# Yang model for requesting Path Computation

draft-ietf-teas-yang-path-computation-10

IETF 108 – Virtual Meeting

Italo Busi (Huawei)
Sergio Belotti (Nokia)
Daniele Ceccarelli (Ericsson)
Victor Lopez, Oscar Gonzales de Dios (Telefonica)
Michael Scharf (Individual)
Anurag Sharma (Google)
Yan Shi (China Unicom)
Ricard Vilalta (CTTC)
Karthik Sethuraman (NEC)

# Summary of changes from v07

- Major changes to the YANG model
  - path computation for protected paths (as discussed in IETF 104)
  - path computation for bidirectional paths
  - errors in case path computation fails
  - multi-layer path computation
- Addressed comments from TEAS WG
  - Thanks to Haomian Zheng, Yanlei Zheng and Tom Petch
- Changes described in new sections 5.3 and 5.4

# Basic Yang tree concept (1)

```
+-- path-request* [request-id]
|  +-- request-id                          uint32
|  +-- (tunnel-attributes)?
|  |  +--:(reference)
|  |  |  +-- (tunnel-exist)?
|  |  |  |  +--:(tunnel-ref)
|  |  |  |  |  +-- tunnel-ref              te:tunnel-ref
|  |  |  |  +--:(tunnel-attributes-ref)
|  |  |  |     +-- tunnel-attributes-ref    leafref
|  |  |  .....
|  |  +--:(value)
|  |     +-- tunnel-name?                  string
|  |     .....
|  .....
+-- tunnel-attributes* [tunnel-name]
|  +-- tunnel-name              string
|  .....
.....
```

- The `(reference)` case, references either
  - existing tunnel: e.g., when computing the protection path to add protection for an existing tunnel
  - an entry to the new `tunnel-attributes` list when computing multiple paths for a tunnel that does not exist yet: tunnel attributes (e.g. tunnel-name, source/destination TTP, encoding and switching-type) are provided here
- The `(reference)` case also provides information about the role and direction of the path being requesting within tunnel (primary or secondary, forward or reverse)

# Basic Yang tree concept (2)

```
+-- path-request* [request-id]
|  +-- request-id                          uint32
|  +-- (tunnel-attributes)?
|  |  +--:(reference)
|  |  |  ...........
|  |  +--:(value)
|  |     +-- tunnel-name?                  string
|  |     ...........
|  ...........
...........
```

- The (`value`) case provides the set of tunnel attributes (e.g. tunnel-name, source/destination TTP, encoding and switching-type) in case there is no need to associated multiple path requests (e.g., path computation for an unprotected tunnel which does not exist yet)

- The server will have all the information to know how to create a tunnel within the operational DS,  when requested (alignment with the tunnel model is strengthened)

# Basic Yang tree concept (3)

- Support of multi-layer path computation based on the same approach with dependency tunnels as in [TE-TUNNEL] draft
  - Dependency tunnel is either already present in the datastore in the tunnel-attributes list

```
+-- dependency-tunnels
|   +-- dependency-tunnel* [name]
|   |   +-- name
|   |   |       -> ../../../../../../tunnels/tunnel/name
|   |   +-- encoding?         identityref
|   |   +-- switching-type?   identityref
|   +-- dependency-tunnel-attributes* [name]
|   |   +-- name              leafref
|   |   +-- encoding?         identityref
|   |   +-- switching-type?   identityref
```

- Server-layer tunnel should provide the information regarding the dynamic link in the client layer topology supported by that tunnel.

```
+-- hierarchical-link
   +-- local-te-node-id?         te-types:te-node-id
   +-- local-te-link-tp-id?      te-types:te-tp-id
   +-- remote-te-node-id?        te-types:te-node-id
   +-- te-topology-identifier
      +-- provider-id?   te-global-id
      +-- client-id?     te-global-id
      +-- topology-id?   te-topology-id
```

# Open Issues status

- GitHub Repository
  https://github.com/rvilalta/ietf-te-path-computation
- Tracking Open Issues, discussions and resolutions linked to YANG model
  - Still 9 open, 5 specific for path computation RPC
    - 1 pending feedbacks from YANG experts (#76)
    - 2 editorial (#77: hop cumulative metric, #58: review terminology)
    - 2 pending the YANG model becoming stable (#75: security, #40: example)

# YANG Open Issue (#76, 1)

```
+-- path-request* [request-id]
|  +-- request-id                            uint32
|  ...........
|  |  |  |     +-- tunnel-attributes-ref     leafref
|  ...........
+-- tunnel-attributes* [tunnel-name]
|  +-- tunnel-name               string
|  ...........
...........
```

- How to specify the leafrefs within RPC?

- This code compiles with pyang 1.7.5 while it fails with pyang 2.1:

```
leaf tunnel-attributes-ref {
  type leafref {
    path "/te:tunnels-path-compute/te:input/"
       + "te:path-compute-info/"
       + "te-pc:tunnel-attributes/te-pc:tunnel-name";
}
```

- This code compiles with pyang 2.1 while it fails with pyang 1.7.5:

```
leaf tunnel-attributes-ref {
  type leafref {
    path "/te:tunnels-path-compute/"
       + "te:path-compute-info/"
       + "te-pc:tunnel-attributes/te-pc:tunnel-name";
}
```

# YANG Open Issue (#76, 2)

- How to condition data definition in the RPC output based on the RPC input?

- This when statement compiles only with pyang 1.7.5 while it fails with pyang 2.1

```
augment "/te:tunnels-actions/te:output" {
      container path-computed-delete-result {
        when "derived-from-or-self(../../te:input/te:action-info/"
           + "te:action, 'tunnel-action-path-compute-delete')";
      }   // container path-computed-delete-result
    }   // path-delete rpc output
```

# Next Steps

- Ready for YANG doctor review
- Resolve few open issues
  - Continue cooperation with TE Tunnel model authors
- Plan for WG LC after IETF 109