

# Profiles for Traffic Engineering (TE) Topology Data Model and Applicability to non-TE-centric Use Cases

TEAS WG, IETF 125, Shenshen

**draft-ietf-teas-te-topology-profiles-05**

## **Authors:**

[Italo Busi](#) (Huawei)

Xufeng Liu (Alef Edge)

Igor Bryskin

Tarek Saad (Cisco)

Oscar Gonzalez de Dios (Telefonica)

## **Contributors:**

Aihua Guo (Futurewei)

Haomian Zheng (Huawei)

Sergio Belotti (Nokia)

# Motivation

- Multiple similar discussions in IETF working groups
  - Scenario: TE-aware but non-TE-centric (“non-TE”) networks
  - Requirement: sub-set TE Topology attributes
- TE Topology Model (RFC8795) looks very complex at the first glance
  - Extensive model to support many features
    - Some applicable only to TE-centric networks
    - Others applicable to both TE-centric and non-TE-centric networks
  - Most of the features/attributes are optional
- Clarify that a sub-set (profile) of TE Topology can be used in specific scenarios
  - Including non-TE-centric scenarios

# Examples of generic profiles

- Multi-domain Link Discovery
- Administrative and Operational State
- Overlay and Underlay Topology
- Nodes with switching limitations
- Multipoint links

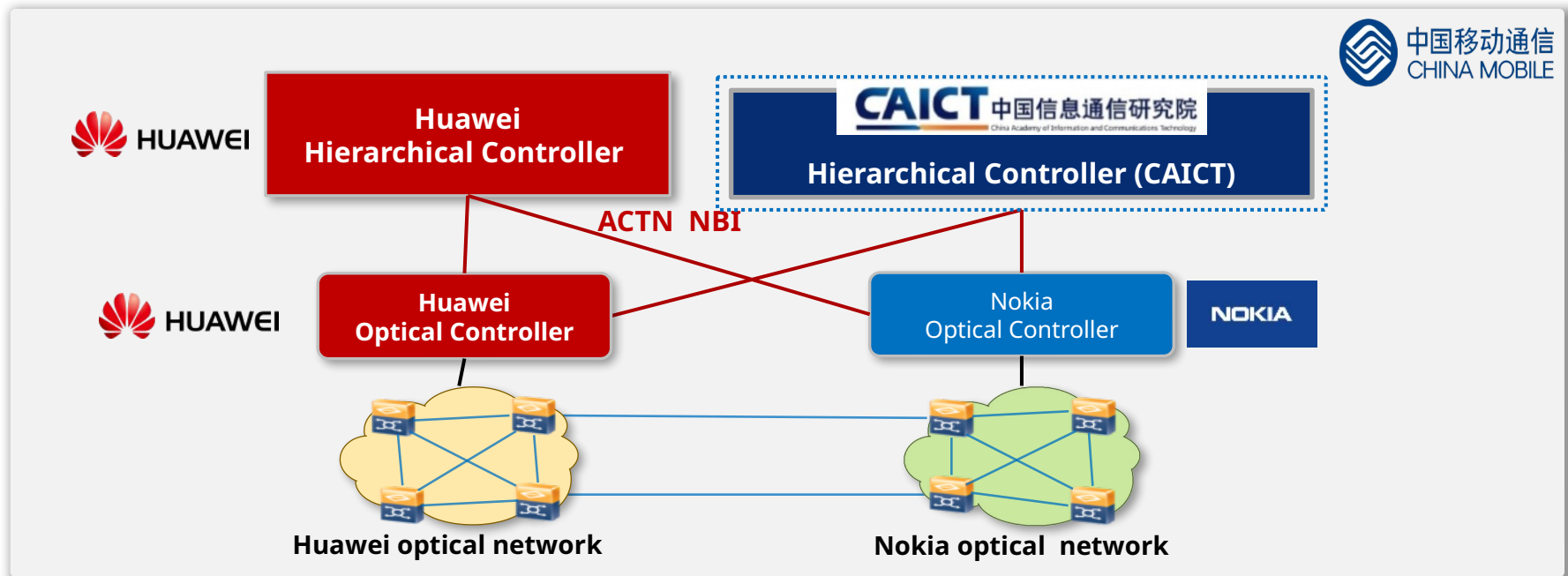
# Updates since IETF 124

- Renamed UNI Topology Discovery as Multi-domain Link discovery
  - Based on offline feedbacks after IETF 124 presentation
- Re-emphasized that the boundary between TE and non-TE is blurred
  - See section 1.2 of RFC9522
- Added references to existing public implementations
  - Note: approach already endorsed in RFC9656 (MW topology model)
- Editorial updates and figures clean-up

# Related Work

- New I-D submitted to NMOP WG
  - draft-busi-nmop-simap-rfc8795-applicability
- These two drafts are serving different purposes
  - This I-D clarifies that the RFC8795 model can be profiled for TE-centric and non-TE-centric applications
  - The SIMAP I-D describes how RFC8795 can be profiled to support SIMAP requirements

# ACTN multi-vendor interoperability tests (September 2017)



Reference: <https://ieeexplore.ieee.org/document/8334928>

# ETSI MW Plugtests (January 2019 and March 2022)

## Microwave and Ethernet Topology Discovery

[TE-Topo]

[MW-Topo]

[ETH-TE-Topo]

### Network and Service Discovery

- Vendors: Huawei, Ericsson, Nokia, NEC, SIAE, Ceragon, Intracom (>90% market)
- Operators: Vodafone, DT, BT, Orange

## References:

[https://portal.etsi.org/Portals/0/TBpages/CTI/Docs/mWT\\_Plugtest1\\_TestPlan\\_v1.0.pdf](https://portal.etsi.org/Portals/0/TBpages/CTI/Docs/mWT_Plugtest1_TestPlan_v1.0.pdf)

[https://portal.etsi.org/Portals/0/TBpages/CTI/Docs/mWT\\_Plugtests2-3\\_TestPlan\\_v1\\_0.pdf](https://portal.etsi.org/Portals/0/TBpages/CTI/Docs/mWT_Plugtests2-3_TestPlan_v1_0.pdf)

# Open Issue

- How to report to the client the profiles implemented by a server
  - May be different for different instances/network types
- For existing public implementations, a manually pruned/profiled YANG tree was sufficient
  - YANG deviation statements can also be used to programmatically generate pruned/profiled YANG tree
  - What else would be needed?
- Proposal
  - Tooling issue: out of scope from this draft
- See: <https://github.com/tsaad-dev/te/issues/161>

# Next Steps

- Content pretty stable
  - No pending open issue in the scope
  - Editorial clean-up
  - Ready for WG LC
- Trigger work on the tooling issue
- Github: <https://github.com/tsaad-dev/te>

Backup

# Multi-domain Link Discovery

```
module: ietf-te-topology
  augment /nw:networks/nw:network/nw:network-types:
    +-rw te-topology!
  augment /nw:networks/nw:network/nw:node/nt:termination-point:
    +-rw te-tp-id?    te-types:te-tp-id
    +-rw te!
      +-rw admin-status?
        |      te-types:te-admin-status
      +-rw inter-domain-plug-id?      binary
      +-ro oper-status?                  te-types:te-oper-status
```

# Administrative and Operational State

```
module: ietf-te-topology
augment /nw:networks/nw:network/nw:network-types:
  +--rw te-topology!
augment /nw:networks/nw:network:
  +--rw te-topology-identifier
  | +--rw provider-id?   te-global-id
  | +--rw client-id?    te-global-id
  | +--rw topology-id?  te-topology-id
  +--rw te!
    +--rw name?          string
augment /nw:networks/nw:network/nw:node:
  +--rw te-node-id?    te-types:te-node-id
  +--rw te!
    +--rw te-node-attributes
    | +--rw admin-status?  te-types:te-admin-status
    | +--rw name?          string
    +--ro oper-status?    te-types:te-oper-status
augment /nw:networks/nw:network/nt:link:
  +--rw te!
    +--rw te-link-attributes
    | +--rw name?          string
    | +--rw admin-status?  te-types:te-admin-status
    +--ro oper-status?    te-types:te-oper-status
augment /nw:networks/nw:network/nw:node/nt:termination-point:
  +--rw te-tp-id?     te-types:te-tp-id
  +--rw te!
    +--rw admin-status?  te-types:te-admin-status
    +--rw name?          string
    +--ro oper-status?   te-types:te-oper-status
```

# Overlay and Underlay Topology

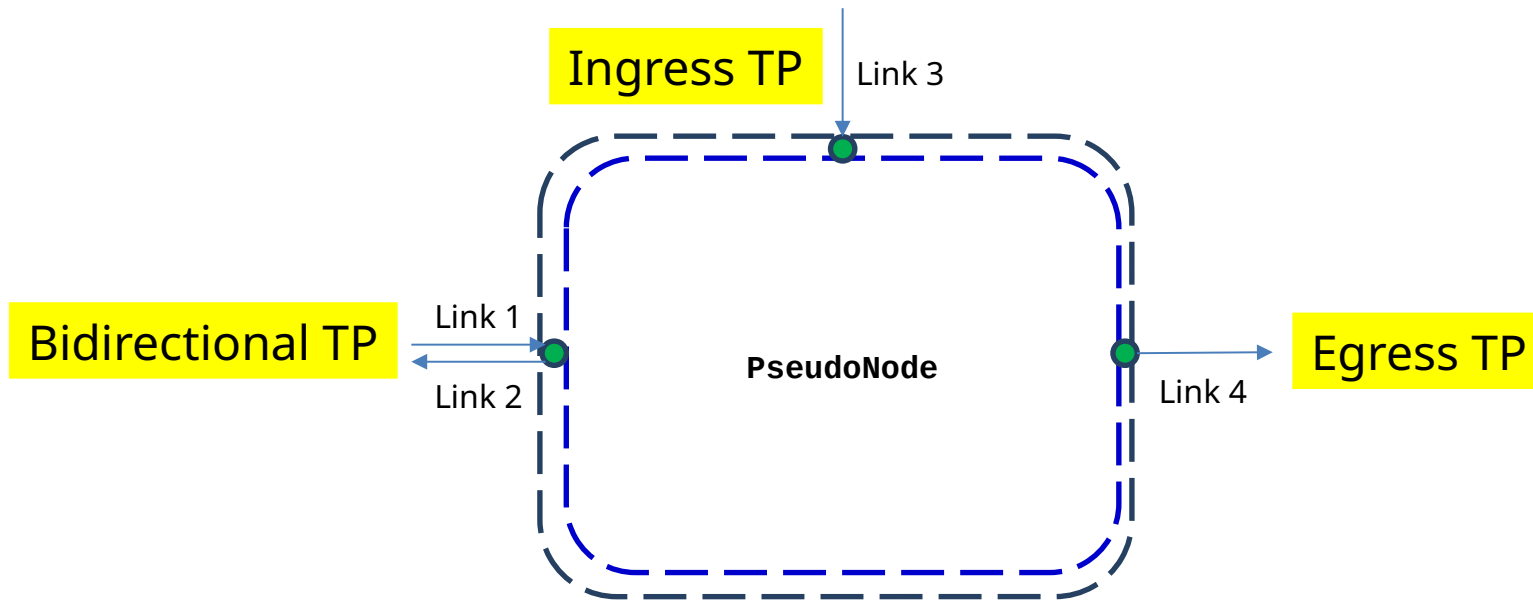
```
module: ietf-te-topology
augment /nw:networks/nw:network/nw:network-types:
  +--rw te-topology!
augment /nw:networks/nw:network/nw:node:
  +--rw te-node-id?   te-types:te-node-id
  +--rw te!
    +--rw te-node-attributes
      +--rw underlay-topology {te-topology-hierarchy}?
        +--rw network-ref?   -> /nw:networks/network/network-id
augment /nw:networks/nw:network/nt:link:
  +--rw te!
    +--rw te-link-attributes
      +--rw underlay {te-topology-hierarchy}?
        +--rw enabled?          boolean
        +--rw primary-path
          +--rw network-ref?
            | -> /nw:networks/network/network-id
          +--rw path-element* [path-element-id]
            +--rw path-element-id      uint32
            +--rw (type)?
              +--:(numbered-link-hop)
                | +--rw numbered-link-hop
                |   +--rw link-tp-id   te-tp-id
                |   +--rw hop-type?   te-hop-type
                |   +--rw direction?  te-link-direction
              +--:(unnumbered-link-hop)
                +--rw unnumbered-link-hop
                  +--rw link-tp-id   te-tp-id
                  +--rw node-id      te-node-id
                  +--rw hop-type?   te-hop-type
                  +--rw direction?  te-link-direction
```

# Nodes with switching limitations

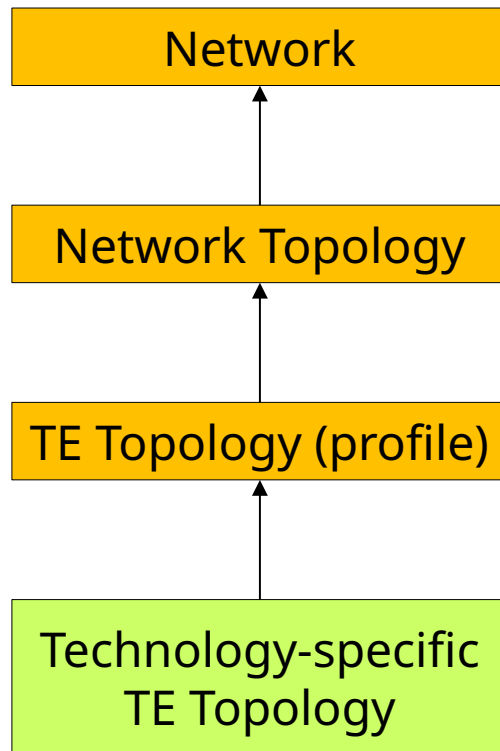
```
module: ietf-te-topology
  augment /nw:networks/nw:network/nw:network-types:
    +--rw te-topology!
  augment /nw:networks/nw:network/nw:node:
    +--rw te-node-id?   te-types:te-node-id
    +--rw te!
      +--rw te-node-attributes
        +--rw connectivity-matrices
          +--rw number-of-entries?      uint16
          +--rw is-allowed?             boolean
          +--rw connectivity-matrix* [id]
            +--rw id                    uint32
            +--rw from
              | +--rw tp-ref?           leafref
            +--rw to
              | +--rw tp-ref?           leafref
            +--rw is-allowed?           boolean
```

# Multipoint Link

- Defined in [draft-ietf-nmop-simap-concept-07](#)
- Modelled as a pseudonode
  - Defined section 4.4.5 of RFC 8345
  - Connectivity constraints specified using Connectivity Matrix, defined in RFC 8795
  - Directionality of the TPs defined by the terminated Links

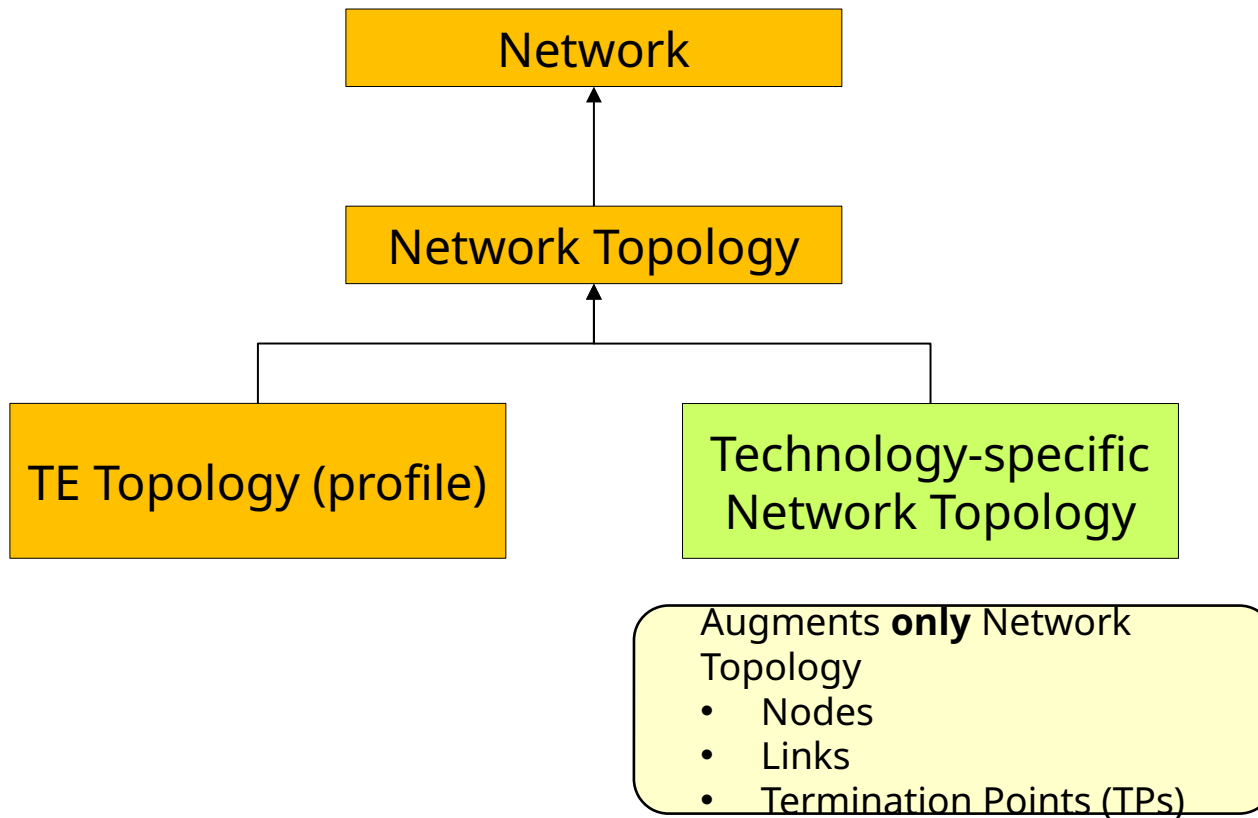


# Technology-specific Augmentations Option 1

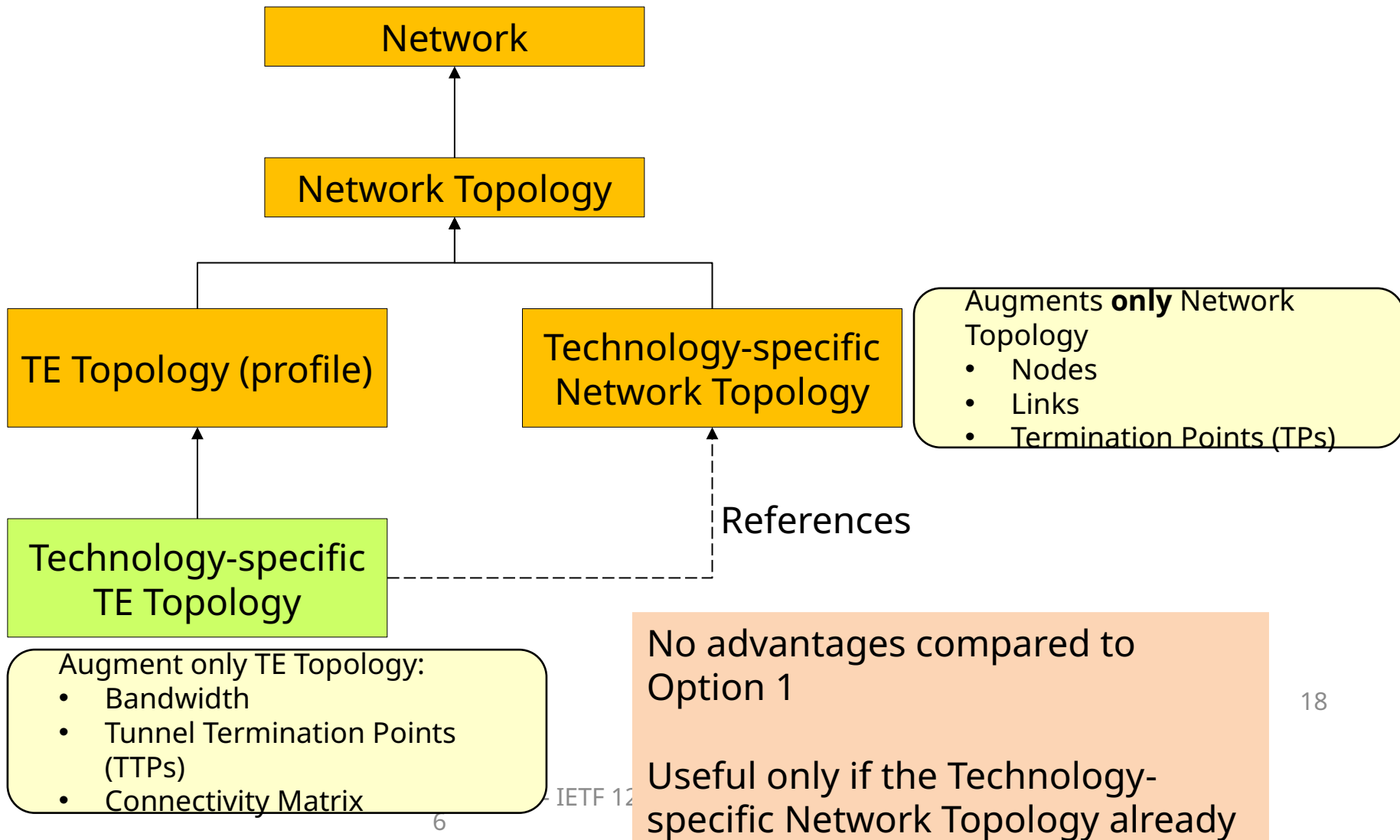


- Augments Network Topology:
- Nodes
  - Links
  - Termination Points (TPs)
- Augment **also** TE Topology:
- Bandwidth
  - Tunnel Termination Points (TTPs)
  - Connectivity Matrix

# Technology-specific Augmentations Option 2



# Technology-specific Augmentations Option 3



# Example: Technology-specific Augmentations (Link)

```
+--rw link* [link-id]
  +--rw link-id          link-id
  <...>
  +--rw example-link-attributes // augment TE (Option 1) OR
  |   <...> // augment NT (Option 2 or 3)
+--rw te!
  +--rw te-link-attributes
    +--rw name?          string
    +--rw example-te-link-attributes // augment TE (Option 1 or 3)
    |   <...>
    +--rw max-link-bandwidth
      +--rw te-bandwidth
        +--rw (technology)?
          +--:(generic)
            |   +--rw generic?    te-bandwidth
          +--:(foo) // augment TE (Option 1 or 3)
            +--rw foo?          foo-bandwidth
```