

Yang Data Model for TE Topologies

draft-ietf-teas-yang-te-topo-08

Github: <https://github.com/ietf-mpls-yang/te/blob/master/ietf-te-topology.yang>

Xufeng Liu (Jabil)

Vishnu Pavan Beeram (Juniper Networks)

Igor Bryskin (Huawei Technologies)

Tarek Saad (Cisco)

Himanshu Shah (Ciena)

Oscar Gonzalez De Dios (Telefonica)

Contributors:

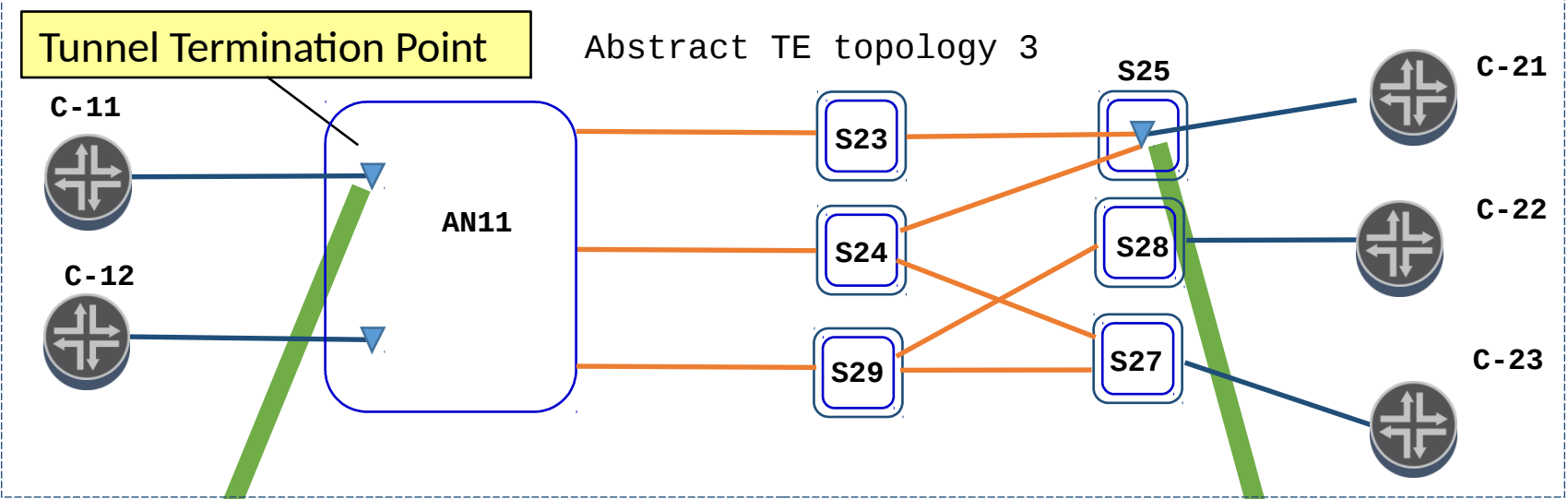
Sergio Belotti (Nokia)

Diete Beller (Nokia)

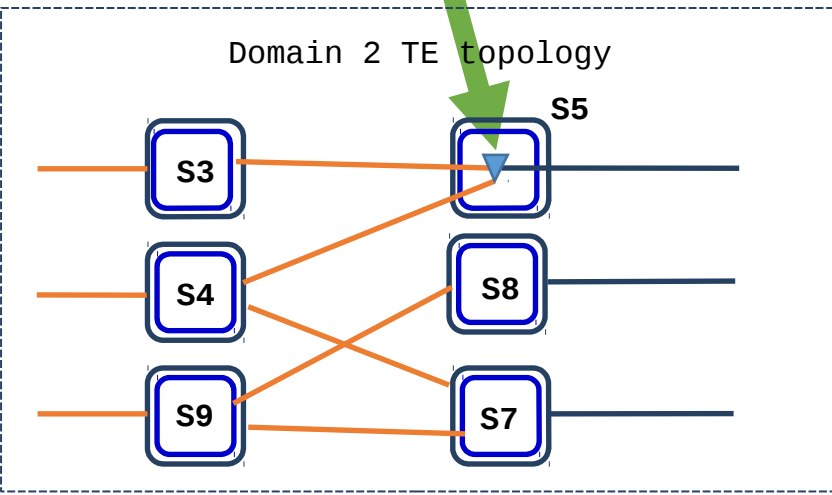
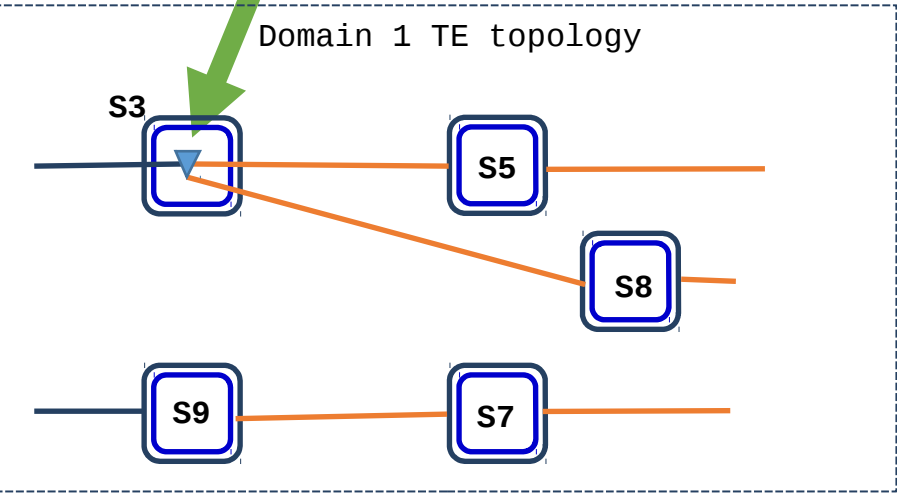
Summary of Changes

- Added modeling of supporting tunnel termination point.
- Added support for specifying default attributes of connectivity matrix and LLCL.
- Added NSRLG (Not Sharing Risk Link Group).
- Allowed configuration of one or more underlay tunnels.
- Provided an option to set underlay tunnel shared or dedicated.
- Improved client layer adaption to support more than two layers in a multi-layer topology.
- Clarified customized and abstract topologies in the draft text.

Supporting Tunnel Termination Point



Supported by



Supporting Tunnel Termination Point

- Each tunnel termination point may be supported by one or more tunnel termination points in the supporting node or a node in the underlay topology.

```
augment /nw:networks/nw:network/nw:node:
  +--rw te-node-id?    te-types:te-node-id
  +--rw te!
    +--rw tunnel-termination-point* [tunnel-tp-id]
      +--rw tunnel-tp-id                binary
      +--rw supporting-tunnel-termination-point*
        [node-ref tunnel-tp-ref]
          +--rw node-ref                inet:uri
          +--rw tunnel-tp-ref            binary
```

Connectivity Matrix

- Be consistent with other models, and support defaults.

```
augment /nw:networks/nw:network/nw:node:
  +--rw te!
    +--rw config
      +--rw te-node-attributes
        +--rw connectivity-matrices
          +--rw number-of-entries?      uint16
          +--rw is-allowed?             boolean
          +--rw label-restriction* [inclusive-exclusive label-start]
          +--rw underlay! {te-topology-hierarchy}?
          +--rw max-lsp-bandwidth* [priority]
          +--rw max-link-bandwidth?     te-bandwidth
          +--rw max-resv-link-bandwidth? te-bandwidth
          +--rw unreserved-bandwidth* [priority]
          +--rw te-default-metric?      uint32
          +--rw te-delay-metric?       uint32
          +--rw te-srlgs
          +--rw te-nsrlgs {nsrlg}?
        +--rw connectivity-matrix* [id]
          +--rw id                      uint32
          +--rw from
            | +--rw tp-ref?  leafref
          +--rw to
            | +--rw tp-ref?  leafref
          +--rw is-allowed?             boolean
          +--rw label-restriction* [inclusive-exclusive label-start]
          +--rw underlay! {te-topology-hierarchy}?
          +--rw max-lsp-bandwidth* [priority]
          +--rw max-link-bandwidth?     te-bandwidth
          +--rw max-resv-link-bandwidth? te-bandwidth
          +--rw unreserved-bandwidth* [priority]
          +--rw te-default-metric?      uint32
          +--rw te-delay-metric?       uint32
          +--rw te-srlgs
          +--rw te-nsrlgs {nsrlg}?
```

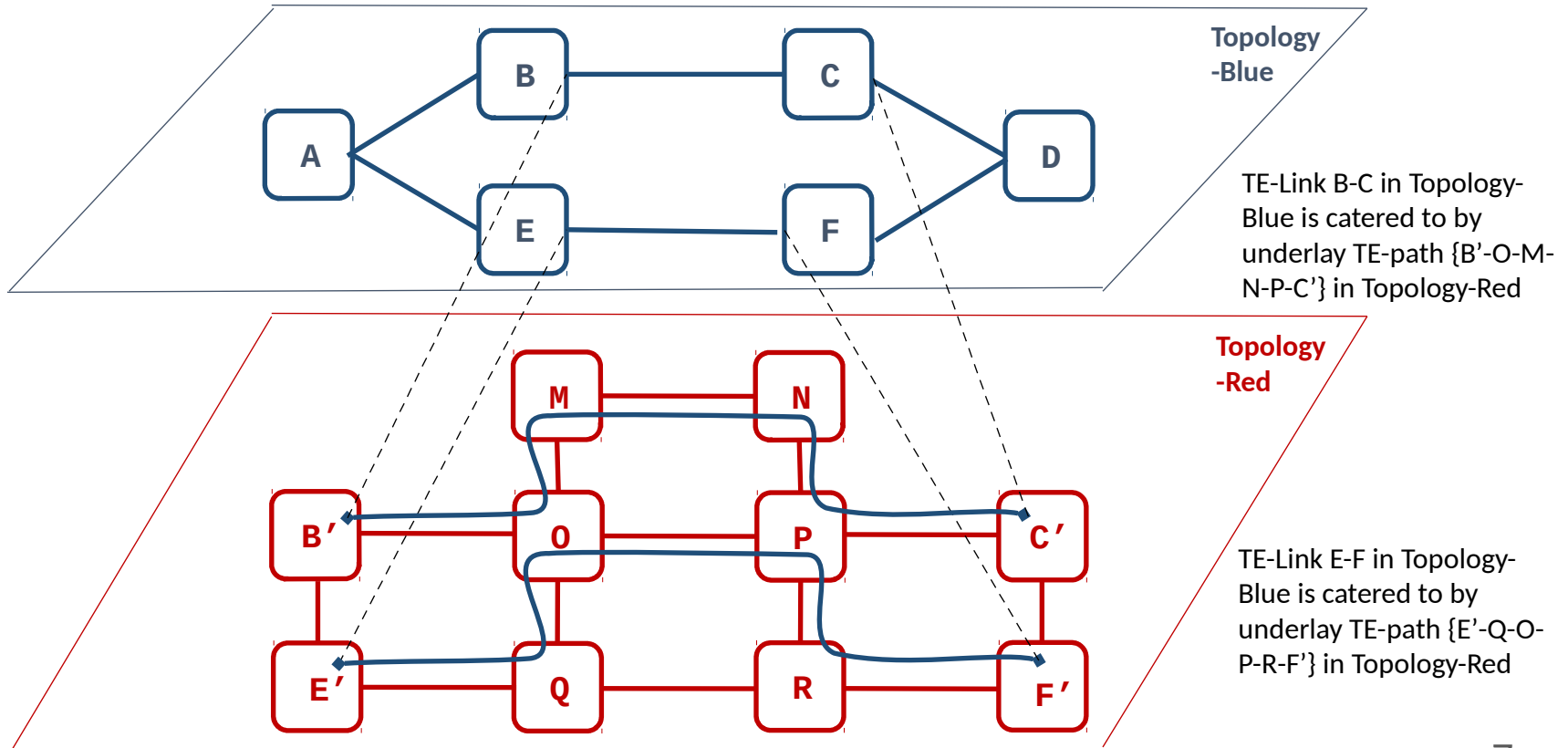
LLCL (Local Link Connectivity List)

- Same reasons as Connectivity Matrix.

```
augment /nw:networks/nw:network/nw:node:
+--rw te-node-id?   te-types:te-node-id
+--rw te!
  +--rw tunnel-termination-point* [tunnel-tp-id]
    +--rw config
      |  +--rw local-link-connectivities
      |  |  +--rw number-of-entries?      uint16
      |  |  +--rw is-allowed?             boolean
      |  |  +--rw label-restriction* [inclusive-exclusive label-start]
      |  |  +--rw underlay! {te-topology-hierarchy}?
      |  |  +--rw max-lsp-bandwidth* [priority]
      |  |  +--rw max-link-bandwidth?      te-bandwidth
      |  |  +--rw max-resv-link-bandwidth? te-bandwidth
      |  |  +--rw unreserved-bandwidth* [priority]
      |  |  +--rw te-default-metric?       uint32
      |  |  +--rw te-delay-metric?        uint32
      |  |  +--rw te-srlgs
      |  |  +--rw te-nsrlgs {nsrlg}?
      |  +--rw local-link-connectivity* [link-tp-ref]
      |  |  +--rw link-tp-ref              leafref
      |  |  +--rw is-allowed?              boolean
      |  |  +--rw label-restriction* [inclusive-exclusive label-start]
      |  |  +--rw underlay! {te-topology-hierarchy}?
      |  |  +--rw max-lsp-bandwidth* [priority]
      |  |  +--rw max-link-bandwidth?      te-bandwidth
      |  |  +--rw max-resv-link-bandwidth? te-bandwidth
      |  |  +--rw unreserved-bandwidth* [priority]
      |  |  +--rw te-default-metric?       uint32
      |  |  +--rw te-delay-metric?        uint32
      |  |  +--rw te-srlgs
      |  |  +--rw te-nsrlgs {nsrlg}?
```

NSRLG(Not Sharing Risk Link Group)

- When an abstract TE link is configured, client can request the TE link to be mutually diverse disjoint with other TE links in the same topology
 - Opposite to SRLG.
 - e.g. TE Links B-C and E-F are diverse at link-disjoint level



NSRLG Modeling

■ On TE Link

```
augment /nw:networks/nw:network/nt:link:
  +--rw te!
    +--rw config
      |   +--rw te-link-attributes
      |     +--rw is-abstract?                empty
      |     +--rw underlay! {te-topology-hierarchy}?
      |       .....
      |     +--rw te-srlgs
      |       +--rw value*    te-types:srlg
      |       +--rw te-nsrlgs {nsrlg}?
      |       +--rw id*      leafref
    +--ro state
      |   +--ro te-link-attributes
      |     |   +--ro te-nsrlgs {nsrlg}?
      |     |   +--ro id*      leafref
```

■ NSRLG Definition

```
augment /nw:networks/nw:network:
  +--rw provider-id?    te-types:te-global-id
  +--rw client-id?      te-types:te-global-id
  +--rw te-topology-id? te-types:te-topology-id
  +--rw te!
    +--rw config
      |   +--rw preference?          uint8
      |   +--rw optimization-criterion? identityref
      |   +--rw nsrlg* [id] {nsrlg}?
      |     +--rw id                uint32
      |     +--rw disjointness?    bits {node, link, srlg}
```

Underlay Tunnel Configuration

- A TE link can be requested to utilize one or more existing underlay TE tunnels.
- The underlay TE tunnels can be looked up by tunnel names in the TE tunnel model.
- The tunnel names of the underlay TE tunnels are specified in this model.

```
augment /nw:networks/nw:network/nt:link:
  +--rw te!
    +--rw config
      |   +--rw te-link-attributes
      |     +--rw underlay! {te-topology-hierarchy}?
      |       |   +--rw primary-path
      |       |   +--rw backup-path* [index]
      |       |   +--rw protection-type?   uint16
      |       |   +--rw tunnels
      |       |     +--rw tunnel* [tunnel-name]
      |       |     +--rw tunnel-name      string
```

Underlay Tunnel Sharing

- An underlay TE tunnels may or may not be shared by multiple TE links.
- The operator can request an underlay TE tunnels to shared or dedicated.

```
augment /nw:networks/nw:network/nt:link:
  +--rw te!
    +--rw config
      |   +--rw te-link-attributes
      |   |   +--rw underlay! {te-topology-hierarchy}?
      |   |   |   +--rw primary-path
      |   |   |   +--rw backup-path* [index]
      |   |   |   +--rw protection-type?   uint16
      |   |   |   +--rw tunnels
      |   |   |   |   +--rw sharing?      boolean
      |   |   |   |   +--rw tunnel* [tunnel-name]
      |   |   |   |   |   +--rw tunnel-name      string
      |   |   |   |   |   +--rw sharing?      boolean
```

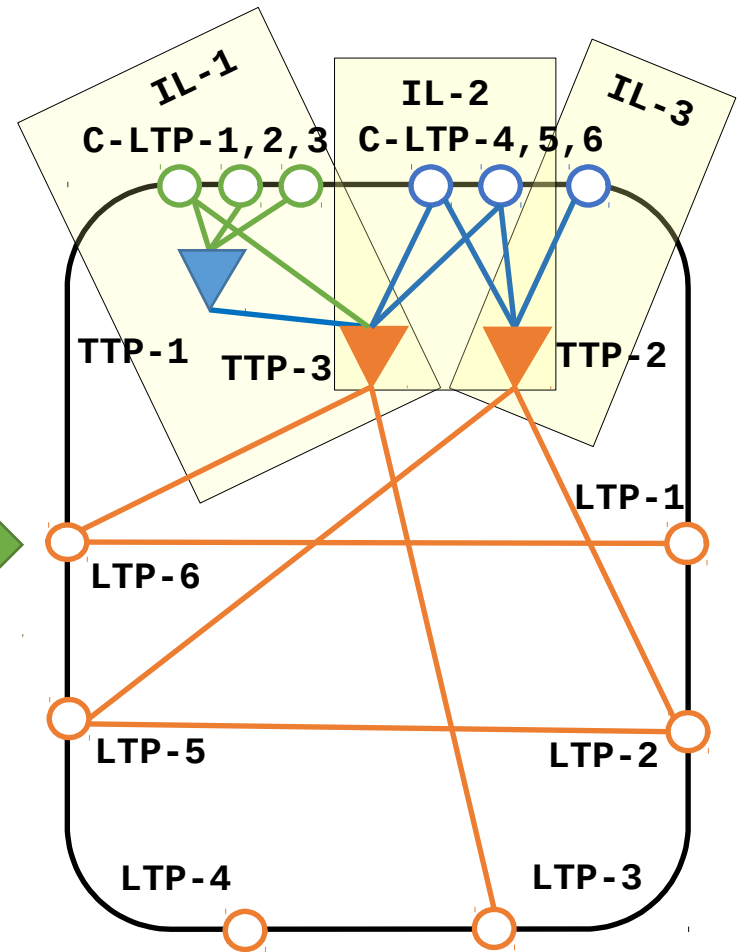
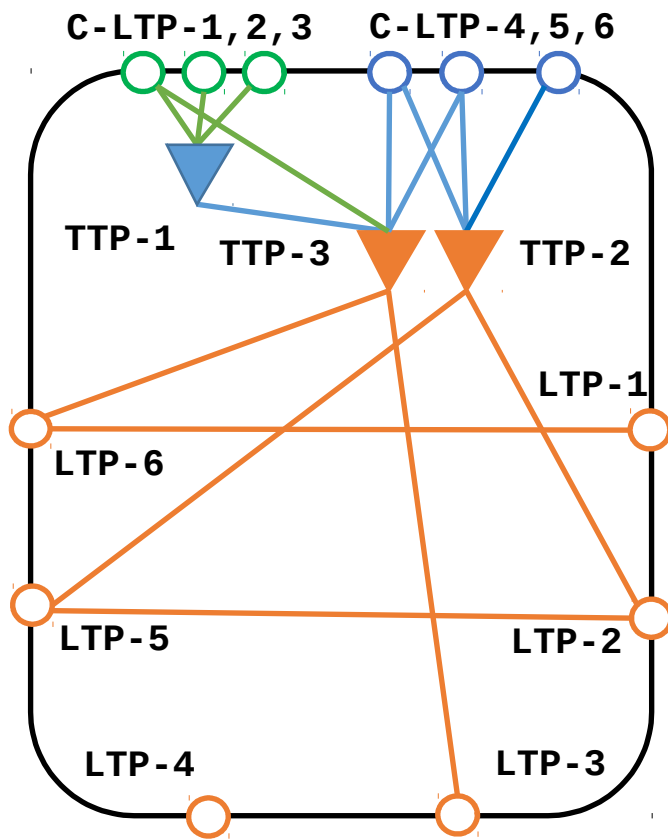
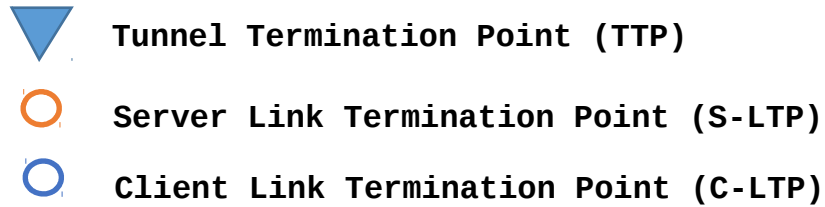
- Transitional link approach

-
- The diagram illustrates the proposed network architecture, showing the connections between different components:
- C-LTP-1, 2, 3** (Green circles) and **C-LTP-4, 5, 6** (Blue circles) are connected to **TTP-1** (Blue triangle) and **TTP-3** (Blue triangle).
 - TTP-1** and **TTP-3** are connected to **TTP-2** (Blue triangle).
 - TTP-2** is connected to **LTP-1** (Orange circle), **LTP-2** (Orange circle), **LTP-3** (Orange circle), **LTP-4** (Orange circle), **LTP-5** (Orange circle), and **LTP-6** (Orange circle).
 - LTP-1**, **LTP-2**, **LTP-3**, **LTP-4**, **LTP-5**, and **LTP-6** are connected to each other in a mesh-like structure.



Multi-layer Topology

■ Inter-layer lock approach



Multi-layer Topology Modeling

- Added client layer adaptation information to tunnel termination point.

```
augment /nw:networks/nw:network/nw:node:
  +--rw te-node-id?    te-types:te-node-id
  +--rw te!
    +--rw tunnel-termination-point* [tunnel-tp-id]
      +--rw tunnel-tp-id                                binary
      +--rw config
        | +--rw switching-capability?                    identityref
        | +--rw encoding?                                identityref
        | +--rw inter-layer-lock-id?                     uint32
        | +--rw protection-type?                         identityref
        | +--rw client-layer-adaptation
        | | +--rw switching-capability*
        | | | [switching-capability encoding]
        | | | +--rw switching-capability                identityref
        | | | +--rw encoding                            identityref
        | | | +--rw bandwidth?                          te-bandwidth
```

Customized and Abstract Topologies

- Abstract TE topology
 - contains abstract topological elements (nodes, links, tunnel termination points).
- Customized TE topology
 - was modified by the provider to honor a particular client's requirements or preferences.
- A customized TE topology is not necessarily an abstract TE topology; a customized TE topology may not be abstract.
- The client ID field indicates which client the TE topology is customized for.
- A client may receive a TE topology
 - with any client ID field
- A client can customize a TE topologie
 - with matching the ID, or
 - client ID == 0.

Next Steps

- Align to potential change in draft-ietf-i2rs-yang-network-topo
 - Use revised datastores.
 - No impact to TE topology model.

```
module: ietf-network
  +--rw networks
    +--rw network* [network-id]
      +--rw network-types
      +--rw network-id network-id
      +--ro server-provided? Boolean
      +--rw supporting-network* [network-ref]
        | +--rw network-ref -> /networks/network/network-id
      +--rw node* [node-id]
        | +--rw node-id node-id
        | +--rw lnk:termination-point* [tp-id]
        |   +--rw lnk:tp-id tp-id
      +--rw lnk:link* [link-id]
        +--rw lnk:source
          | +--rw lnk:source-node? leafref
          | +--rw lnk:source-tp? leafref
        +--rw lnk:destination
          | +--rw lnk:dest-node? leafref
          | +--rw lnk:dest-tp? leafref
```

Next Steps

- Request further review
- WG last call

Yang Data Model for Layer 3 TE Topologies

draft-liu-teas-yang-l3-te-topo-03

Xufeng Liu (Jabil)

Vishnu Pavan Beeram (Juniper Networks)

Igor Bryskin (Huawei Technologies)

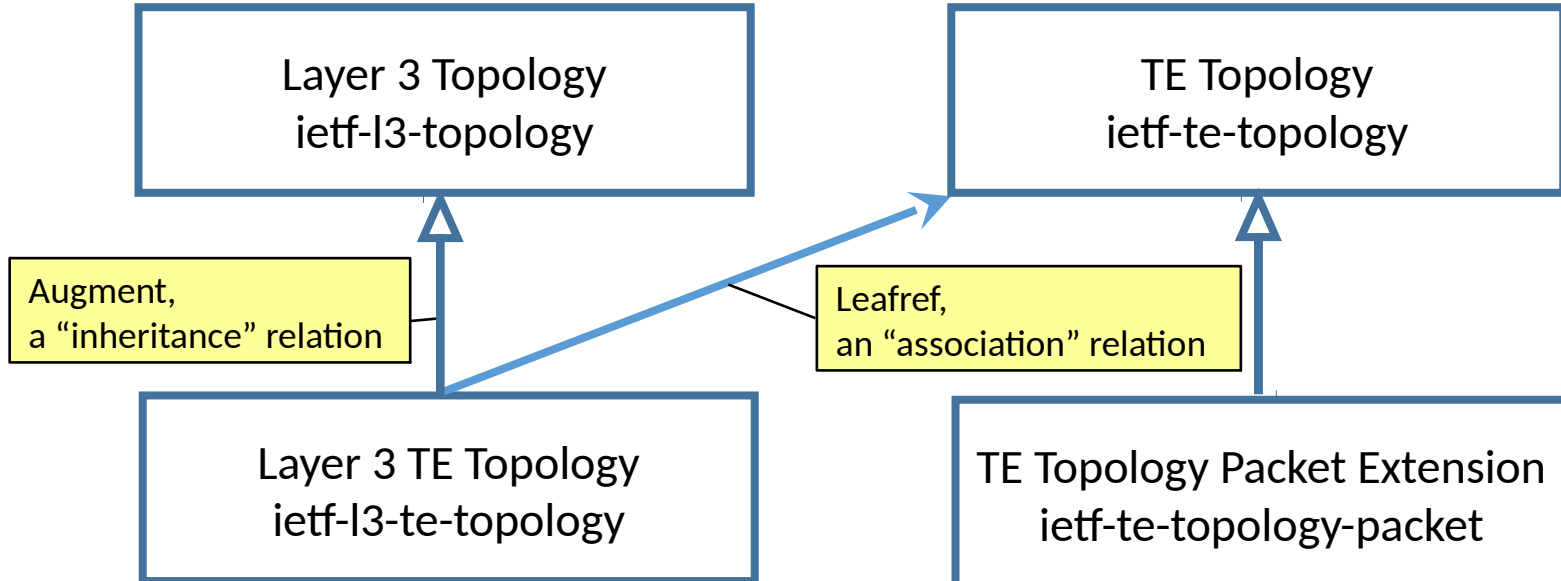
Tarek Saad (Cisco)

Himanshu Shah (Ciena)

Oscar Gonzalez De Dios (Telefonica)

Augmentation Hierarchy

- L3 TE Topology augments L3 Topology and references TE Topology.
- Packet extension module augments ietf-te-topology.



Model Reorganizations

- Module ietf-te-topology-packet contains packet specific augmentations to ietf-te-topology.
- Moved the performance metric related groupings to generic TE types module.
- Imported ietf-routing-types module to use common types.

Next Steps

- Request further review.
- Ask for WG adoption.

Yang Data Model for SR and SR TE Topologies

draft-liu-teas-yang-sr-te-topo-02

Xufeng Liu (Jabil)

Vishnu Pavan Beeram (Juniper Networks)

Igor Bryskin (Huawei Technologies)

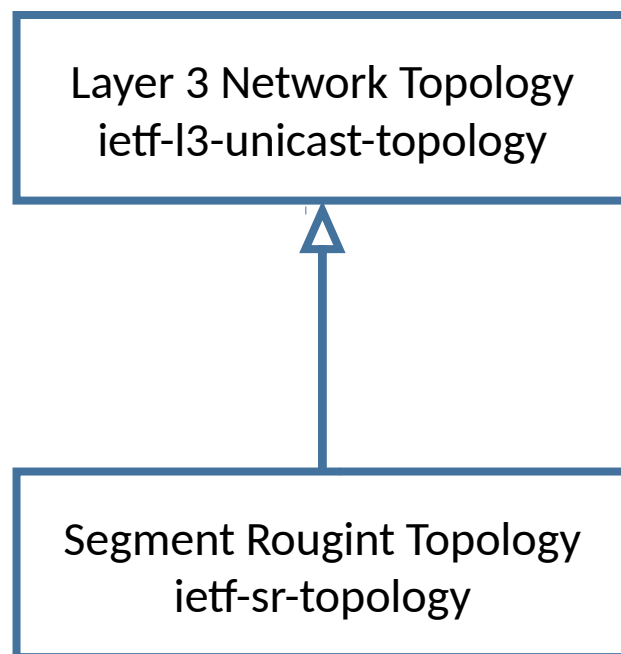
Tarek Saad (Cisco)

Himanshu Shah (Ciena)

Stephane Litkowski (Orange)

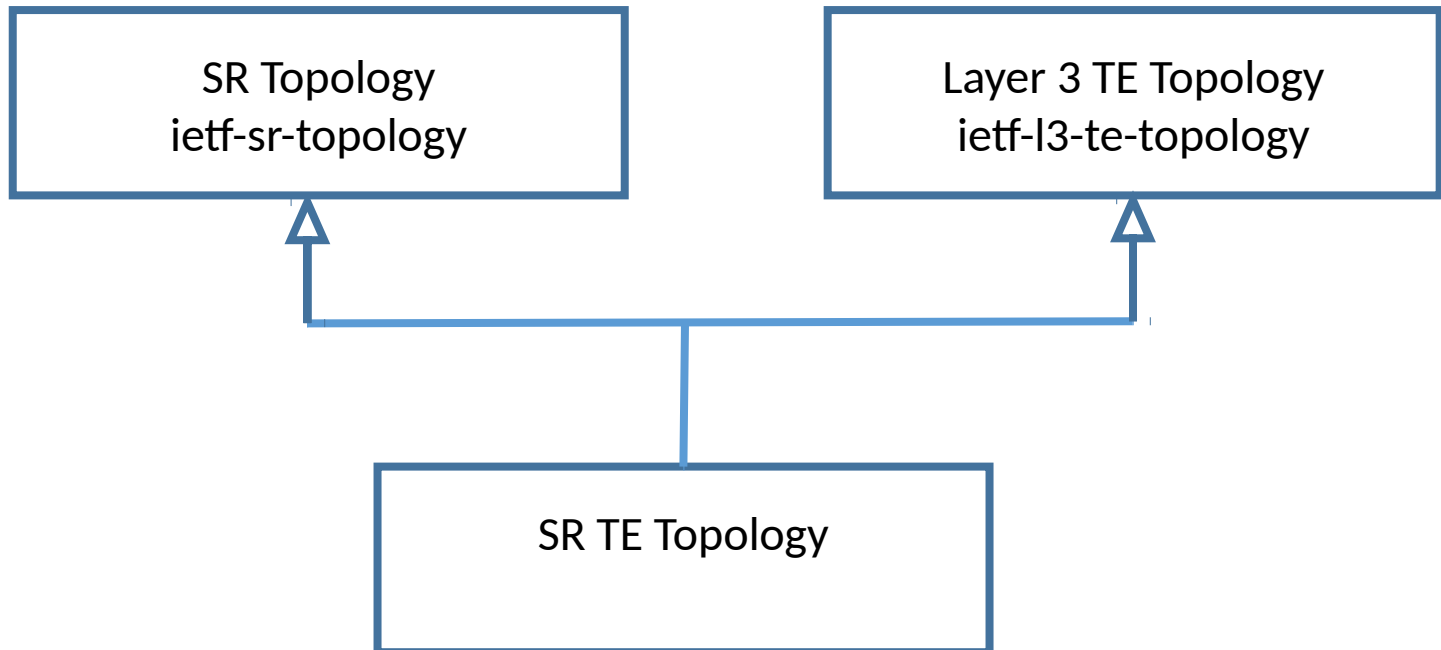
SR (Segment Routing) Topology

- Augment layer 3 network topology model



SR (Segment Routing) TE Topology

- Multiple inheritance:
 - Is both SR topology and layer 3 TE topology model.
 - Uses multiple network types: “l3-te” and “sr”.



Collaboration with SR Yang Model

- Worked with authors of draft-ietf-spring-sr-yang
 - Updated to synch up with latest ietf-segment-routing-common.yang
 - Added support for SRLB

```
augment /nw:networks/nw:network/nw:node/l3t:l3-node-attributes:
  +--rw sr
    +--rw config
      |   +--rw srgb* [lower-bound upper-bound]
      |   |   +--rw lower-bound    uint32
      |   |   +--rw upper-bound    uint32
      |   |   +--rw srlb* [lower-bound upper-bound]
      |   |   |   +--rw lower-bound    uint32
      |   |   |   +--rw upper-bound    uint32
      |   +--rw node-capabilities
      |       +--rw transport-planes* [transport-plane]
      |       |   +--rw transport-plane    identityref
      |       +--rw readable-label-stack-depth?    uint8
```

Next Steps

- Request further review.
- Ask for WG adoption.