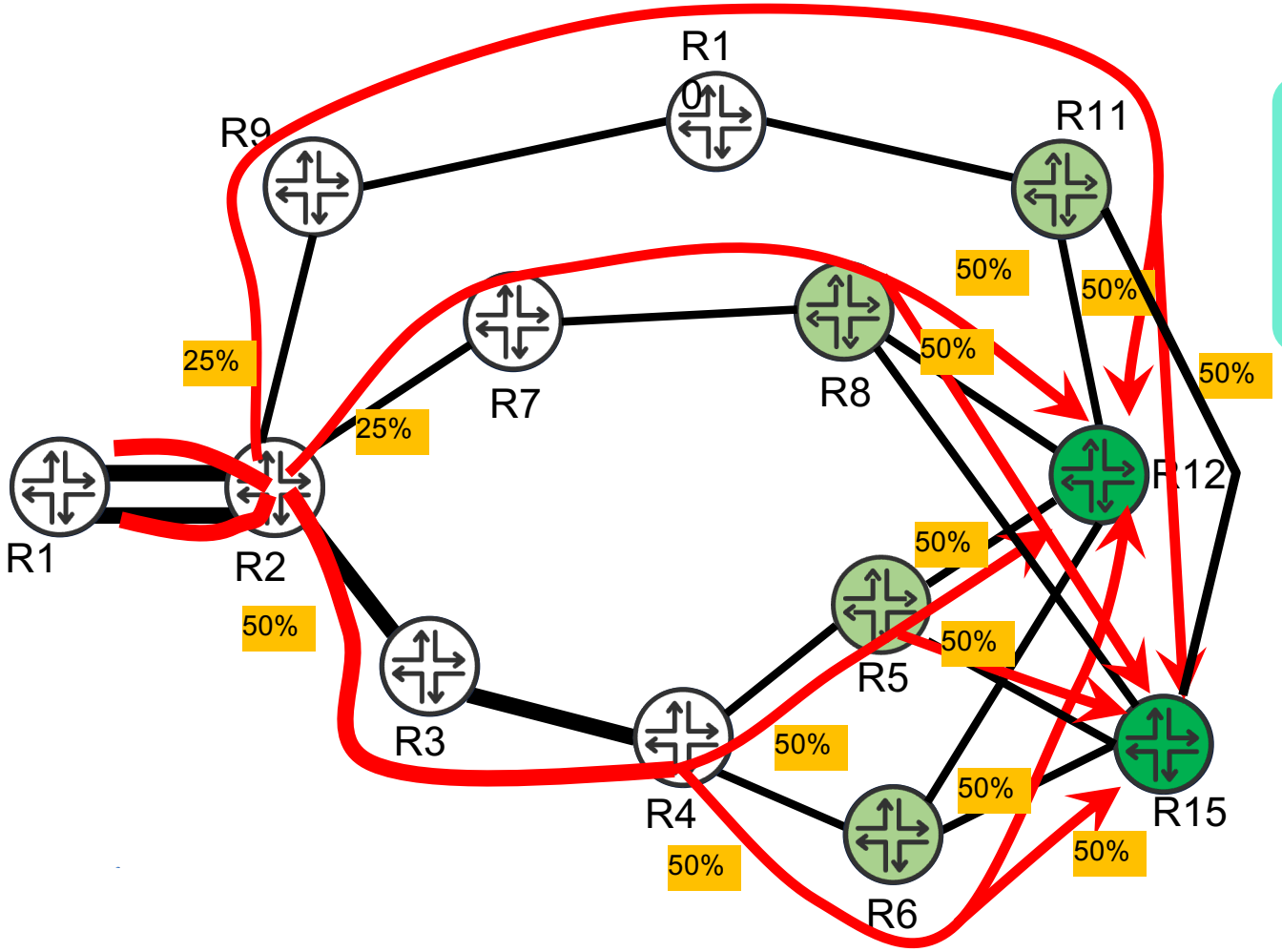
400G link

800G link

Can have constraints, just like with conventional TE

E.g. “exclude links with purple admin-group”

MPTE with Multiple Egresses



Can have multiple egress PEs for the same MPTE DAG!

Note: this is *not* a multicast LSP: traffic is load-balanced across the egress PEs.

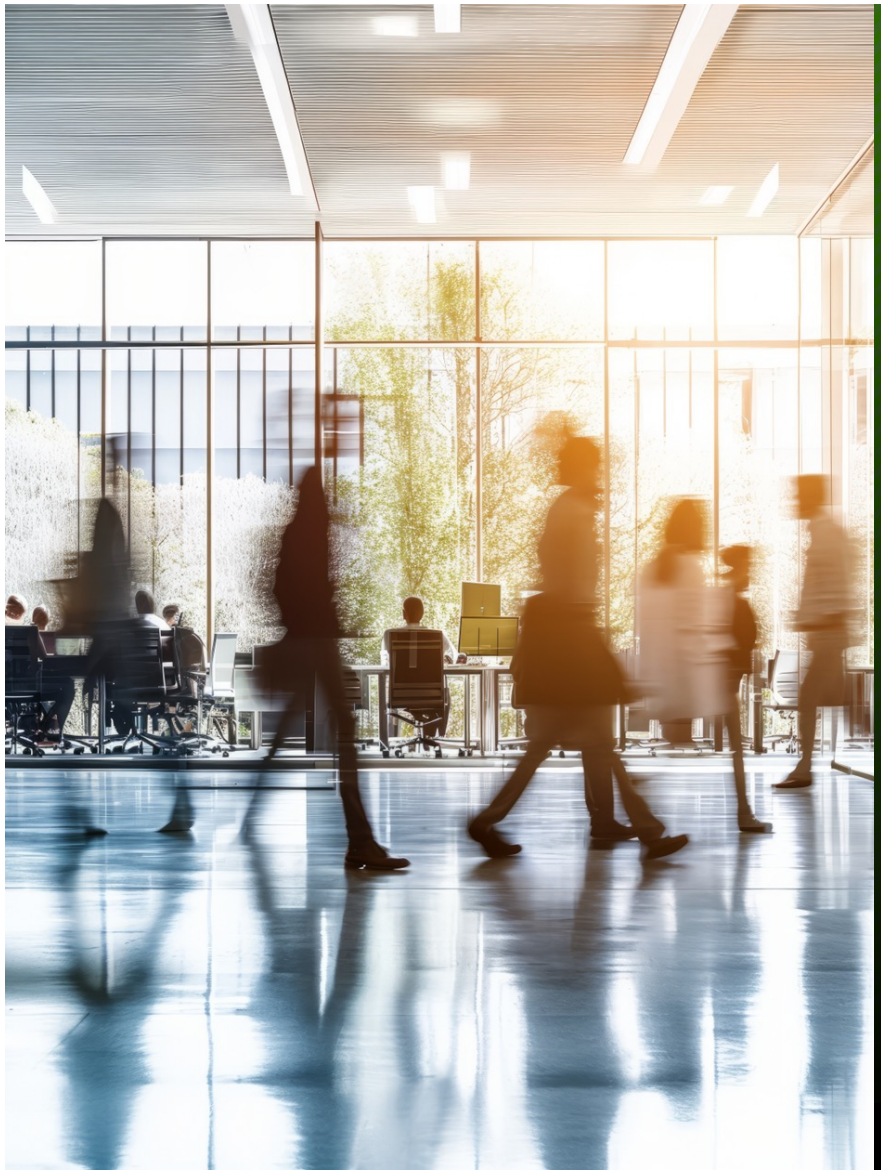
Suppose R15 is "as good as" R12 as an egress (e.g., iBGP multipath; VPN multihoming)

R15 can be accommodated by the MPTE DAG with relatively little extra state

(only R11, R8, R5 and R6 have updated state (more nhops); rest of DAG is unchanged)

MPTE – Key Differentiators / Attributes

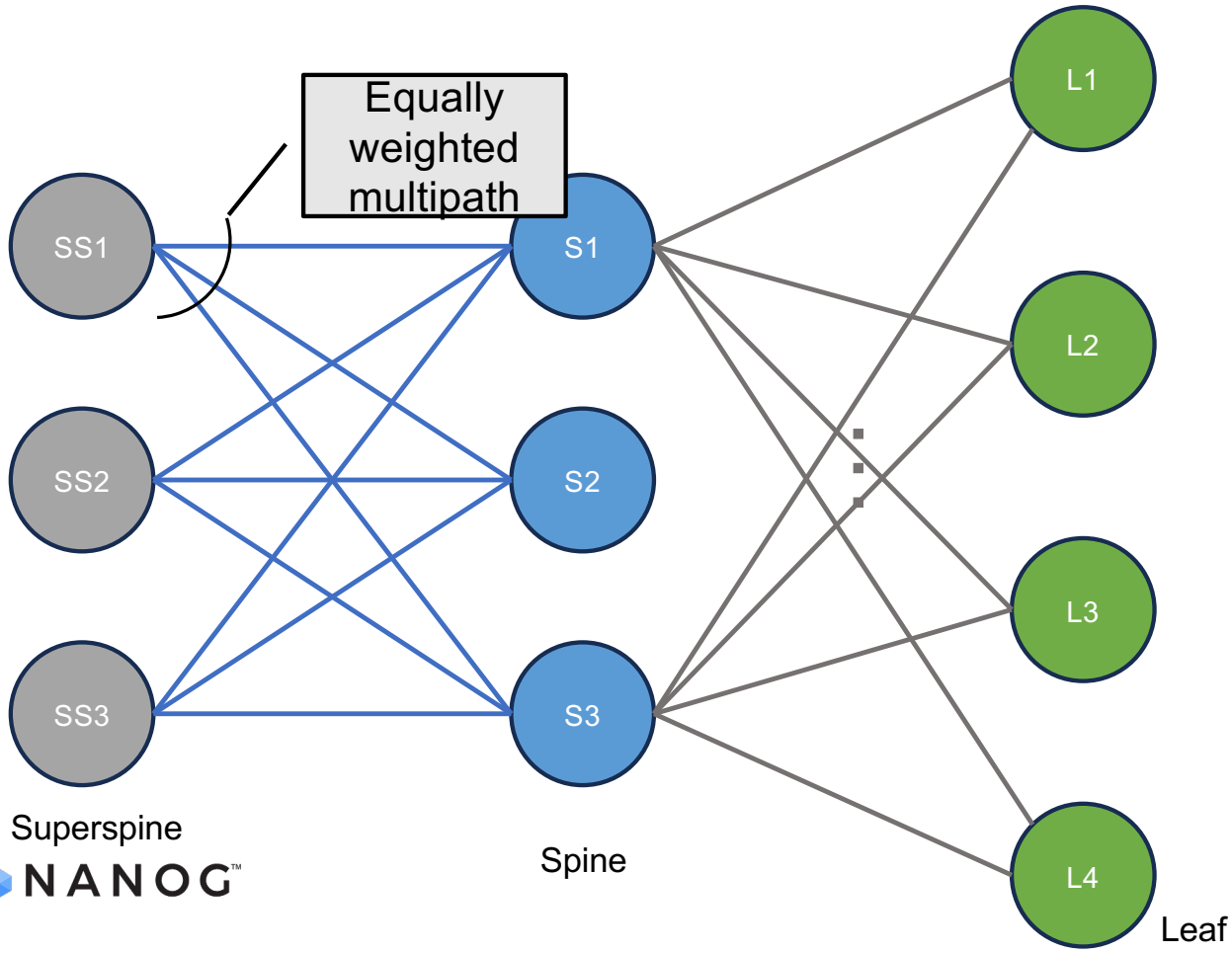
- Enables unequal-cost load balancing at every junction on the DAG
- Supports multiple ingresses and multiple egresses
- Multipath spread is maximized in the provisioned DAG within practical constraints
- Amount of state needed to setup the DAG is significantly less
  - Setup junction state at each node on the DAG (as opposed to setting up path state)
- Amount of churn after a resource-failure/resource-degradation/traffic-demand-change event is significantly less
  - Shape of the DAG is largely static post setup
    - No unnecessary addition/deletion of routes or next-hops
  - Automatic adjustment of junction bandwidth and next-hop load-share



Multipath Traffic Engineering MPTE

- Adding TE to Multipath

Clos Network – AIML Cluster
(Canonical Multipath Network)

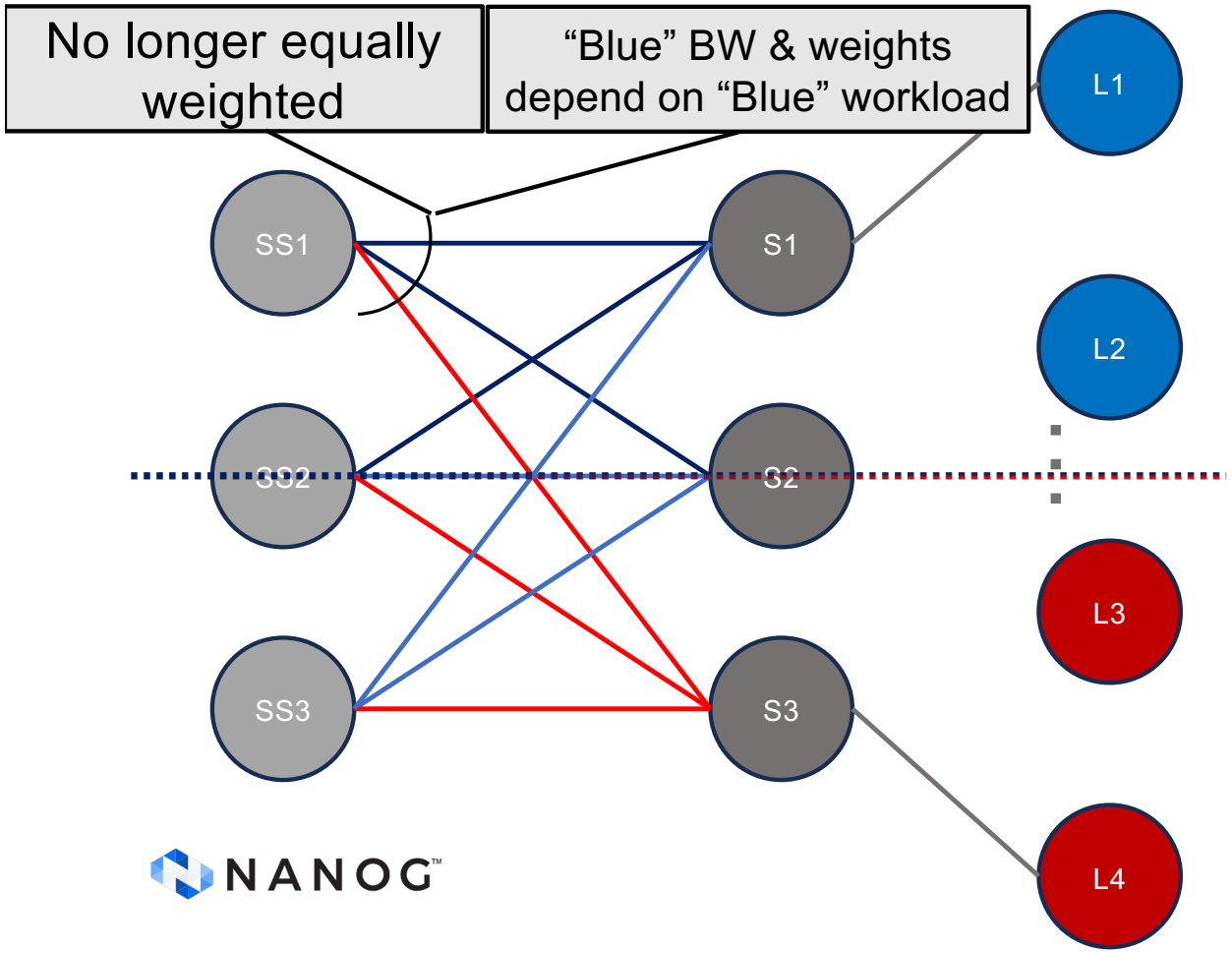


Very structured network: as symmetric as possible

All links between nodes SS_m and S_n are of the same cost/capacity

Similarly for links between S_n and L_k

Adding TE to the ML cluster



Non-multipath TE is a total non-starter in ML clusters!

The network has been purpose-engineered for ECMP. Can TE also help?

NO, if the entire cluster is working on a single training task

YES, if the cluster is split among multiple ML inference tasks

COLOR CPUs/GPUs/ network and reserve resources ***by task***

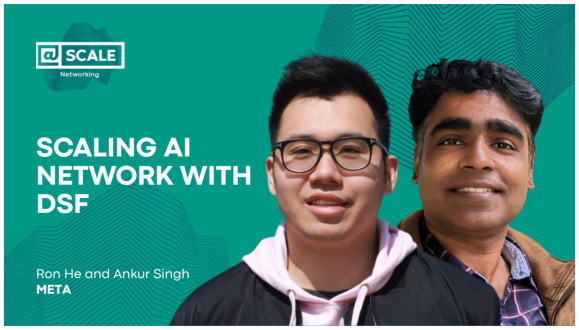
Adding TE to Multipath Enables Network Scheduling

AI/ML workloads are scheduled: how many CPUs/GPUs, how much memory does the job need? Where should it be placed?

The spend on compute resources dwarfs that of network …

But nonetheless, the network appears to be an outsize source of GPU stalls, job delays, even aborts
→ requiring new approaches

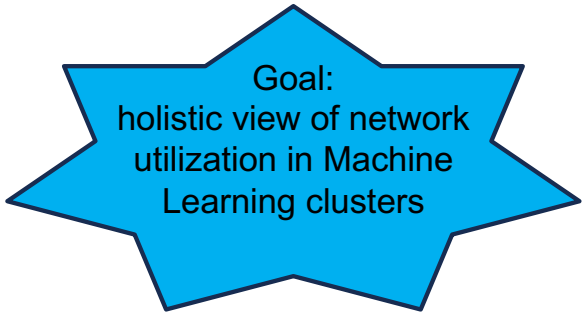
Disaggregated Scheduled Fabric: Scaling Meta's AI Journey



POSTED ON OCTOBER 16, 2025 TO DATA CENTER ENGINEERING

10X Backbone: How Meta Is Scaling Backbone Connectivity for AI

ML Network Scheduling



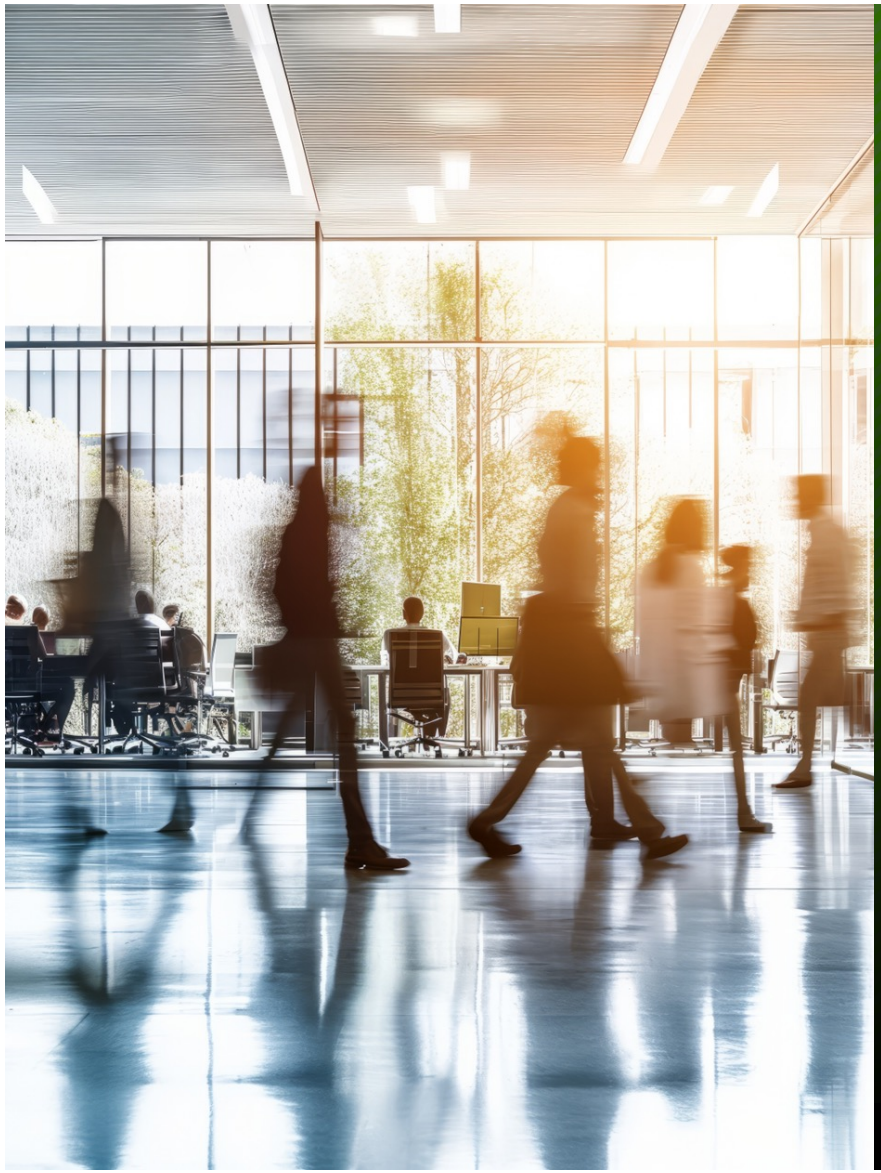
Currently focused on reactive approach (signaling and reacting to congestion)

Proactive approach: avoid congestion, prepare for link/node failures.

In addition, add mechanisms to detect and react to congestion and/or failure

draft-kompella-rtgwg-mlnwsched

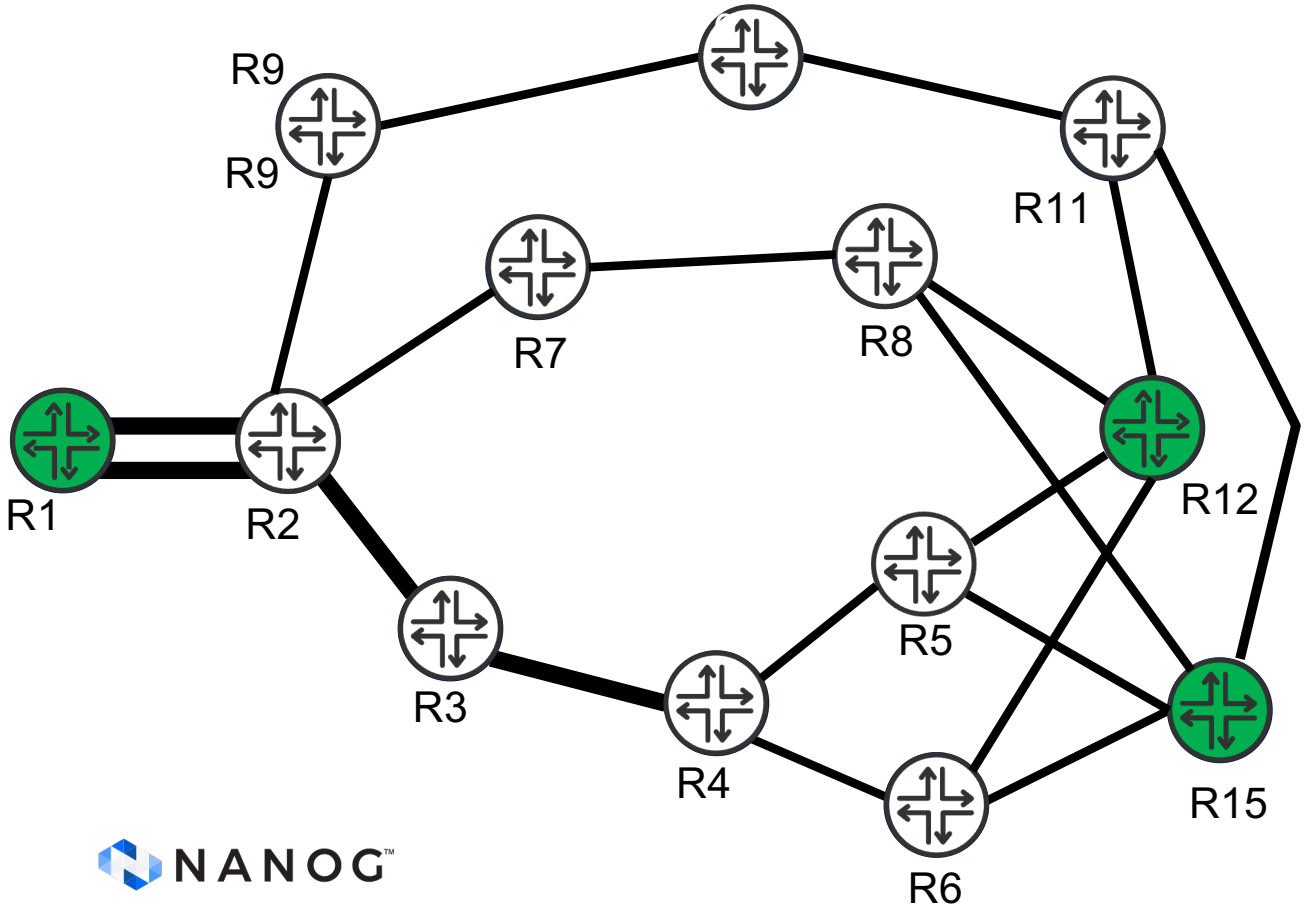
Submitted 2025-10-20



MPTE Directed Acyclic Graph MPTED

- Key Concepts

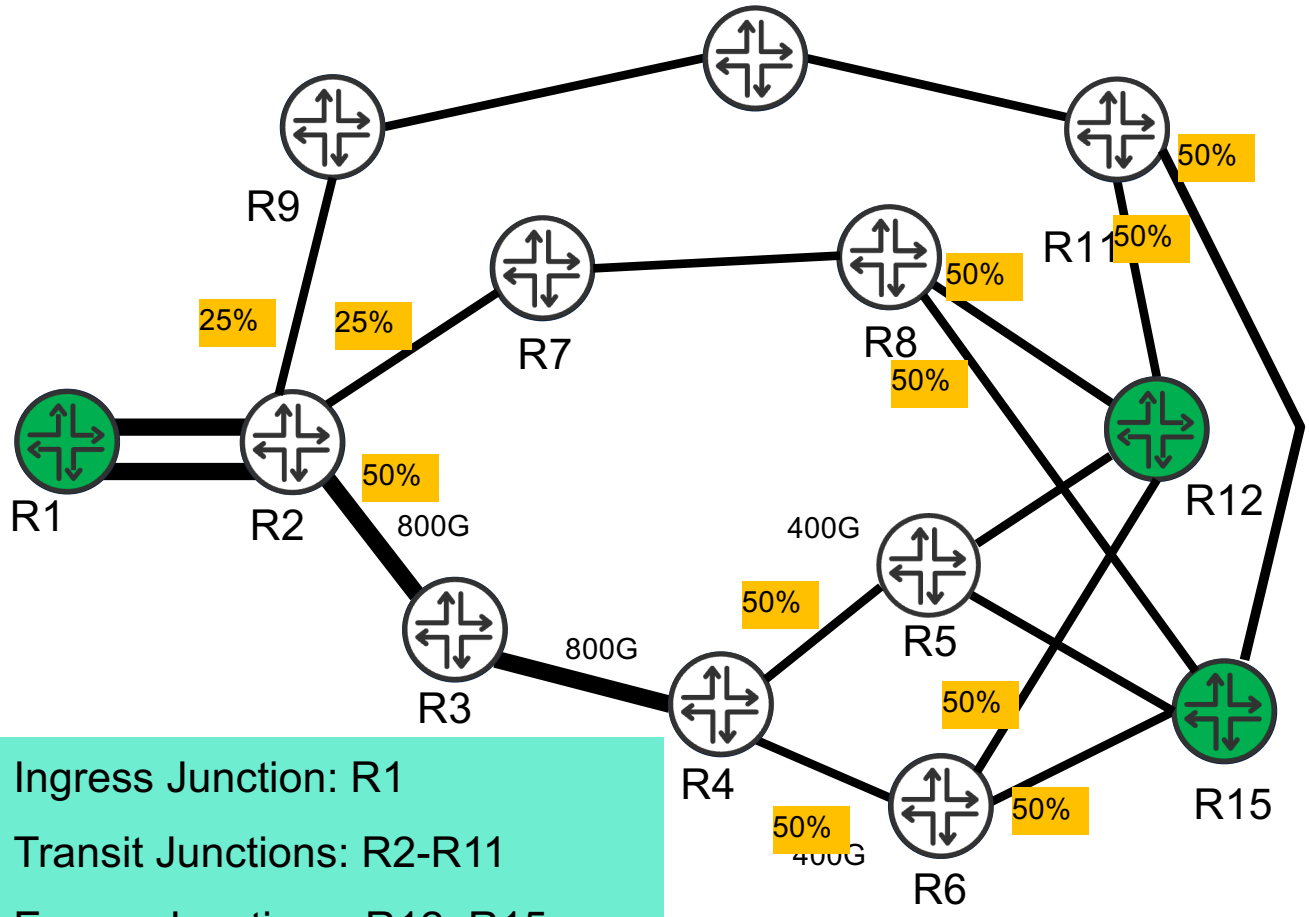
MPTED Tunnel



- TE construct that contains a constrained set of paths representing an optimized Directed Acyclic Graph (DAG) from one or more ingresses to one or more egresses
- The paths that make up an MPTED tunnel traverse a set of junction nodes
- MPLS (Signaled Labels, Static Labels) or IP based

- Ingress: R1
- Egresses: R12, R15

MPTED Junction



- Ingress Junction: R1
- Transit Junctions: R2-R11
- Egress Junctions: R12, R15

- TE construct associated with the MPTED tunnel at each node
- Junction state consists of: *incoming bandwidth*, a set of *previous-hops* (JCT-PHOPs) and a set of *next-hops* (JCT-NHOPs) with weighted load-balancing
  - Each next-hop is associated with a relative load-share

- Provisioning an MPTED tunnel involves signaling the state associated with each junction

Theory of Operation

MPTED Tunnel Originator (TO)

User specifies *intent*:

- Ingresses
- Egresses
- Incoming bandwidth at each ingress
- Constraints and Optimization Objective

MPTED Computer (MC)

MC computes the DAG:

- Takes the specified constraints and optimization objective into account
- Computation result is a set of junctions

MPTED Signaling Source (SS)

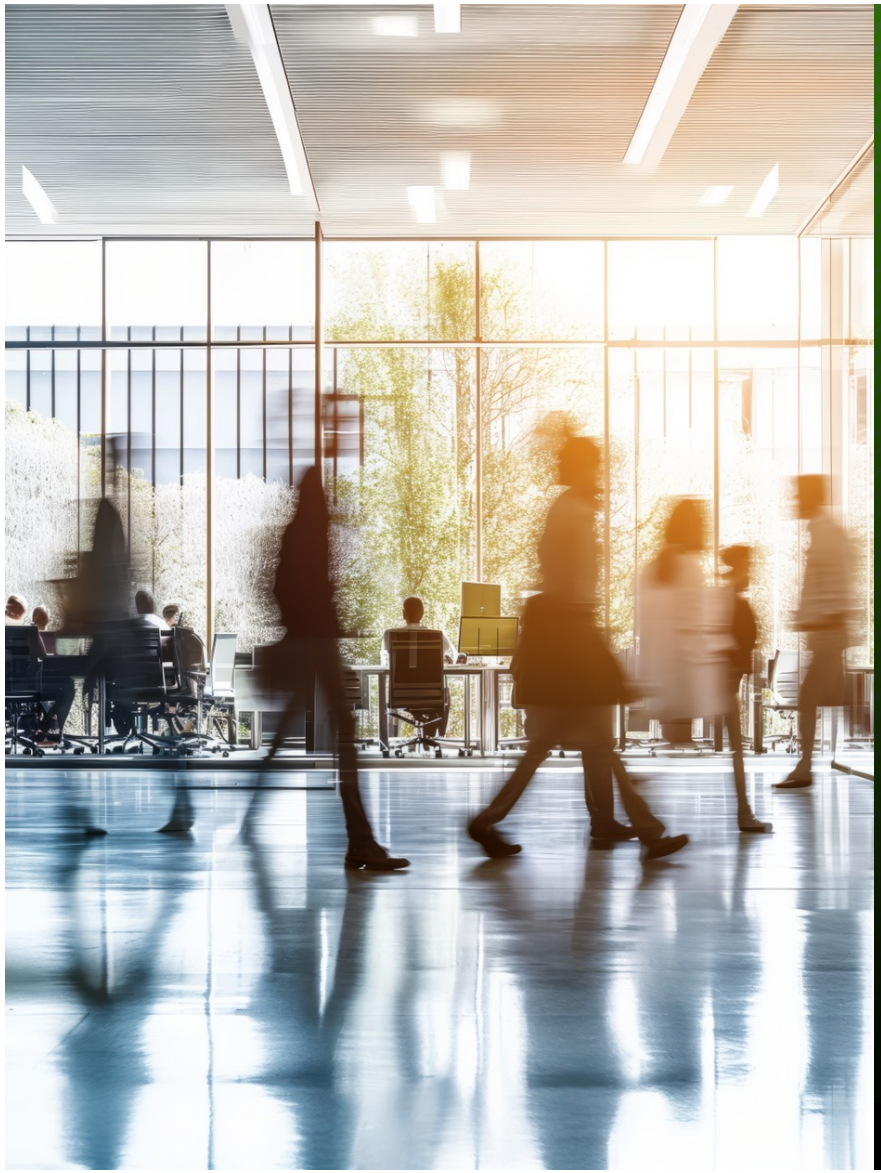
SS provisions the MPTED junction state:

- Signaling messages to each junction node
- When all junctions are provisioned, ingresses can start sending traffic into the MPTED tunnel

Signaling Options

- RSVP-TE
- BGP/PCEP
- Programmable API
  - backed by a YANG data model

- Initial prototype uses RSVP-TE (for good reasons)
- Other prototypes using gRPC and BGP underway



MPTED RSVP

Why RSVP?

- A bandwidth engineered construct requires reservation of resources
- Enables distributed provisioning of MPTED tunnels
- Enables seamless use of signaled MPLS label switching
- Enables ordered admission control and priority-based preemption
- Enables automatic update of link state resource reservation
- Enables ordered programming of labeled routes
- Provides an option to incrementally deploy MPTED tunnels
  - Interoperate with traditional RSVP signaling procedures

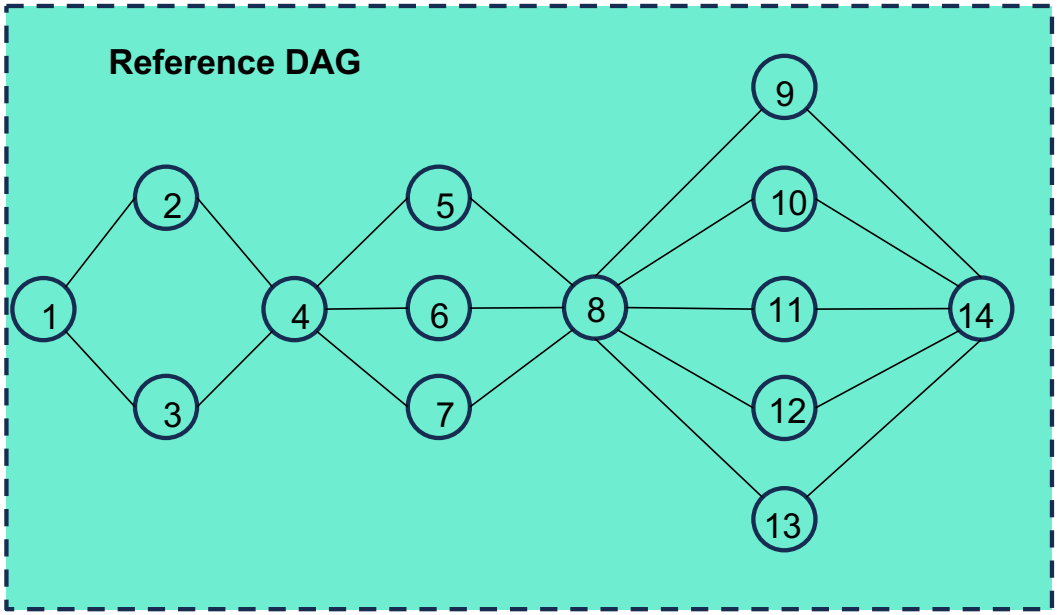
RSVP Messages for Junction Management

- (Signaling) Source to Junction (S2J) Messages
  - JunctionCreate
    - RSVP MPTED Path
  - JunctionUpdate
    - RSVP MPTED Path
  - JunctionDelete
    - RSVP MPTED PathTear (with or without CONDITIONS object)

- Junction to Source (J2S) Messages
  - JunctionNotify
    - RSVP MPTED Notify
  - ResourceNotify
    - RSVP Rsrc Notify

- Junction to Junction (J2J) Messages
  - Upstream (J2JU) Messages
    - JunctionNextHopReservation
      - RSVP MPTED Resv
    - JunctionDown
      - RSVP MPTED Notify
  - Downstream (J2JD) Messages
    - JunctionDelete – Conditional
      - RSVP MPTED PathTear (with CONDITIONS object)
    - JunctionNotReady
      - RSVP MPTED ResvErr

Container LSP vs MPTED Tunnel – Signaling State Comparison

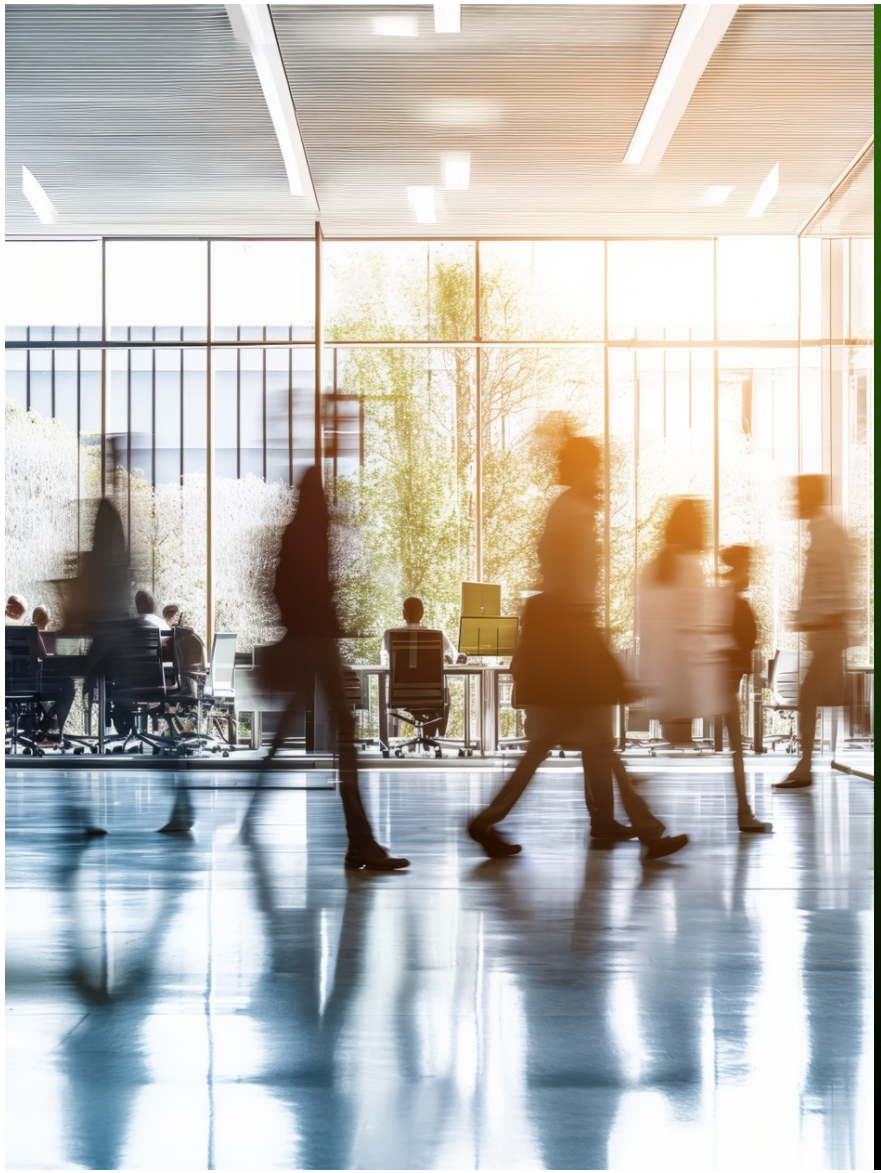


Container LSP-based

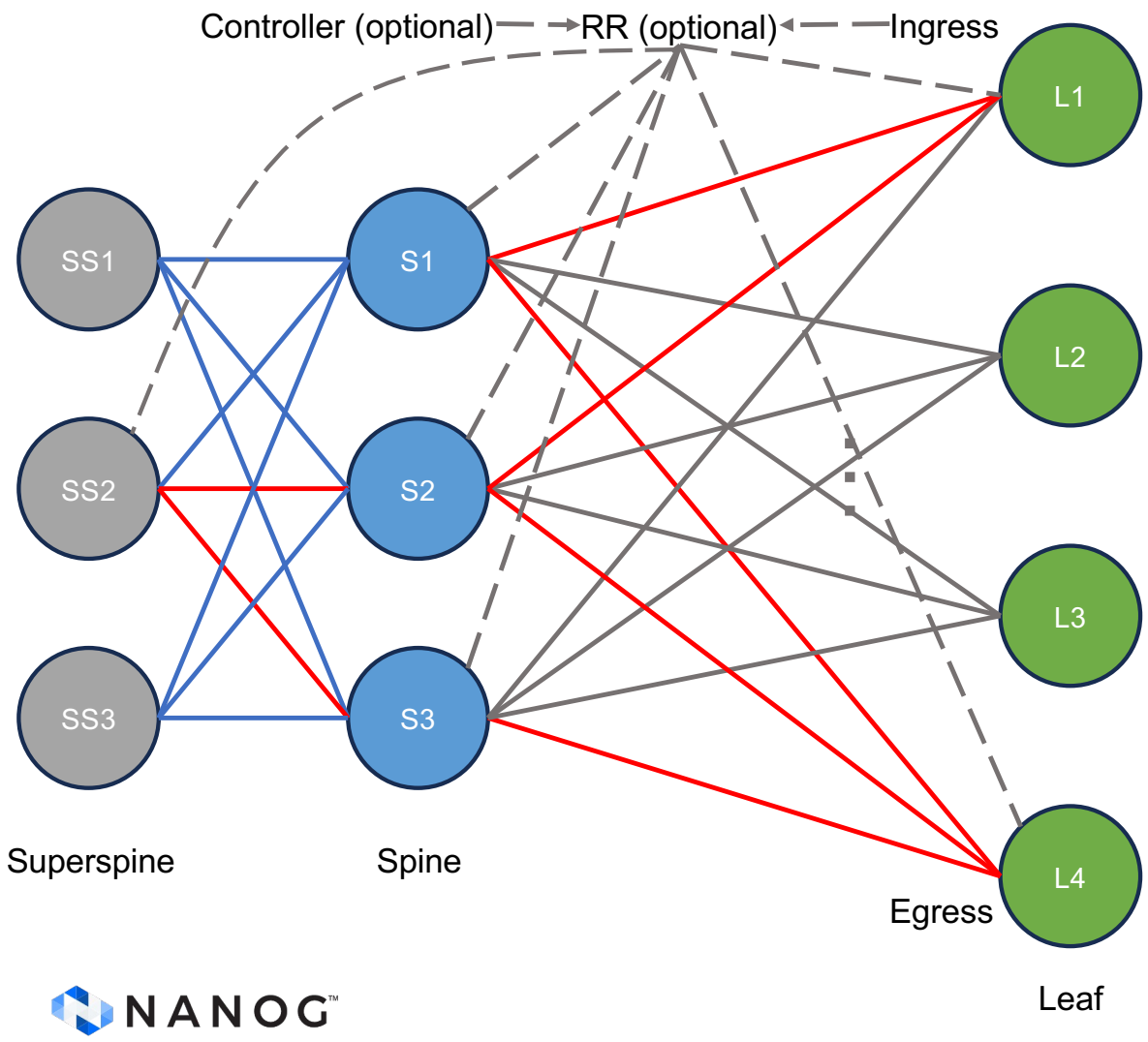
- *30 member LSPs to cover all paths*
- *360 messages on the wire*
- *30 PSBs and 30 RSBs on R4 and R8*

MPTED Tunnel-based

- *1 MPTED tunnel to cover all paths*
- *46 messages on the wire*
- *1 JSB with 2 phops and 3 nhops on R4*
- *1 JSB with 3 phops and 5 nhops on R8*



MPTED BGP



SS (ingress/controller) originates one BGP route for each Junction

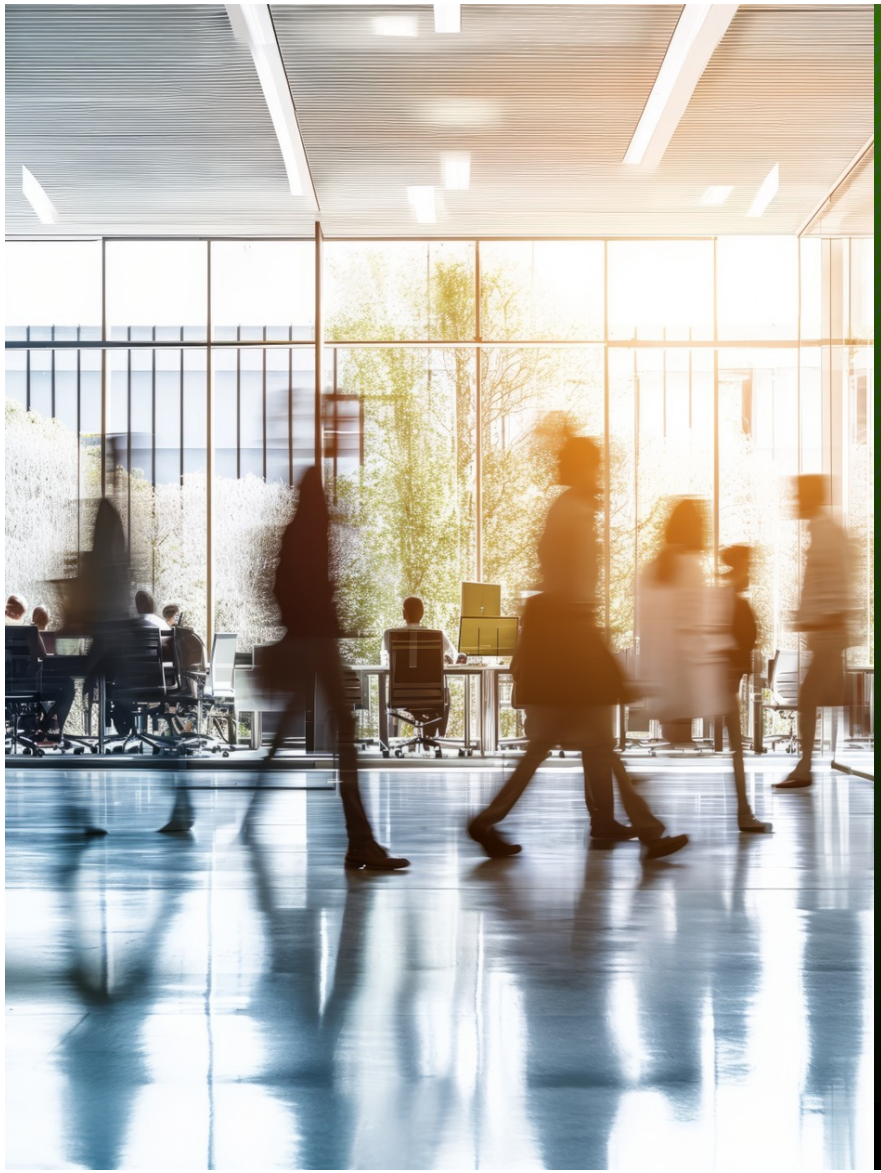
- Each targeted at a node
- Encoding ID/PHOP/NHOP/BW info
  - With "upstream-assigned" encapsulation info (e.g., label) unless ordered control is used
  - This allows the forwarding state to be programmed on the junction node

With ordered control, each junction node originates a BGP route targeted at each of its PHOP

- Carrying downstream-assigned encapsulation info (e.g. label)

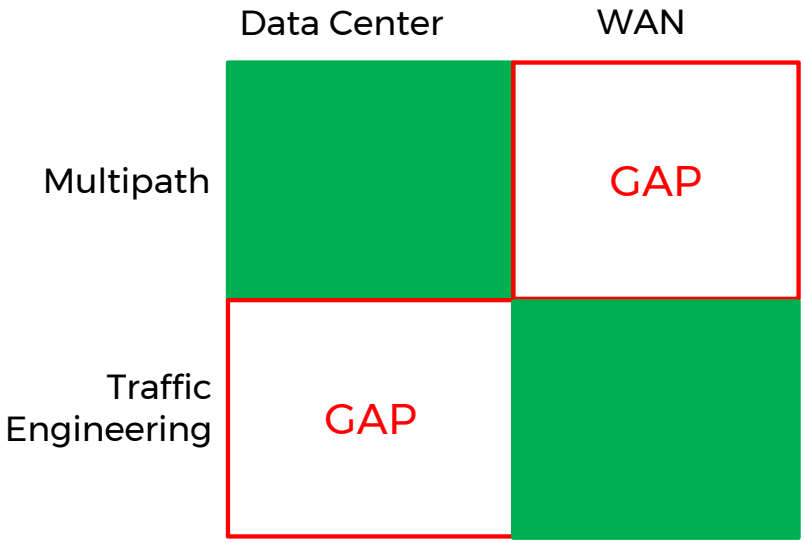
The routes are propagated to and stops at the targeted node

- Following EBGP sessions
- Or optionally via the RR



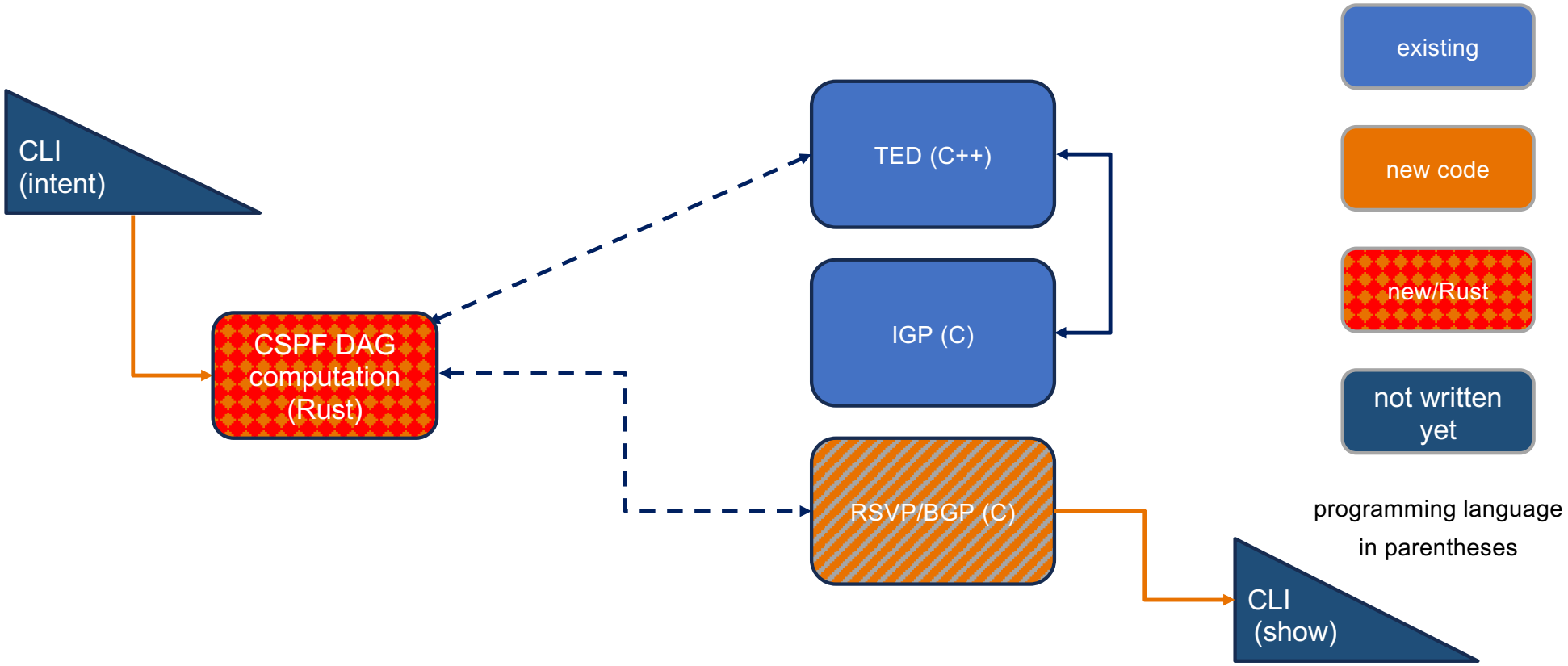
Takeaways

2 Domains + 2 Gaps → 1 Solution

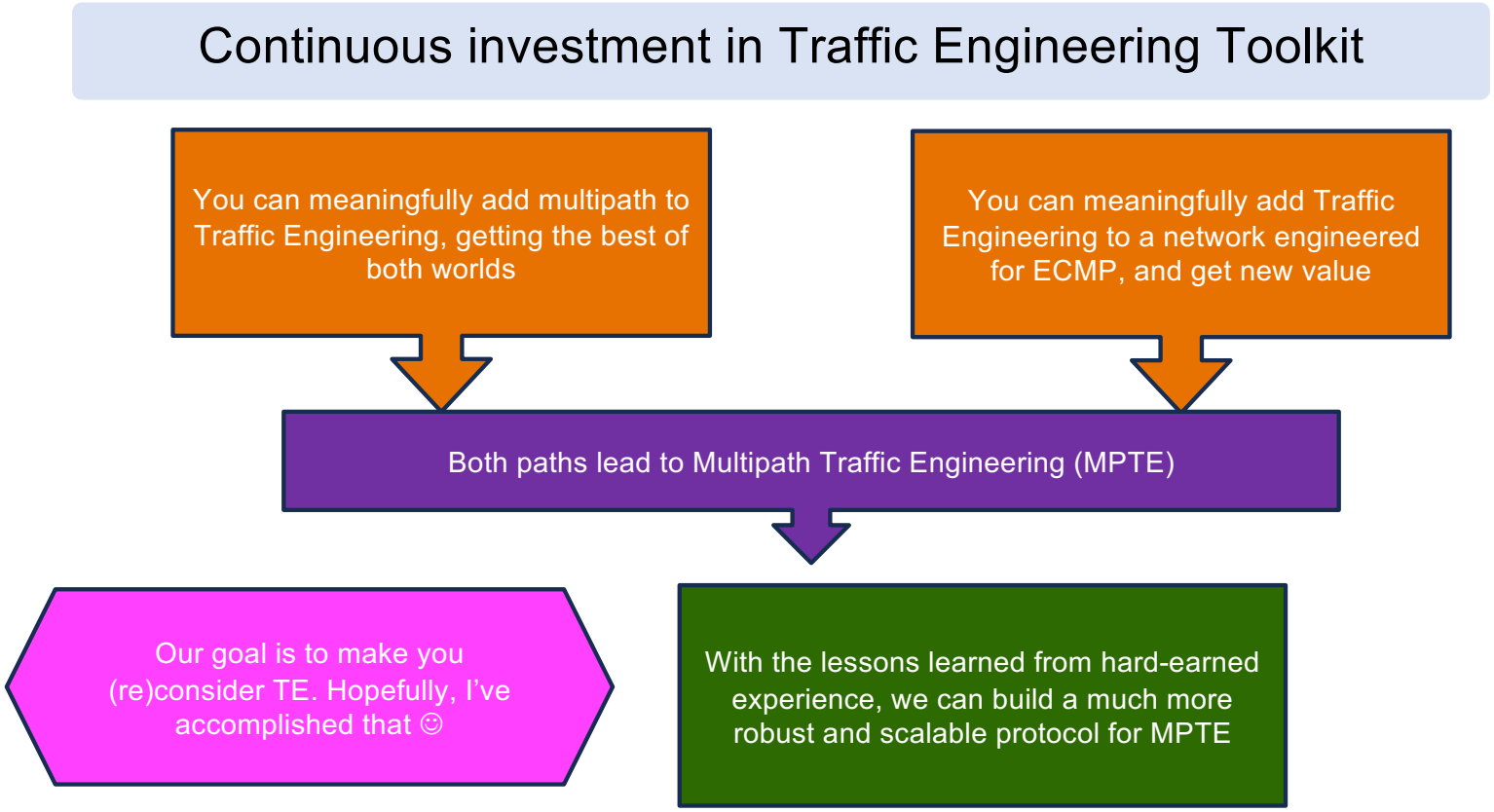


- Multi-Path Traffic Engineering
- Same solution - WAN and AI DC
- High utilization, Resilient
- Congestion-free, Low latency/loss

Current Status: Prototype Underway



Key takeaways



Thank you!

(some supplemental slides follow)

04-FEB-2026

kireeti.kompella@hpe.com
zhaohui.zhang@hpe.com

NANOG™

References [1] – IETF Drafts

Workgroup: TEAS WG
Internet-Draft: draft-kompella-teas-mpte-01
Published: 7 July 2025
Intended Status: Standards Track
Expires: 8 January 2026

K. Kompella
Juniper Networks
L. Jalil
Verizon
M. Khaddam
Cox Communications
A. Smith
Oracle Cloud Infrastructure

Multipath Traffic Engineering

Workgroup: LSR WG
Internet-Draft: draft-kompella-lsr-mptecap-00
Updates: 5073 (if approved)
Published: 7 July 2025
Intended Status: Standards Track
Expires: 8 January 2026

K. Kompella
Juniper Networks

Multipath Traffic Engineering Capabilities

TEAS WG
Internet-Draft
Intended status: Standards Track
Expires: 8 January 2026

K. Kompella
V. P. Beeram
C. Ramachandran
Juniper Networks
7 July 2025

RSVP-TE Extensions for Multipath Traffic Engineered Directed Acyclic Graph Tunnels
draft-kbr-teas-mptersvp-01

TEAS Working Group
Internet-Draft
Intended status: Standards Track
Expires: 8 January 2026

V. P. Beeram
K. Kompella
Juniper Networks
7 July 2025

A YANG Data Model for Multipath Traffic Engineering Directed Acyclic Graph (MPTED) Tunnels and Junctions
draft-beeram-teas-yang-mpted-00

PCEP Working Group
Internet-Draft
Intended status: Standards Track
Expires: 8 January 2026

V. P. Beeram
K. Kompella
Juniper Networks
7 July 2025

Path Computation Element Communication Protocol (PCEP) Extensions for Multipath Traffic Engineered Directed Acyclic Graph (MPTED) Tunnels
draft-beeram-pce-pcep-mpted-00

NANOG™

References [2] – Videos

MPLS NETWORLD Paris 2025

https://player.vimeo.com/video/1069205740?autoplay=1

NANOG

IETF 122 Bangkok 2025

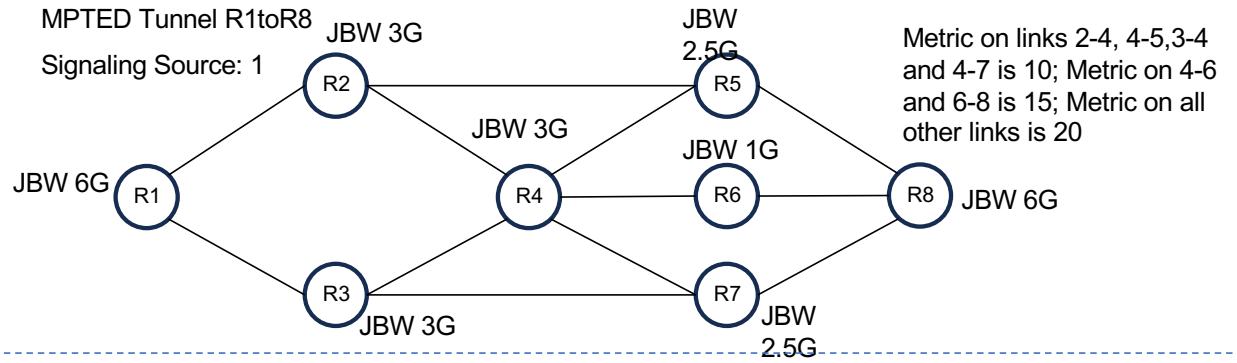
https://www.youtube.com/watch?v=Osm0uUddYQ0

Optimizing Signaling – Design Guidelines

- Minimize “Refresh” message processing
  - Refresh-interval independent RSVP [RFC8370] procedures are always ON
- Minimize signaling adjacency failure notifications
  - Relaxed hello-interval by default
- Minimize signaling notifications when a link fails/degrades
  - Resource Notifications are always ON
    - No per-state notifications sent when a topological-element goes down or gets degraded
- Minimize “Trigger” message processing
  - Signaling-Source sends PATH message (JUNCTION state setup) directly to the junction
  - Avoid unnecessary junction state updates

Initial Setup Sequence

- Initiation of setup sequence on MPTED tunnel signaling source, R1:
  - R1 sends an M-Path message to each junction node (R2, R3, R4, R5, R6, R7, and R8)
  - R1 processes the ingress JUNCTION, constructs a JSB, and waits for an M-Resv message to arrive from each JCT-NHOP (R2 and R3).



- M-Path message processing on transit junction nodes (R2, R3, R4, R5, R6, R7):
  - Each transit junction node processes the JUNCTION, constructs a JSB, and waits for an M-Resv message to arrive from each JCT- NHOP.

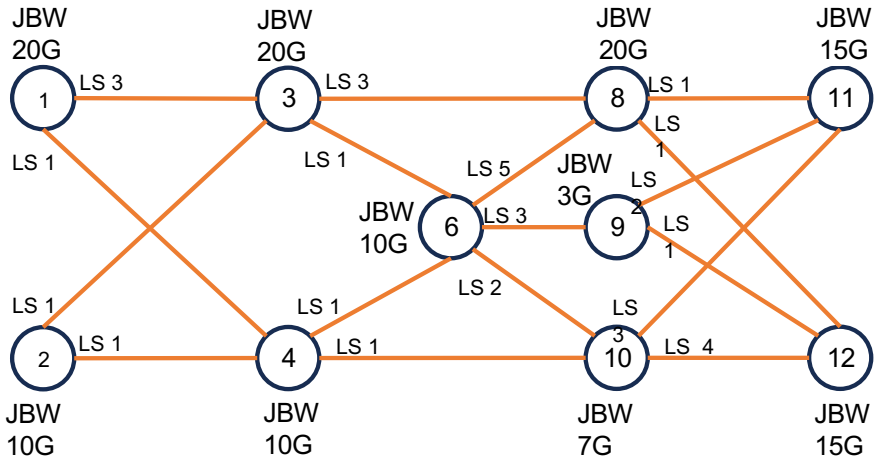
- M-Path message processing on egress junction node, R8:
  - R8 processes the JUNCTION and constructs a JSB.
  - R8 sends an M-Resv message to each JCT-PHOP (R5, R6, and R7) with IMPLICIT NULL Label (3).
  - R8 sends an M-Notify message to R1, indicating that the junction processing is complete at R8.

- M-Resv message processing on transit junction nodes (R2, R3, R4, R5, R6, R7):
  - Each transit junction node waits until M-Resv messages are received from all available JCT-NHOPs and then:
    - Updates BW reservation on TE-links.
    - Allocates a label for each JCT-PHOP and programs the corresponding labeled route.
    - Sends an M-Resv message to each JCT-PHOP with the corresponding allocated label.
    - Sends an M-Notify message to R1, indicating that the junction processing is complete on the node.

- M-Resv message processing on ingress junction node, R1:
  - R1 waits until M-Resv messages are received from all JCT-NHOPs (R2 and R3) and then:
    - Updates BW reservation on TE-links.
    - Programs a route for the MPTED tunnel.
    - Notifies the signaling source (itself) that the junction processing is

- M-Notify message processing on the signaling source:
  - The signaling source (R1) considers the setup sequence complete when confirmation of junction provisioning is received from all junctions.

Current Status: Prototype Underway



Goal: Provision the following MPTED Tunnel and display the state associated with the tunnel and the junctions

- MPTED Tunnel:
  - MTNL_West_to_East_001
  - Ingresses:
    - 1.1.1.1 (bandwidth 20g), 1.1.1.2 (10g)
  - Egresses: 1.1.1.11, 1.1.1.12
  - Include admin-group green
  - Install 1.1.1.100 (anycast address)

Scenario 1 – Initial Target:

- Tunnel Originator, DAG Computer and Signaling Source are located on the ingress
  - Type – MPLS-Signaled-Labels
  - Signaling-Type - RSVP-TE

Scenario 2:

- Tunnel Originator, DAG Computer and Signaling Source are located on a controller
  - Type – MPLS-Static-Labels
  - Signaling-Type - BGP

MPTED YANG Module: High-Level Model Structure

```
module: ietf-mpted

  augment /te:te:
    +--rw mpted-tunnels
      +--rw tunnel* [originator identifier]
         +--rw originator                  inet:ip-address
         +--rw identifier                  uint32
         +  ..
         +--ro junctions
            +--ro junction* [node-id]
               +--ro node-id               inet:ip-address
               +  ..
               +--ro phops
               |  +--ro phop* [hop-address hop-index]
               |     +--ro hop-address     inet:ip-address
               |     +--ro hop-index       uint32
               |     +  ..
               +--ro nhops
               |  +--ro nhop* [hop-address hop-index]
               |     +--ro hop-address     inet:ip-address
               |     +--ro hop-index       uint32
               |     +  ..
               +--ro phops-pending-deletion
               |  +--ro phop* [hop-address hop-index]
               |     +--ro hop-address     inet:ip-address
               |     +--ro hop-index       uint32
               |     +  ..
               +--ro nhops-pending-deletion
                  +--ro nhop* [hop-address hop-index]
                     +--ro hop-address     inet:ip-address
                     +--ro hop-index       uint32
```

- The top-level 'te' container [I-D.draft-ietf-teas-yang-te] is augmented with a set of MPTED tunnels.
- The 'mpted-tunnels' container carries a list of tunnel entries.
  - Each tunnel entry includes the parameters required to produce a list of junctions that need to programmed in the network.
  - The state for each junction entry consists of previous-hops ('phops' container) and next-hops ('nhops' container) associated with the current version, as well as those that are pending deletion ('phops-pending-deletion' and 'nhops-pending-deletion' containers).

MPTED-JCT YANG Module: High-Level Model Structure

```
module: ietf-mpted-jct

    augment /te:te:
      +--rw mpted-junctions
         +--rw junction* [node-id originator identifier]
            +--rw node-id                inet:ip-address
            +--rw originator             inet:ip-address
            +--rw identifier             uint32
            +  ..
            +--rw phops
            |  +--rw phop* [hop-address hop-index]
            |     +--rw hop-address      inet:ip-address
            |     +--rw hop-index        uint32
            |     +  ..
            +--rw nhops
               +--rw nhop* [hop-address hop-index]
                  +--rw hop-address      inet:ip-address
                  +--rw hop-index        uint32
```

- The top-level 'te' container [I-D.draft-ietf-teas-yang-te] is augmented with a set of MPTED junctions.
- The 'mpted-junctions' container carries a list of junction entries.
  - Each junction entry includes information about the associated set of previous-hops ('phops' container) and next-hops ('nhops' container).