Multipath Traffic Engineering
Drafts Update

draft-kompella-teas-mpte

draft-kbr-teas-mptersvp

draft-beeram-teas-yang-mpted

Presenter: Vishnu Pavan Beeram, HPE

Multipath Traffic Engineering

draft-kompella-teas-mpte-02

Kireeti Kompella, HPE

Luay Jalil, Verizon

Mazen Khaddam, Cox Communications

Andy J. Smith, Arrcus Inc

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
- Uses Directed Acyclic Graphs (DAGs) as the primitive TE construct as opposed to Paths

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

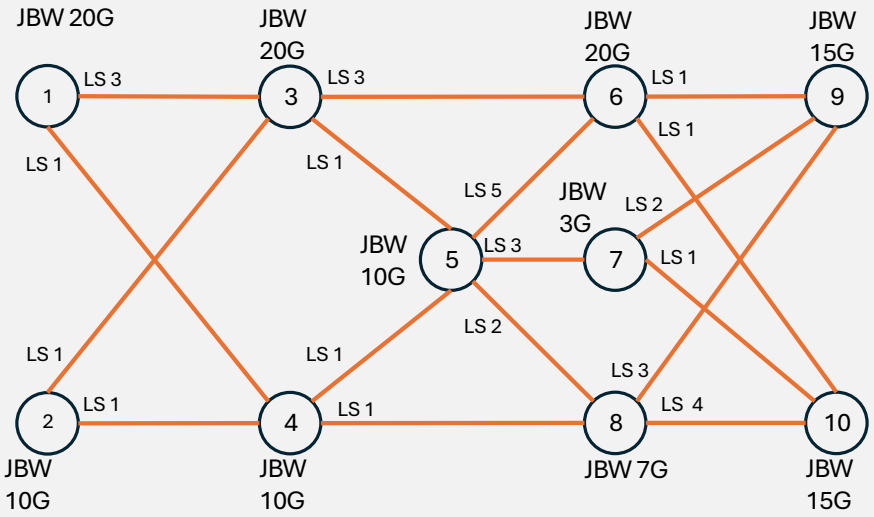
- MPTED Tunnel
  - TE construct that contains a constrained set of paths representing an optimized Directed Acyclic Graph (DAG) from one or more ingresses to one or more egresses
  - The paths that make up an MPTED tunnel traverse a set of junction nodes

- 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

Rev 02 Changes

- Editorial Changes
  - Update Authors' credentials
  - Update references
    - Initial version of BGP signaling specification available

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

draft-kbr-teas-mptersvp-03

Kireeti Kompella, HPE

Vishnu Pavan Beeram, HPE

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 current focus 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
    - 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/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

- Introduce the concept of a "HOP version" to handle subgraph updates efficiently using an in-place update approach.
  - Each JUNCTION PHOP or NHOP affected in a subgraph update are associated with a new "HOP version".
    - Update relevant Objects/TLVs
  - The DAG version carried in the VERSION object remains unchanged during an in-place update.

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

draft-beeram-teas-yang-mpted-03

Vishnu Pavan Beeram, HPE

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      uint32
                +  ..
                +--ro junctions
                   +--ro junction* [node-id]
                      +--ro node-id                inet:ip-address
                      +  ..
                      +--ro phops
                      |  +--ro phop*
                      |           [hop-address hop-index hop-version]
                      |     +--ro hop-address     inet:ip-address
                      |     +--ro hop-index       uint32
                      |     +--ro hop-version     uint32
                      +  ..
                      +--ro nhops
                         +--ro nhop*
                                 [hop-address hop-index hop-version]
                            +--ro hop-address
                            |       inet:ip-address
                            +--ro hop-index              uint32
                            +--ro hop-version            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 programmed in the network.
  - Each tunnel entry may have more than one instance associated with it, where a unique version identifies each instance. Each tunnel instance has a list of junctions associated with it.
  - Each junction entry consists of the set of previous-hops ('phops' container) and next-hops ('nhops' container).

MPTED-JCT YANG Module: High-Level Model Structure

```
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               uint32
          +--rw sig-src                inet:ip-address
          +  ..
          +--rw phops
          |  +--rw phop* [hop-address hop-index hop-version]
          |     +--rw hop-address    inet:ip-address
          |     +--rw hop-index      uint32
          |     +--rw hop-version    uint32
          |     +  ..
          +--rw nhops
             +--rw nhop* [hop-address hop-index hop-version]
                +--rw hop-address            inet:ip-address
                +--rw hop-index              uint32
                +--rw hop-version?           uint32
                +  ..
```

- The top-level 'te' container is [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).

Rev 02/03 Changes

- Introduce the concept of a "HOP version" to handle subgraph updates efficiently using an in-place update approach.
  - Each JUNCTION PHOP or NHOP affected in a subgraph update are associated with a new "HOP version".
    - Update relevant containers.
  - The DAG version remains unchanged during an in-place update.

Next Steps

- To Do List
  - Architecture:
    - Add details on realizing MPTED tunnels over native IPv4/v6 forwarding plane
  - Signaling:
    - Add more signaling sequence examples
    - Add details on graceful restart procedures
  - Data Model:
    - Add more operational state
- Share implementation report at IETF126
  - Comparison with “Container Tunnel/Policy” based Multipath-TE approach
- Request WG adoption for the 3 drafts at IETF126

Thank You