Network Working Group PC. Fu
Internet-Draft JZ. FU
Intended status: Standards Track Huawei Technologies
Expires: May 1, 2017 AJ. Wang
RQ. Jing
China Telecom
October 28, 2016

A YANG Model for VNT (IP Link) Protection Group
draft-fu-pce-vnt-protection-group-01

Abstract

IP+optical is a cross-layer collaboration technology for unified management of IP and optical networks.

Based on framework proposed in [ACTN-FWK][I-D.ietf-teas-actn-framework], this draft presents specific information about the IP+optical solution: hierarchical controllers + disabled GMPLS UNIs. This solution does not involve UNI tunnel objects.

The system uses loose-coupled dual-controllers to implement cross-layer collaborative path calculation on the virtual network topology (VNT), where the VNT provides the E2E cross-pathcalculation bridging function. The VNT needs to be defined as a YANG model for configuration management.

This draft provides a YANG model for the RESTCONF/NETCONF protocol. This YANG module defines NBIs for the IP+optical super controller.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on May 1, 2017.

Copyright Notice

Table of Contents

1. Introduction

1.1. IP+optical solution

IP+optical is a cross-layer collaboration technology for unified management of IP and optical networks. IP+optical adopts the C/S architecture, where the IP network is the client-layer network and the optical network is the server-layer network.

IP+optical use cases include multi-layer topology visualization, automated network deployment, multi-layer automated service deployment, multi-layer protection and restoration, multi-layer optimization, and multi-layer maintenance window.

Based on framework proposed in [ACTN-FWK][I-D.ietf-teas-actn-framework], this draft presents specific information about the IP+optical solution: hierarchical controllers + disabled GMPLS UNIs. This solution does not involve UNI tunnel objects. Therefore, the mapping between IP links and transport services is key point of this solution.

```
                                     +----------+
                                     |IP+Optical|
                                     |   Super  |
                                     |Controller|
                                     +-/------\-+
                                      /        \
                           +---------/--+    +--\--------+
                           |    IP      |    |  Optical  |
                           | Controller |    |Controller |
                           +----+-------+    +---+-------+
                                |                |
                           +----+----------------+-------------+
                          /R    |     R          |R      R    /
                         /         R         R   |           /
             IP         /    R     .    R    .   |  R    R  /
             Layer     +-------------------------+---------+
                             .     .    .    .   |  .    .
                             .     .    .    .   |  .    .
                             .     .    .    .   |  .    .
                             .     .    .    .   |  .    .
                             .     .    .    .   |  .    .
                           +---------------------+-------------+
                          /  O     .  O . O  .   |  O    O    /
                         /         O    .    O  O            /
             Optical    / O    O        O           O    O  /
             Layer     +-----------------------------------+
```

Figure 1: IP+optical solution

In real-world situations, IP+optical super controllers can be separately deployed or combined with other controllers. For example, in IP+optical single-domain scenarios, an IP+optical super controller can be combined with an IP domain controller. In IP multi-domain and optical multi-domain scenarios, you can deploy one separate IP super controller and one separate optical super controller. The two super controllers communicate through RESTConf interfaces and use the IP+VNT algorithm to complete E2E cross-layer path calculation. In such multi-domain scenarios, you can also deploy only one IP+optical super controller and use a unified cross-layer algorithm in the controller to complete E2E cross-layer path calculation.

The system uses loose-coupled dual-controllers to implement cross-layer collaborative path calculation on the virtual network topology (VNT), where the VNT provides the route calculation bridging function. The VNT needs to be defined as a YANG model for configuration management.

This draft provides a YANG model for the RESTCONF/NETCONF protocol. This YANG module defines NBIs for the IP+optical super controller.

1.2. Unified cross-layer algorithm

In this model, inter-layer path computation is performed by a single PCE of a Unified controller that has topology visibility into all layers. Such a PCE is called a multi-layer PCE. In Figure 2, the network is comprised of two layers. NEs H1, H2,H3, and H4 belong to the higher layer, and NEs H2, H3, L1, and L2 belong to the lower layer. The PCE is a multi-layer PCE that has visibility into both layers. It can perform end-to-end path computation across layers (single PCE path computation). For instance, it can compute an optimal path H1-H2-L1-L2-H3-H4. Of course, more complex cooperation may be required if an optimal end-to-end path is desired.

```
                                   Controller
                        +-----------------------------------+
                        |        +--------------+           |
                        |        |Multilayer-PCE|           |
                        |        +--------------+           |
                        | +--------++-----------++--------+ |
                        | |IP      ||Cross-Layer||Optical | |
                        | |Topology||Topology   ||Topology| |
                        | +--------++-----------++--------+ |
                        +----------------+------------------+
                                         |
                                         |
                                         |
                  -----    -----         |        -----    -----
                 | R   |--| R   |........|.......| R   |--| R   |
                 | H1  |  | H2  |               | H3  |  | H4  |
                  -----    -----\               /-----    -----
                                 \-----    -----/
                                 | O   |--| O   |
                                 | L1  |  | L2  |
                                  -----    -----
                      Figure 2: Unified cross-layer algorithm
```

1.3. IP+VNT algorithm

In this model, there is at least one PCE of controller per layer, and each PCE of controller has topology visibility restricted to its own layer. Some providers may want to keep the layer boundaries due to factors such as organizational and/or service management issues. The choice for multiple PCE computation instead of single PCE computation may also be driven by scalability considerations, as in this mode a PCE only needs to maintain topology information for one layer (resulting in a size reduction for the Traffic Engineering Database (TED)). Figure 3 shows multiple PCE inter-layer computation with inter-PCE communication. There is one PCE in each layer. The PCEs from each layer collaborate to compute an end-to-end path across layers. An IP-PCE of IP-domain controller uses IP topology and VNT topology information to perform path calculation at the higher layer. If a VNT link is selected, the IP-domain controller collaborates with the optical-domain controller for path calculation. The optical-PCE of optical-domain controller then uses cross-layer topology and optical topology information to calculate an underlying VNT path.A simple example of cooperation between the PCEs could be as follows:

- o IP controller sends a request to IP-PCE for a path H1-H4 with ip topo and VNT topo.
- o IP-PCE selects VNT link as the entry point and exit point to the lower layer.
- o IP-PCE of IP controller requests a path both ends of VNT link from Optical-PCE of optical controller.
- o Optical-PCE returns H2-L1-L2-H3 to IP-PCE.
- o IP-PCE is now able to compute the full path (H1-H2-L1-L2-H3-H4)

```
          IP Controller                      Optical Controller
    +------------------------+      +------------------------------------+
    |       +-------+        |      |           +------------+           |
    |       |IP-PCE|        +-------+           |Optical-PCE|            |
    |       +-------+        |      |           +------------+           |
    |+---------++-----------+|      | +---------++-----------++---------+ |
    ||IP       ||VNT        ||      | |VNT      ||Cross-Layer||Optical  | |
    ||Topology||Topology    ||      | |Topology||Topology    ||Topology| |
    |+---------++-----------+|      | +---------++-----------++---------+ |
    +-----------+------------+      +-----------+------------------------+
                |                              |
                |                              |
                |                              |
                |                              |
                |                              |
         -----     -----       VNT Link        |  -----    -----
        | R   |-+| R   |.......................|.| R   |--| R   |
        | H1  |  | H2  |                       | | H3  |  | H4  |
         -----     -----\                      | /-----    -----
                         \                      /
                          \                    /
                           \                  /
                            \                /
                             \-----    -----/
                             | O   |--| O   |
                             | L1  |  | L2  |
                              -----    -----
```
Figure 3: IP+VNT algorithm

1.4. VNT Proctect-Group

VNT links support on-demand creation and deletion, and therefore protection can be implemented based on IP links. To implement the protection function, plan and deploy VNT protection groups. IP link switchover can be then implemented if network faults occur or network traffic reaches a threshold.

VNT protection groups support:.

- o Manual and automatic service switchover and switchback
- o 1:1 and N:1 working modes
- o Protection of links with the same source but different sinks, protection of links with different sources but the same sink, and protection of links with both different sources and sinks

   (preamble)

    Please view in a fixed-width font such as Courier.

```
                     VNT W
                   +------------------------------------+
                  /R\---------R           R      R    /
                 /   \               R               /
     IP         /    R VNT P    R    .      R    R  /
     Layer     +------------------------------------+
                     .          .    .      .    .
                     .          .    .      .    .
                     .          .    .      .    .
                     .          .    .      .    .
                     .          .    .      .    .
                   +------------------------------------+
                  /  O        O . O  .      O    O    /
                 /              .    O  O            /
     Optical    / O    O        O           O    O  /
     Layer     +------------------------------------+
                Figure 4: VNT protection groups
```

   (postamble)

2. VNT (IP Link) Model - YANG Tree

(preamble)

```
module: ietf-vnt
   +--rw vnts
      +--rw vnt* [vnt-id]
         +--rw vnt-id                        string
         +--rw vnt-name?                     string
         +--rw src-node-id?                  string
         +--rw src-interface-type?           string
         +--rw src-interface-ip?             inet:ipv4-address
         +--rw src-bind-if-name?             string
         +--rw sink-node-id?                 string
         +--rw sink-interface-type?          string
         +--rw sink-interface-ip?            inet:ipv4-address
         +--rw sink-bind-if-name?            string
         +--rw switch-type?                  uint16
         +--rw vlan-id?                      vlan
         +--rw latency?                      uint32
         +--rw max-reservable-bandwidth?     decimal64
         +--rw bandwidth?                    decimal64
         +--rw te-metric?                    uint32
         +--rw explicit-path-name?           string
         +--rw optical-protection-type?      string
```

(postamble)

3. VNT (IP Link) Model - YANG Code

(preamble)

```
module ietf-vnt {
    namespace "urn:ietf:params:xml:ns:yang:ietf-vnt";
    prefix "vnt";

    import ietf-inet-types {
        prefix inet;
    }

    organization
        "Huawei Technologies";

    contact
        "fupengcheng@huawei.com";

    description
        "The YANG module defines vnt.";

    revision 2016-10-28 {
```

```
            description "Initial revision.";
        }

        /* Typedefs */
        typedef vlan {
          type uint16 {
            range "0..4095";
          }
          description "VLAN ID";
        }

        typedef protection-type {
            type string;
            description
             "ip or optical protection type.";
        }

      /* Grouping */
      grouping vnt-para {
       list vnt {
            key "vnt-id";
            description
            "The list of configured interfaces on the device.";

          leaf vnt-id {
            type string;
            description
              "Id of vnt.";
          }

          leaf vnt-name {
            type string;
            description
              "Name of vnt.";
          }

          leaf src-node-id {
            type string;
            description
              "Id of node.";
          }

          leaf src-interface-type {
            type string;
            description
              "interface type.";
          }
```

```
         leaf src-interface-ip {
           type inet:ipv4-address;
           description
             "Ip of interface.";
         }

         leaf src-bind-if-name {
           type string;
           description
             "source node bind interface name.";
         }

         leaf sink-node-id {
           type string;
           description
             "Id of node.";
         }

         leaf sink-interface-type {
           type string;
           description
             "interface type.";
         }

         leaf sink-interface-ip {
           type inet:ipv4-address;
           description
             "Ip of interface.";
         }

         leaf sink-bind-if-name {
           type string;
           description
             "sink node bind interface name.";
         }

         leaf switch-type {
          type uint16;
          description
            "switch type.";
         }

         leaf vlan-id {
          type vlan;
          description
            "vlan id.";
         }
```

```
         leaf latency {
          type uint32;
          description
            "latency.";
         }

         leaf max-reservable-bandwidth {
          type decimal64 {
           fraction-digits 2;
          }
          description
            "max reservable bandwidth.";
         }

         leaf bandwidth {
          type decimal64 {
           fraction-digits 2;
          }
          description
            "bandwidth.";
         }

         leaf te-metric{
          type uint32;
          description
            "te metric.";
         }

         leaf explicit-path-name {
          type string;
          description
            "explicit path name";
         }

         leaf optical-protection-type {
          type string;
           description
             "optical protection type.";
         }
       }
     }

       /* Main blocks */
       container vnts {
         description
           "Virtual network topology.";

         uses vnt-para;
```

```
    }
  }
```

(postamble)

4. VNT (IP Link) Protection Group Model - YANG Tree

(preamble)

```
      +--rw vnt-protect-groups
         +--rw vnt-protect-group* [group-id]
            +--rw group-id              uint32
            +--rw group-name?           string
            +--rw group-type?           enumeration
            +--rw work-vnt-list
            |  +--rw vnt* [vnt-id]
            |     +--rw vnt-id                        string
            |     +--rw vnt-name?                     string
            |     +--rw src-node-id?                  string
            |     +--rw src-interface-type?           string
            |     +--rw src-interface-ip?             inet:ipv4-address
            |     +--rw src-bind-if-name?             string
            |     +--rw sink-node-id?                 string
            |     +--rw sink-interface-type?          string
            |     +--rw sink-interface-ip?            inet:ipv4-address
            |     +--rw sink-bind-if-name?            string
            |     +--rw switch-type?                  uint16
            |     +--rw vlan-id?                      vlan
            |     +--rw latency?                      uint32
            |     +--rw max-reservable-bandwidth?     decimal64
            |     +--rw bandwidth?                    decimal64
            |     +--rw te-metric?                    uint32
            |     +--rw explicit-path-name?           string
            |     +--rw optical-protection-type?      string
            +--rw protect-vnt-list
            |  +--rw vnt* [vnt-id]
            |     +--rw vnt-id                        string
            |     +--rw vnt-name?                     string
            |     +--rw src-node-id?                  string
            |     +--rw src-interface-type?           string
            |     +--rw src-interface-ip?             inet:ipv4-address
            |     +--rw src-bind-if-name?             string
            |     +--rw sink-node-id?                 string
            |     +--rw sink-interface-type?          string
            |     +--rw sink-interface-ip?            inet:ipv4-address
            |     +--rw sink-bind-if-name?            string
            |     +--rw switch-type?                  uint16
            |     +--rw vlan-id?                      vlan
            |     +--rw latency?                      uint32
            |     +--rw max-reservable-bandwidth?     decimal64
            |     +--rw bandwidth?                    decimal64
            |     +--rw te-metric?                    uint32
            |     +--rw explicit-path-name?           string
            |     +--rw optical-protection-type?      string
            +--rw is-autoaction?        boolean
            +--rw is-return?            boolean
```

(postamble)

5. VNT (IP Link) Protection Group Model - YANG Code

(preamble)

```
module ietf-vnt-protect-group {
    namespace "urn:ietf:params:xml:ns:yang:ietf-vnt-protect-group";
    prefix "vnt-protect-grp";

    import ietf-vnt {
        prefix vnt;
    }

    organization
        "Huawei Technologies";

    contact
        "fupengcheng@huawei.com";

    description
        "The YANG module defines vnt protect group.";

    revision 2016-10-28 {
        description "Initial revision.";
    }

    /* Main blocks */
    container vnt-protect-groups {
      description
        "vnt protect groups.";

      list vnt-protect-group {
        key "group-id";
        description
        "The list of vnt protect groups.";

      leaf group-id {
        type uint32;
        description
          "Id of vnt protect group.";
      }

      leaf group-name {
        type string;
        description
          "Name of vnt protect group.";
      }
```

```
         leaf group-type {
           type enumeration {
               enum "1:1" {
                 description
                   "1:1 type.";
               }
               enum "n:1" {
                 description
                 "n:1 type";
               }
           }
           description
             "type of vnt protect group.";
         }

         container work-vnt-list {
           uses vnt:vnt-para;
           description
             "work vnt list.";
         }

         container protect-vnt-list {
           uses vnt:vnt-para;
           description
             "protect vnt list.";
         }

         leaf is-autoaction {
           type boolean;
           description
             "if it is autoaction.";
         }

         leaf is-return {
           type boolean;
           description
             "if it need return.";
         }
       }
     }
   }
```

(postamble)

6. IANA Considerations

This document makes no request of IANA.

Note to RFC Editor: this section may be removed on publication as an RFC.

7. Security Considerations

8. Acknowledgements

9. Normative References

[I-D.ietf-teas-actn-framework]
Ceccarelli, D. and Y. Lee, "Framework for Abstraction and Control of Traffic Engineered Networks", draft-ietf-teas-actn-framework-01 (work in progress), October 2016.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://www.rfc-editor.org/info/rfc2119>.

Authors' Addresses

Pengcheng FU
Huawei Technologies

Email: fupengcheng@huawei.com

Jianzhong FU
Huawei Technologies

Email: fujianzhong@huawei.com

Aijun.Wang
China Telecom

Email: wangaj@ctbri.com.cn

Ruiquan Jing
China Telecom

Email: jingrq@ctbri.com.cn