

CCAMP Working Group
Internet-Draft
Intended status: Standards Track
Expires: September 12, 2017

X. Zhang
K. Xiang
Huawei Technologies
A. Sharma
R. Rao
Infinera
March 11, 2017

OTN Tunnel YANG Model
draft-sharma-ccamp-otn-tunnel-model-01

Abstract

This document describes the YANG data model for OTN Tunnels.

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 [[RFC2119](#)].

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

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 September 12, 2017.

Copyright Notice

Copyright (c) 2017 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents

carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1.	Introduction	2
2.	Terminology and Notations	2
3.	Model Overview	3
3.1.	Mux Service in Multi-Domain OTN Network	3
3.2.	Bookended and Non-BookEnded OTN Tunnel	4
3.3.	Network and Client side tunnel services	4
3.4.	OTN Tunnel YANG Tree	4
3.5.	OTN Tunnel YANG Code	5
3.6.	Transport Types YANG Code	10
4.	Security Considerations	19
5.	IANA Considerations	19
6.	Acknowledgements	19
7.	Normative References	19
	Authors' Addresses	19

[1.](#) Introduction

OTN transport networks can carry various types of client services. In many cases, the client signal is carried over an OTN tunnel across connected domains in a multi-domain network. These OTN services can either be transported or switched in the OTN network. If an OTN tunnel is switched, then additional parameters need to be provided to create a Mux OTN service.

This document provides YANG model for creating OTN tunnel. The model augments the TE Tunnel model, which is an abstract model to create TE Tunnels.

[2.](#) Terminology and Notations

A simplified graphical representation of the data model is used in this document. The meaning of the symbols in the YANG data tree presented later in this draft is defined in . They are provided below for reference.

- o Brackets "[" and "]" enclose list keys.
- o Abbreviations before data node names: "rw" means configuration (read-write) and "ro" state data (read-only).

- o Symbols after data node names: "?" means an optional node, "!" means a presence container, and "*" denotes a list and leaf-list.
- o Parentheses enclose choice and case nodes, and case nodes are also marked with a colon (":").
- o Ellipsis ("...") stands for contents of subtrees that are not shown.

3. Model Overview

3.1. Mux Service in Multi-Domain OTN Network

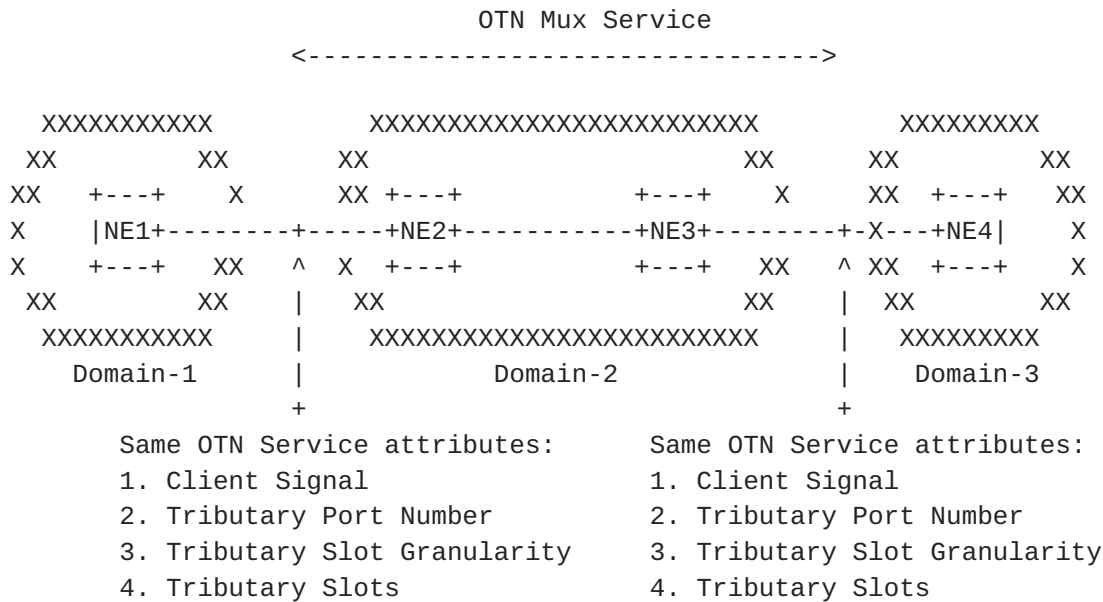


Figure 1: OTN Mux Service in a multi-domain network topology

Figure 1 shows a multi-domain OTN network with three domains. In this example, user wants to setup an end-to-end OTN service that passes through Domain-2. In order to create an OTN mux service in Domain-2, user will need to specify the exact details of the client side LO-ODU on NE2 and NE3, so that these service endpoints can be paired with the LO-ODU endpoints on NE1 and NE4, respectively.

Let's assume that ODU4 is the client side HO-ODU on NE2 and NE3, and the client signal is ODU2. User will need to specify the OTN client signal (ODU2 in this example), the Tributary Port Number (TPN), Tributary Slot Granularities (TSG) and tributary slots to be used.

As shown in the figure above, these service parameters must be the same between NE1 and NE2, and NE3 and NE4.

Once the OTN Mux service is setup in Domain-2, the incoming signal from either NE1 and/or NE4 will be switched inside Domain-2, and delivered to NE at the other end.

3.2. Bookended and Non-BookEnded OTN Tunnel

OTN tunnel model provides support for both bookended and non-bookended OTN tunnels.

For bookended tunnels, the same client signal is present on source and destination endpoints. For example, ODU2e bookended tunnel will have the same ODU2e client signal at both source and destination endpoints.

For non-bookended tunnels, different client signals are present on source and destination endpoints. For example, the client signal can be ODU2e on the source endpoint and the handoff at the destination can be 10GbE-LAN client signal.

3.3. Network and Client side tunnel services

The OTN tunnel model provides support for both network to network and client to client tunnels. For network to network tunnel, network termination points on source and destination node represent source and destination endpoints. For client to client tunnel, client termination points on source and destination node represent source and destination endpoints.

If a client to client tunnel needs to use one or more HO (or server) network to network tunnels, ERO and routing constraints, defined in the base TE model, can be used to route the client tunnel over one or more server tunnels.

3.4. OTN Tunnel YANG Tree


```

module: ietf-otn-tunnel
augment /te:te/te:tunnels/te:tunnel/te:config:
  +--rw payload-treatment?      enumeration
  +--rw src-client-signal?      identityref
  +--rw src-tpn?                uint16
  +--rw src-tsg?                identityref
  +--rw src-tributary-slot-count? uint16
  +--rw src-tributary-slots
  | +--rw values*              uint8
  +--rw dst-client-signal?      identityref
  +--rw dst-tpn?                uint16
  +--rw dst-tsg?                identityref
  +--rw dst-tributary-slot-count? uint16
  +--rw dst-tributary-slots
    +--rw values*              uint8
augment /te:te/te:tunnels/te:tunnel/te:state:
  +--ro payload-treatment?      enumeration
  +--ro src-client-signal?      identityref
  +--ro src-tpn?                uint16
  +--ro src-tsg?                identityref
  +--ro src-tributary-slot-count? uint16
  +--ro src-tributary-slots
  | +--ro values*              uint8
  +--ro dst-client-signal?      identityref
  +--ro dst-tpn?                uint16
  +--ro dst-tsg?                identityref
  +--ro dst-tributary-slot-count? uint16
  +--ro dst-tributary-slots
    +--ro values*              uint8

```

3.5. OTN Tunnel YANG Code

<CODE BEGINS>file "ietf-otn-tunnel@2017-03-11.yang"

```

module ietf-otn-tunnel {
  yang-version 1.1;

  namespace "urn:ietf:params:xml:ns:yang:ietf-otn-tunnel";
  prefix "otn-tunnel";

  import ietf-te { prefix "te"; }
  import ietf-transport-types { prefix "tran-types"; }
  //import yang-ext { prefix ext; revision-date 2013-07-09; }

  organization

```



```
"IETF CCAMP Working Group";
```

```
contact
```

```
"WG Web: <http://tools.ietf.org/wg/ccamp/>
```

```
WG List: <mailto:ccamp@ietf.org>
```

```
Editor: Anurag Sharma  
<mailto:AnSharma@infinera.com>
```

```
Editor: Rajan Rao  
<mailto:rrao@infinera.com>
```

```
Editor: Xian Zhang  
<mailto:zhang.xian@huawei.com>
```

```
Editor: Kun Xiang  
<mailto:xiangkun@huawei.com>";
```

```
description
```

```
"This module defines a model for OTN Tunnel Services.";
```

```
revision "2017-03-11" {  
  description  
    "Revision 0.3";  
  reference "TBD";  
}
```

```
grouping otn-tunnel-endpoint {  
  description "Parameters for OTN tunnel.";
```

```
    leaf payload-treatment {  
      type enumeration {  
        enum switching;  
        enum transport;  
      }  
      default switching;
```

```
      description  
        "Treatment of the incoming payload. Payload can  
        either be switched, or transported as is.";
```

```
    }
```

```
  leaf src-client-signal {  
    type identityref {  
      base tran-types:client-signal;  
    }  
    description
```

```
      "Client signal at the source endpoint of  
      the tunnel.";
```



```
}

leaf src-tpn {
  type uint16 {
    range "0..4095";
  }
  description
    "Tributary Port Number. Applicable in case of mux
    services.";
  reference
    "RFC7139: GMPLS Signaling Extensions for Control of
    Evolving G.709 Optical Transport Networks.";
}

leaf src-tsg {
  type identityref {
    base tran-types:tributary-slot-granularity;
  }
  description
    "Tributary slot granularity. Applicable in case of mux
    services.";
  reference
    "G.709/Y.1331, February 2016: Interfaces for the
    Optical Transport Network (OTN)";
}

leaf src-tributary-slot-count {
  type uint16;
  description
    "Number of tributary slots used at the source.";
}

container src-tributary-slots {
  description
    "A list of tributary slots used by the client
    service. Applicable in case of mux services.";
  leaf-list values {
    type uint8;
    description
      "Tributary tributary slot value.";
    reference
      "G.709/Y.1331, February 2016: Interfaces for the
      Optical Transport Network (OTN)";
  }
}

leaf dst-client-signal {
  type identityref {
```



```
        base tran-types:client-signal;
    }
    description
        "Client signal at the destination endpoint of
        the tunnel.";
}

leaf dst-tpn {
    type uint16 {
        range "0..4095";
    }
    description
        "Tributary Port Number. Applicable in case of mux
        services.";
    reference
        "RFC7139: GMPLS Signaling Extensions for Control of
        Evolving G.709 Optical Transport Networks.";
}

leaf dst-tsg {
    type identityref {
        base tran-types:tributary-slot-granularity;
    }
    description
        "Tributary slot granularity. Applicable in case of mux
        services.";
    reference
        "G.709/Y.1331, February 2016: Interfaces for the
        Optical Transport Network (OTN)";
}

leaf dst-tributary-slot-count {
    type uint16;
    description
        "Number of tributary slots used at the destination.";
}

container dst-tributary-slots {
    description
        "A list of tributary slots used by the client
        service. Applicable in case of mux services.";
    leaf-list values {
        type uint8;
        description
            "Tributary slot value.";
        reference
            "G.709/Y.1331, February 2016: Interfaces for the
            Optical Transport Network (OTN)";
    }
}
```



```
    }
  }
}

/*
Note: Comment has been given to authors of TE Tunnel model to add
tunnel-types to the model in order to identify the technology
type of the service.

grouping otn-service-type {
  description
    "Identifies the OTN Service type.";
  container otn-service {
    presence "Indicates OTN Service.";
    description
      "Its presence identifies the OTN Service type.";
  }
} // otn-service-type

augment "/te:te/te:tunnels/te:tunnel/te:tunnel-types" {
  description
    "Introduce OTN service type for tunnel.";
  ext:augment-identifier otn-service-type-augment;
  uses otn-service-type;
}
*/

/*
Note: Comment has been given to authors of TE Tunnel model to add
list of endpoints under config to support P2MP tunnel.
*/
augment "/te:te/te:tunnels/te:tunnel/te:config" {
  description
    "Augment with additional parameters required for OTN
    service.";
  //ext:augment-identifier otn-tunnel-endpoint-config-augment;
  uses otn-tunnel-endpoint;
}

augment "/te:te/te:tunnels/te:tunnel/te:state" {
  description
    "Augment with additional parameters required for OTN
    service.";
  //ext:augment-identifier otn-tunnel-endpoint-state-augment;
  uses otn-tunnel-endpoint;
}

/*
```


Note: Comment has been given to authors of TE Tunnel model to add tunnel-lifecycle-event to the model. This notification is reported for all lifecycle changes (create, delete, and update) to the tunnel or lsp.

```
augment "/te:tunnel-lifecycle-event" {
  description
    "OTN service event";
  uses otn-service-type;
  uses otn-tunnel-params;

  list endpoint {
    key
      "endpoint-address tp-id";
    description
      "List of Tunnel Endpoints.";
    uses te:tunnel-endpoint;
    uses otn-tunnel-params;
  }
}
*/
}
```

<CODE ENDS>

3.6. Transport Types YANG Code

<CODE BEGINS> file "ietf-transport-types@2016-10-25.yang"

```
module ietf-transport-types {
  namespace "urn:ietf:params:xml:ns:yang:ietf-transport-types";
  prefix "tran-types";

  organization
    "IETF CCAMP Working Group";
  contact
    "WG Web: <http://tools.ietf.org/wg/ccamp/>
    WG List: <mailto:ccamp@ietf.org>

    Editor: Anurag Sharma
           <mailto:AnSharma@infinera.com>

    Editor: Rajan Rao
           <mailto:rrao@infinera.com>

    Editor: Xian Zhang
```



```
        <mailto:zhang.xian@huawei.com>";

description
    "This module defines transport types.";

revision "2016-10-25" {
    description
        "Revision 0.2";
    reference "TBD";
}

identity tributary-slot-granularity {
    description
        "Tributary slot granularity.";
    reference
        "G.709/Y.1331, February 2016: Interfaces for the
        Optical Transport Network (OTN)";
}

identity tsg-1.25G {
    base tributary-slot-granularity;
    description
        "1.25G tributary slot granularity.";
}

identity tsg-2.5G {
    base tributary-slot-granularity;
    description
        "2.5G tributary slot granularity.";
}

identity tributary-protocol-type {
    description
        "Base identity for protocol framing used by
        tributary signals.";
}

identity prot-OTU1 {
    base tributary-protocol-type;
    description
        "OTU1 protocol (2.66G)";
}

/*
identity prot-OTU1e {
    base tributary-protocol-type;
    description
        "OTU1e type (11.04G)";
```



```
}

identity prot-OTU1f {
    base tributary-protocol-type;
    description
        "OTU1f type (11.27G)";
}
*/

identity prot-OTU2 {
    base tributary-protocol-type;
    description
        "OTU2 type (10.70G)";
}

identity prot-OTU2e {
    base tributary-protocol-type;
    description
        "OTU2e type (11.09G)";
}

/*

identity prot-OTU2f {
    base tributary-protocol-type;
    description
        "OTU2f type (11.31G)";
}
*/

identity prot-OTU3 {
    base tributary-protocol-type;
    description
        "OTU3 type (43.01G)";
}

/*

identity prot-OTU3e1 {
    base tributary-protocol-type;
    description
        "OTU3e1 type (44.57G)";
}

identity prot-OTU3e2 {
    base tributary-protocol-type;
    description
        "OTU3e2 type (44.58G)";
}
*/
```



```
identity prot-OTU4 {
  base tributary-protocol-type;
  description
    "OTU4 type (111.80G)";
}

identity prot-OTUCn {
  base tributary-protocol-type;
  description
    "OTUCn type (beyond 100G)";
}

identity prot-ODU0 {
  base tributary-protocol-type;
  description
    "ODU0 protocol (1.24G).";
}

identity prot-ODU1 {
  base tributary-protocol-type;
  description
    "ODU1 protocol (2.49G).";
}

/*
identity prot-ODU1e {
  base tributary-protocol-type;
  description
    "ODU1e protocol (10.35G).";
}

identity prot-ODU1f {
  base tributary-protocol-type;
  description
    "ODU1f protocol (10.56G).";
}
*/

identity prot-ODU2 {
  base tributary-protocol-type;
  description
    "ODU2 protocol (10.03G).";
}

identity prot-ODU2e {
  base tributary-protocol-type;
  description
    "ODU2e protocol (10.39G).";
```



```
}

/*
identity prot-ODU2f {
    base tributary-protocol-type;
    description
        "ODU2f protocol (10.60G).";
}
*/

identity prot-ODU3 {
    base tributary-protocol-type;
    description
        "ODU3 protocol (40.31G).";
}

/*
identity prot-ODU3e1 {
    base tributary-protocol-type;
    description
        "ODU3e1 protocol (41.77G).";
}

identity prot-ODU3e2 {
    base tributary-protocol-type;
    description
        "ODU3e2 protocol (41.78G).";
}
*/

identity prot-ODU4 {
    base tributary-protocol-type;
    description
        "ODU4 protocol (104.79G).";
}

identity prot-ODUFlex-cbr {
    base tributary-protocol-type;
    description
        "ODU Flex CBR protocol for transporting constant bit
        rate signal.";
}

identity prot-ODUFlex-gfp {
    base tributary-protocol-type;
    description
        "ODU Flex GFP protocol for transporting stream of packets
        using Generic Framing Procedure.";
```



```
}

identity prot-ODUCn {
  base tributary-protocol-type;
  description
    "ODUCn protocol (beyond 100G).";
}

identity prot-1GbE {
  base tributary-protocol-type;
  description
    "1G Ethernet protocol";
}

identity prot-10GbE-LAN {
  base tributary-protocol-type;
  description
    "10G Ethernet LAN protocol";
}

identity prot-40GbE {
  base tributary-protocol-type;
  description
    "40G Ethernet protocol";
}

identity prot-100GbE {
  base tributary-protocol-type;
  description
    "100G Ethernet protocol";
}

identity client-signal {
  description
    "Base identity from which specific client signals for the
    tunnel are derived.";
}

identity client-signal-1GbE {
  base client-signal;
  description
    "Client signal type of 1GbE";
}

identity client-signal-10GbE-LAN {
  base client-signal;
  description
    "Client signal type of 10GbE LAN";
}
```



```
    }

    identity client-signal-10GbE-WAN {
        base client-signal;
        description
            "Client signal type of 10GbE WAN";
    }

    identity client-signal-40GbE {
        base client-signal;
        description
            "Client signal type of 40GbE";
    }

    identity client-signal-100GbE {
        base client-signal;
        description
            "Client signal type of 100GbE";
    }

    identity client-signal-OC3_STM1 {
        base client-signal;
        description
            "Client signal type of OC3 & STM1";
    }

    identity client-signal-OC12_STM4 {
        base client-signal;
        description
            "Client signal type of OC12 & STM4";
    }

    identity client-signal-OC48_STM16 {
        base client-signal;
        description
            "Client signal type of OC48 & STM16";
    }

    identity client-signal-OC192_STM64 {
        base client-signal;
        description
            "Client signal type of OC192 & STM64";
    }

    identity client-signal-OC768_STM256 {
        base client-signal;
        description
            "Client signal type of OC768 & STM256";
```



```
    }

    identity client-signal-ODU0 {
        base client-signal;
        description
            "Client signal type of ODU0 (1.24G)";
    }

    identity client-signal-ODU1 {
        base client-signal;
        description
            "ODU1 protocol (2.49G)";
    }

    identity client-signal-ODU2 {
        base client-signal;
        description
            "Client signal type of ODU2 (10.03G)";
    }

    identity client-signal-ODU2e {
        base client-signal;
        description
            "Client signal type of ODU2e (10.39G)";
    }

    identity client-signal-ODU3 {
        base client-signal;
        description
            "Client signal type of ODU3 (40.31G)";
    }

    /*
    identity client-signal-ODU3e2 {
        base client-signal;
        description
            "Client signal type of ODU3e2 (41.78G)";
    }
    */

    identity client-signal-ODU4 {
        base client-signal;
        description
            "Client signal type of ODU4 (104.79G)";
    }

    identity client-signal-ODUFlex-cbr {
        base client-signal;
```



```
        description
            "Client signal type of ODU Flex CBR";
    }

    identity client-signal-ODUFlex-gfp {
        base client-signal;
        description
            "Client signal type of ODU Flex GFP";
    }

    identity client-signal-ODUCn {
        base client-signal;
        description
            "Client signal type of ODUCn (beyond 100G).";
    }

    identity client-signal-FC400 {
        base client-signal;
        description
            "Client signal type of Fibre Channel FC400.";
    }

    identity client-signal-FC800 {
        base client-signal;
        description
            "Client signal type of Fibre Channel FC800.";
    }

    identity client-signal-FICON-4G {
        base client-signal;
        description
            "Client signal type of Fibre Connection 4G.";
    }

    identity client-signal-FICON-8G {
        base client-signal;
        description
            "Client signal type of Fibre Connection 8G.";
    }
}
<CODE ENDS>
```


4. Security Considerations

TBD.

5. IANA Considerations

TBD.

6. Acknowledgements

TBD.

7. Normative References

- [G.709] "Interfaces for the Optical Transport Network(OTN)", G.709/Y.1331 Recommendation , June 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>>.
- [RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", [RFC 6020](#), DOI 10.17487/RFC6020, October 2010, <<http://www.rfc-editor.org/info/rfc6020>>.
- [RFC7139] Zhang, F., Ed., Zhang, G., Belotti, S., Ceccarelli, D., and K. Pithewan, "GMPLS Signaling Extensions for Control of Evolving G.709 Optical Transport Networks", [RFC 7139](#), DOI 10.17487/RFC7139, March 2014, <<http://www.rfc-editor.org/info/rfc7139>>.

Authors' Addresses

Xian Zhang
Huawei Technologies
F3-5-B R&D Center, Huawei Industrial Base, Bantian, Longgang District
Shenzhen, Guangdong 518129
P.R.China

Email: zhang.xian@huawei.com

Kun Xiang
Huawei Technologies
F3 R&D Center, Huawei Industrial Base, Bantian, Longgang District
Shenzhen, Guangdong 518129
P.R.China

Email: xiangkun@huawei.com

Anurag Sharma
Infinera

Email: ansharma@infinera.com

Rajan Rao
Infinera
169 Java Drive
Sunnyvale, CA 94089
USA

Email: rrao@infinera.com

