

Multipath Traffic Engineering

IETF 124 - TEAS WG
draft-kompella-teas-mpte

Kireeti Kompella, HPE

Luay Jalil, Verizon

Mazen Khaddam, Cox Communications

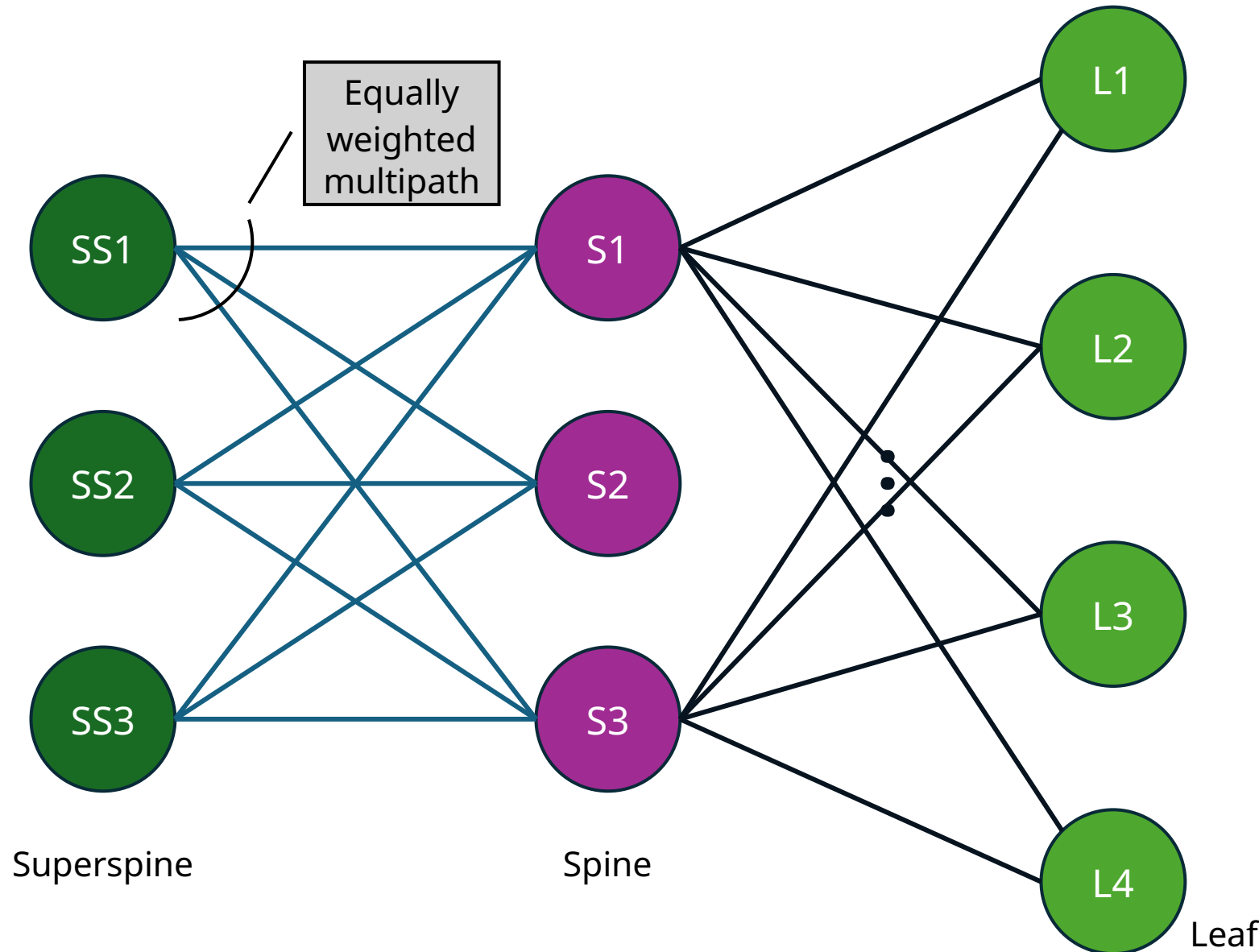
Andy Smith, Arrcus

DAG-based Multipath Traffic Engineering

- New paradigm offering the benefits of TE and Load Balancing
 - Enhance multipath capability in traditional TE networks
 - **Add TE to networks designed for ECMP** and get new benefits
 - As you will see, MPTE has applications in Deep Learning clusters
- Uses Directed Acyclic Graphs (DAGs) as the primitive TE construct as opposed to Paths

Clos Network – For a Deep Learning Cluster

(Epitome of Multipath Networks!)



Very structured network:
as symmetric as possible

All links between nodes
 SS_m and S_n are of equal
length and of the same
capacity

Similarly for links between
 S_n and L_k

Adding TE to the DL cluster

No longer equally weighted multipath

"Blue" bandwidth & weights depend on "Blue" workload

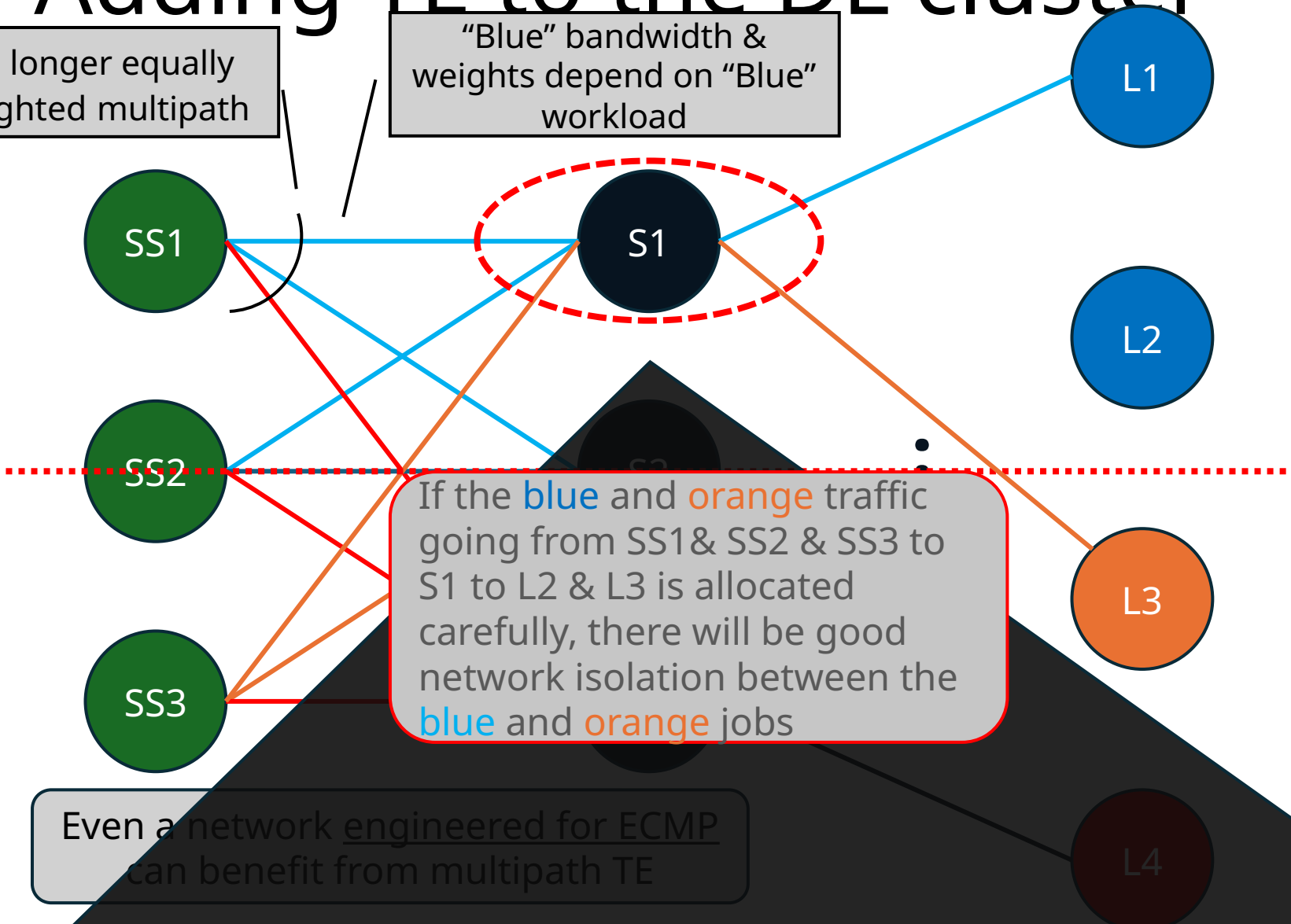
Non-multipath TE is a total non-starter in DL clusters!

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

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

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

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



Even a network engineered for ECMP can benefit from multipath TE

If the blue and orange traffic going from SS1 & SS2 & SS3 to S1 to L2 & L3 is allocated carefully, there will be good network isolation between the blue and orange jobs

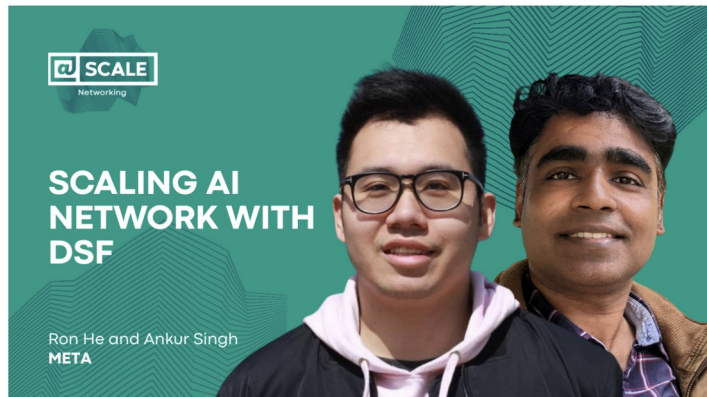
Adding TE to Multipath Enables Network Scheduling

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

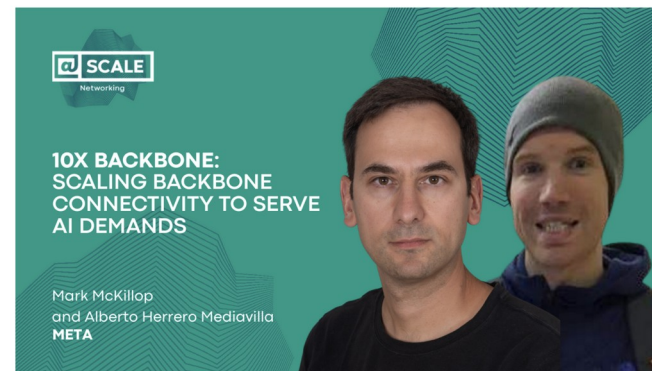
But nonetheless, the network
appears to be an outsize source
of GPU stalls, job delays, even
aborts
□ requiring new approaches

Disaggregated Scheduled Fabric: Scaling Meta's AI Journey



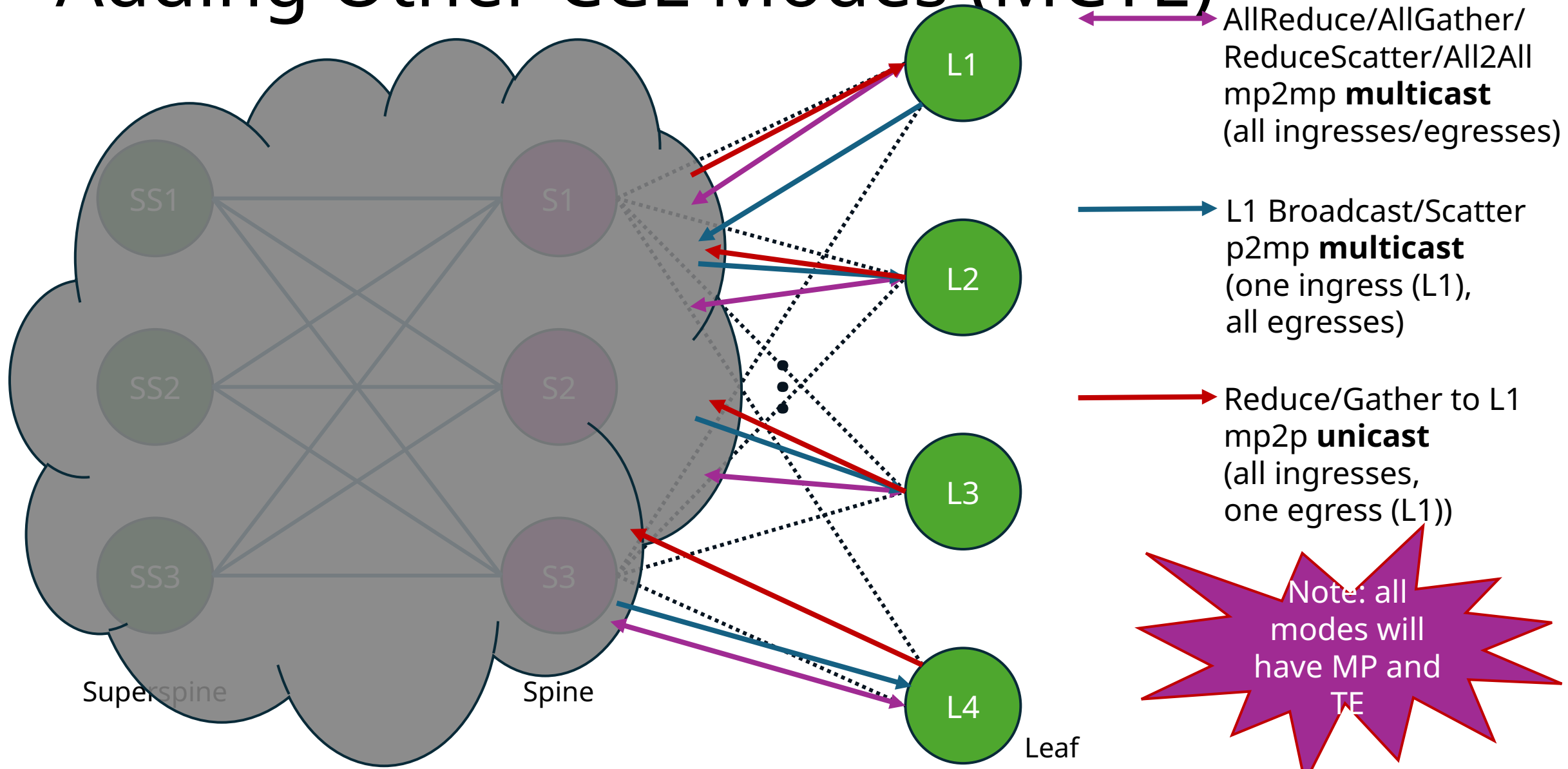
POSTED ON OCTOBER 16, 2025 TO DATA CENTER ENGINEERING

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



See draft-kompella-teas-mcte
(multicast TE)

Adding Other CCL Modes (MCTE)



MPTE – Key Differentiators / Attributes

- Facilitates unequal-cost load balancing at every junction on the DAG
 - Not just on ingress
- 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

MPTED Tunnels and Junctions

▪ 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

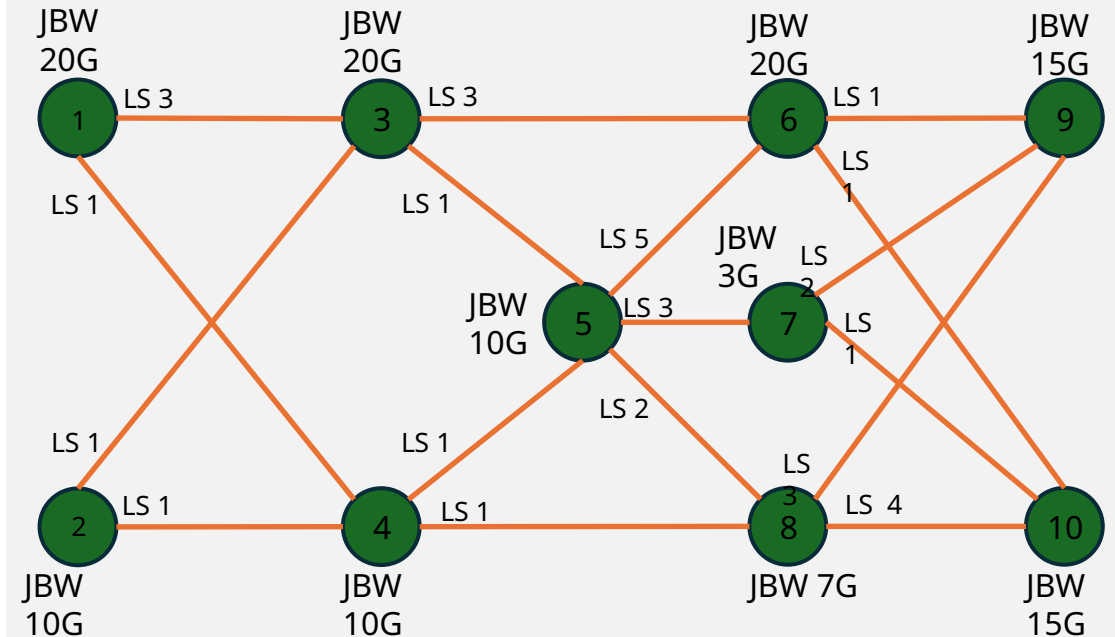
▪ MPTED Junction

- TE construct associated with the MPTED tunnel at each junction node
- Constitutes a set of previous-hops (JCT-PHOPs) and a set of next-hops (JCT-NHOPs) over which traffic is load-balanced in a weighted fashion

- Provisioning an MPTED tunnel involves provisioning the control and forwarding plane state associated with the MPTED junction at each junction node

MPTED Tunnel: Tun-West-To-East [30G]

- Ingresses – {1 [20G], 2 [10G]}; Egresses – {9, 10}
- Constraint – Include Green (Resource Affinity)
- Optimization Objective – TE metric
- Type – MPLS Signaled Labels
- Signaling Type – RSVP



JBW: Junction Bandwidth
LS: Load Share

MPTED Tunnels – Key Functions

▪ MPTED Tunnel Originator (TO)

- Responsible for maintaining configuration and operational state for the tunnel
 - Identifier, Ingresses, Egresses, Constraints, Optimization objective

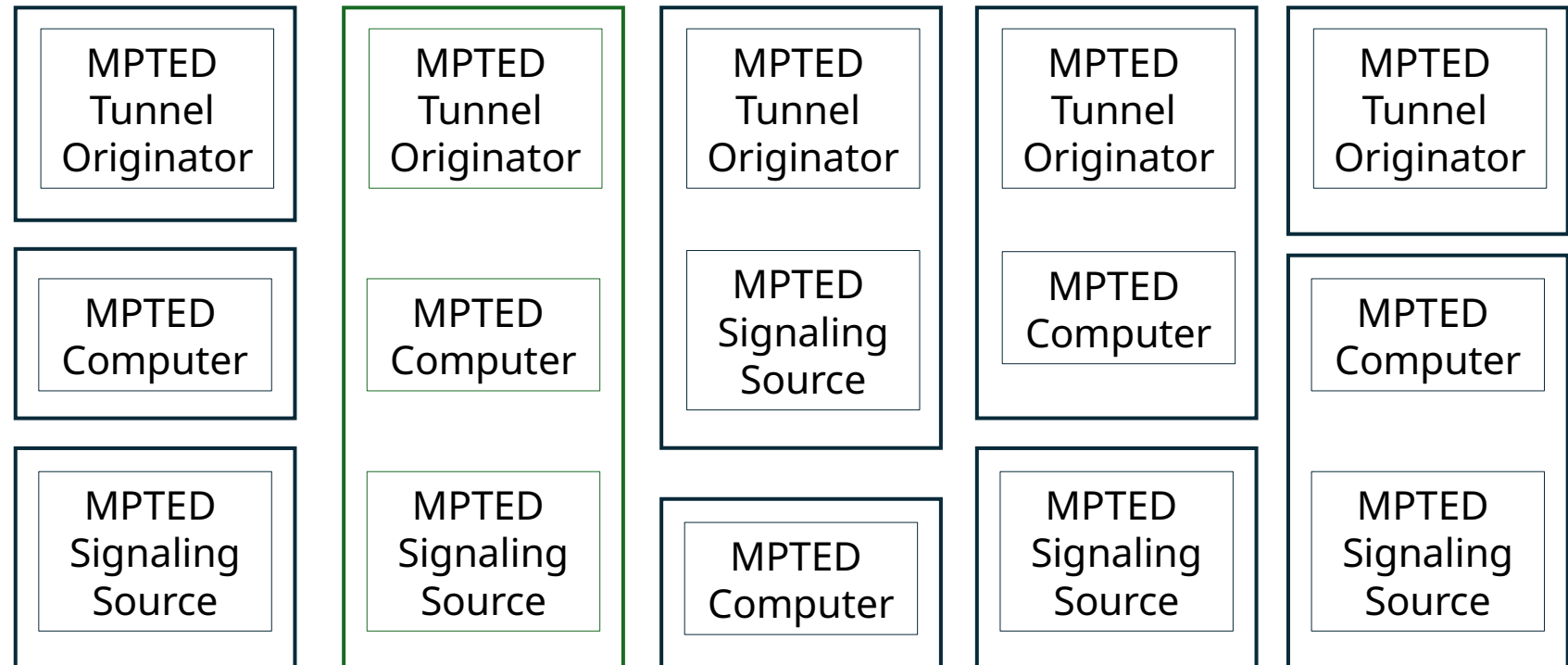
▪ MPTED Computer (MC)

- Responsible for computing an MPTED DAG that caters to the constraints and optimization objective
 - Computation result is a set of unordered elements called JUNCTIONs
 - Each element includes the bandwidth coming in and going out of the junction, a list of previous-hops (JCT-PHOPs), and a list of next-hops (JCT-NHOPs)

▪ MPTED Signaling Source (SS)

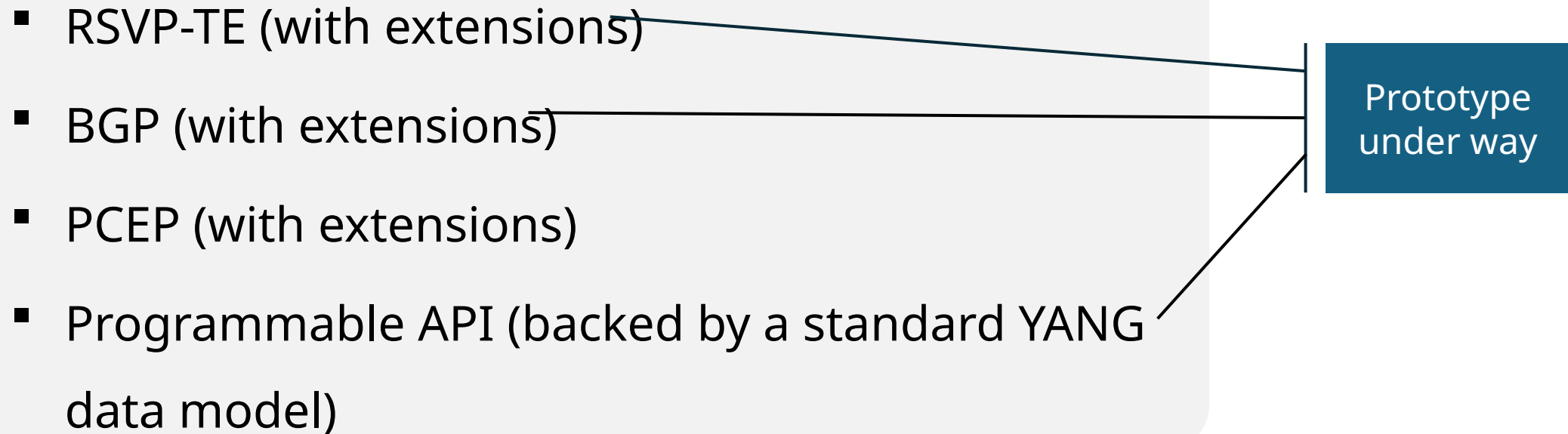
- Responsible for provisioning and maintaining junction state on each JUNCTION
 - Junction State Block (JSB) includes state for JSB-PHOPs and JSB-NHOPs

- These functions may be performed by one or more entities



Signaling Options

- RSVP-TE (with extensions)
- BGP (with extensions)
- PCEP (with extensions)
- Programmable API (backed by a standard YANG data model)



Prototype
under way

Tunnel Types / Data plane

- Allows for several types of data plane:
 - “Signaled” MPLS
 - Static MPLS (shared vs non-shared)
 - Many types of IP tunnels
- Has an overview of how to install FIB entries for each tunnel type
 - Further details will be provided in future updates of the base document as well as the signaling documents

RSVP-TE Extensions for Multipath Traffic Engineered Directed Acyclic Graph Tunnels

draft-kbr-teas-mptersvp-03

Kireeti Kompella
Vishnu Pavan Beeram
Chandra Ramachandran
HPE

Introduction

- This document discusses the extensions to RSVP-TE for use as a signaling protocol to provision MPTED tunnels.
 - MPTED tunnels provisioned using RSVP-TE are referred to as RSVP MPTED Tunnels.
- The focus of this version of the document is on discussing how the RSVP-TE protocol is extended to facilitate distributed provisioning of MPTED Tunnels over an MPLS forwarding plane in an intra-domain TE network.
 - Extensions for provisioning MPTED Tunnels over other forwarding plane types will be added in a subsequent revision

Optimized Signaling Procedures – Design Guidelines

- Minimize “Refresh” message processing
 - Refresh-interval independent RSVP [RFC8370] procedures are always ON
- Avoid unnecessary signaling adjacency failures
 - Relaxed hello-interval by default
- Minimize the number of signaling notifications triggered when a link fails/degrades
 - Resource Notifications are always ON
- Minimize “Trigger” message processing
 - Signaling-Source sends PATH message (JUNCTION state setup/update) directly to the junction
 - No hop-by-hop PATH signaling
 - Avoid unnecessary junction state updates
 - Sub-Graph only updates MUST be accommodated

RSVP Signaling 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

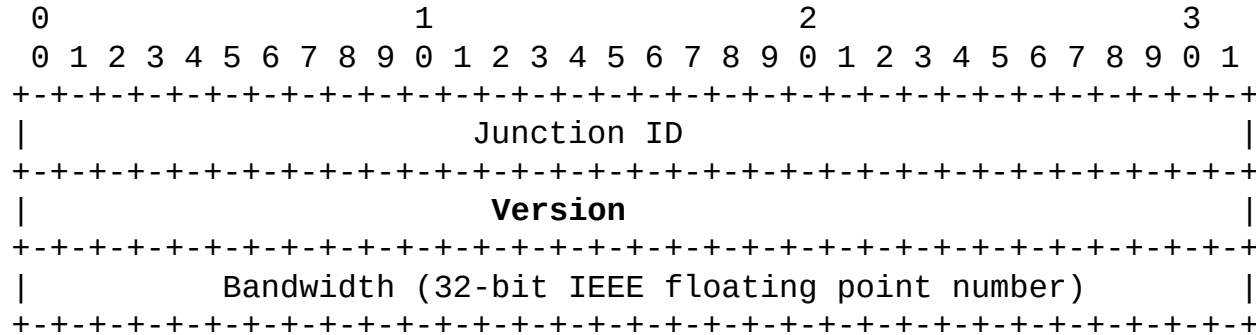
- JunctionNextHopReservation
 - RSVP MPTED Resv
- JunctionDown
 - RSVP MPTED Notify

▪ Downstream (J2JD) Messages

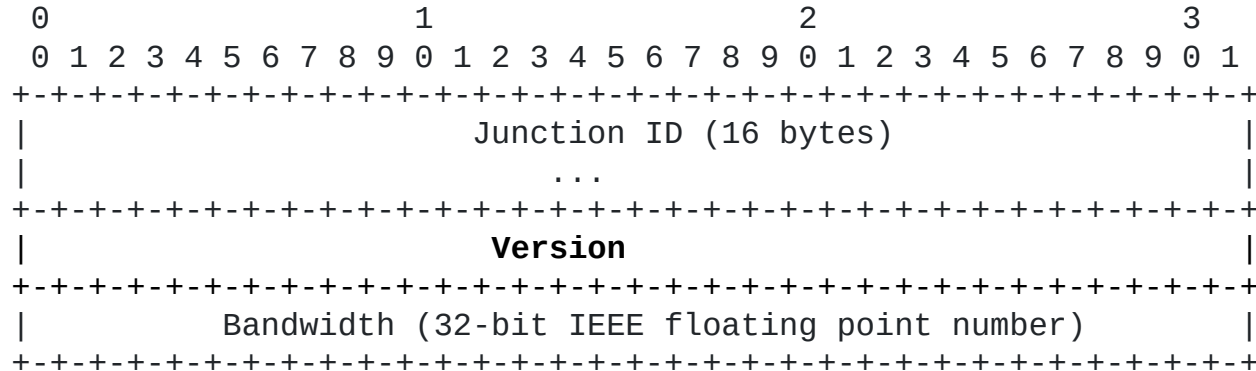
- JunctionDelete – Conditional
 - RSVP MPTED PathTear (with CONDITIONS object)
- JunctionNotReady
 - RSVP MPTED ResvErr

Rev 03 Changes

Class = JUNCTION, JUNCTION_IPv4 C-Type = TBD



Class = JUNCTION, JUNCTION_IPv4 C-Type = TBD



* **Version:** - Instance identifier of the JUNCTION state.

- Added a version field in the JUNCTION object
 - Each JUNCTION entry may be associated with more than one junction instance

A YANG Data Model for Multipath Traffic Engineering Directed Acyclic Graph (MPTED) Tunnels and Junctions

draft-beeram-teas-yang-mpted-01

Vishnu Pavan Beeram

Kireeti Kompella

HPE

Introduction

- This document defines a YANG data model for representing, retrieving, and manipulating Multipath Traffic Engineering Directed Acyclic Graph (MPTED) Tunnels and Junctions.
- The model includes two YANG modules:
 - `ietf-mpted`: For managing MPTED Tunnels on a tunnel originator node.
 - `Ietf-mpted-jct`: For managing MPTED Junctions on a junction node.

MPTED YANG Module: High-Level Model Structure

- The top-level 'te' container is [I-D.draft-ietf-teas-yang-te] is augmented with a set of MPTED tunnels.

```
module: ietf-mpted
augment /te:te:
  +--rw mpted-tunnels
    +--rw tunnel* [originator identifier]
      +--rw originator          inet:ip-address
      +--rw identifier          uint32
      +
      +--ro instances
        +--ro instance* [version]
          +--ro version          uint16
          +
          +--ro junctions
            +--ro junction* [node-id]
              +--ro node-id          inet:ip-address
              +
              +--ro jct-instances
                +--ro jct-instance* [jct-version]
                  +--ro jct-version    uint16
                  +
                  +--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
                          +
                          + ..
```

- The 'mpted-tunnels' container carries a list of tunnel entries.
 - Each tunnel entry includes the set of parameters required to produce a list of junctions that need to be programmed in the network.
 - Each tunnel entry may have more than one instance associated with it.
 - A unique version identifies each instance.
 - Each tunnel instance has a list of junctions associated with it.
 - Each junction entry may have more than one instance (jct-instance) associated with it.
 - A unique junction-version identifies each instance.
 - Each junction instance entry consists of the set of previous-hops ('phops' container) and next-hops ('nhops' container) associated with the given junction version.

MPTED-JCT YANG Module: High-Level Model Structure

- The top-level 'te' container is [I-D.draft-ietf-teas-yang-te] is augmented with a set of MPTED junctions.

```
module: ietf-mpted-jct
  augment /te:te:
    +--rw mpted-junctions
      +--rw junction* [node-id originator tnl-id tnl-vers sig-src]
        +--rw node-id          inet:ip-address
        +--rw originator       inet:ip-address
        +--rw tnl-id           uint32
        +--rw tnl-vers         uint16
        +--rw sig-src          inet:ip-address
      +
      +--rw jct-instances
        +--rw jct-instance* [jct-version]
          +--rw jct-version    uint16
          +
          +--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 'mpted-junctions' container carries a list of junction entries.
 - Each junction entry may be associated with more than one junction instance.
 - Each junction instance includes information about the associated set of previous-hops ('phops' container) and next-hops ('nhops' container) for the given junction version.

Rev 01 Changes

- Each ingress can have a separate static/auto bandwidth profile associated with it
 - When auto-bw is used, only the adjust-interval needs to be common across ingresses
- Each Tunnel entry MAY have more than one tunnel instance associated with it
- Each Junction entry MAY have more than one junction instance associated with it

Next Steps



To Do List

- Architecture
 - Add more details on FIB installation
 - Add details on enabling multicast capability
 - Introduce a new draft on multicast TE using the same constructs as MPTE
- RSVP
 - Add more update sequences
 - Add details on Graceful Restart
- YANG model
 - Add more operational state

- Please send feedback on all the above docs to the TEAS list!

Thank You

draft-kompella-teas-mpte@ietf.org

draft-kbr-teas-mptersvp@ietf.org

draft-beeram-teas-yang-mpted@ietf.org