

Network Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: May 4, 2020

J. Haas  
Juniper Networks, Inc.  
A. Fu  
Bloomberg L.P.  
November 1, 2019

BFD Encapsulated in Large Packets  
draft-ietf-bfd-large-packets-02

Abstract

The Bidirectional Forwarding Detection (BFD) protocol is commonly used to verify connectivity between two systems. BFD packets are typically very small. It is desirable in some circumstances to know that not only is the path between two systems reachable, but also that it is capable of carrying a payload of a particular size. This document discusses thoughts on how to implement such a mechanism using BFD in Asynchronous mode.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" are to be interpreted as described in [RFC2119] only when they appear in all upper case. They may also appear in lower or mixed case as English words, without normative meaning.

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 May 4, 2020.

## Copyright Notice

Copyright (c) 2019 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. BFD Encapsulated in Large Packets . . . . .	3
3. Implementation and Deployment Considerations . . . . .	3
3.1. Implementations that do not support Large BFD Packets . .	3
3.2. Selecting MTU size to be detected . . . . .	4
3.3. Detecting MTU mismatches . . . . .	4
3.4. Equal Cost Multiple Paths (ECMP) or other Load Balancing Considerations . . . . .	5
3.5. S-BFD . . . . .	5
4. Security Considerations . . . . .	5
5. IANA Considerations . . . . .	6
6. Acknowledgments . . . . .	6
7. References . . . . .	6
7.1. Normative References . . . . .	6
7.2. Informative References . . . . .	7
Appendix A. Related Features . . . . .	7
Authors' Addresses . . . . .	7

## 1. Introduction

The Bidirectional Forwarding Detection (BFD) [RFC5880] protocol is commonly used to verify connectivity between two systems. However, some applications may require that the Path MTU [RFC1191] between those two systems meets a certain minimum criteria. When the Path MTU decreases below the minimum threshold, those applications may wish to consider the path unusable.

BFD may be encapsulated in a number of transport protocols. An example of this is single-hop BFD [RFC5881]. In that case, the link MTU configuration is typically enough to guarantee communication between the two systems for that size MTU. BFD Echo mode

(Section 6.4 of [RFC5880]) is sufficient to permit verification of the Path MTU of such directly connected systems. Previous proposals ([I-D.haas-xiao-bfd-echo-path-mtu]) have been made for testing Path MTU for such directly connected systems. However, in the case of multi-hop BFD [RFC5883], this guarantee does not hold.

The encapsulation of BFD in multi-hop sessions is a simple UDP packet. The BFD elements of procedure (Section 6.8.6 of [RFC5880]) covers validating the BFD payload. However, the specification is silent on the length of the encapsulation that is carrying the BFD PDU. While it is most common that the transport protocol payload (i.e. UDP) length is the exact size of the BFD PDU, this is not required by the elements of procedure. This leads to the possibility that the transport protocol length may be larger than the contained BFD PDU.

## 2. BFD Encapsulated in Large Packets

Support for BFD between two systems is typically configured, even if the actual session may be dynamically created by a client protocol. A new BFD variable is defined in this document:

bfd.PaddedPduSize

The BFD transport protocol payload size is increased to this value. The contents of this additional payload MUST be zero. The minimum size of this variable MUST NOT be smaller than permitted by the element of BFD procedure; 24 or 26 - see Section 6.8.6 of [RFC5880].

The Don't Fragment bit (Section 2.3 of [RFC0791]) of the IP payload, when using IPv4 encapsulation, MUST be set.

## 3. Implementation and Deployment Considerations

### 3.1. Implementations that do not support Large BFD Packets

While this document proposes no change to the BFD protocol, implementations may not permit arbitrarily padded transport PDUs to carry BFD packets. While Section 6 of [RFC5880] warns against excessive pedantry, implementations may not work with this mechanism without additional support.

[RFC5880], section 6.8.6, discusses the procedures for receiving BFD Control packets. When an implementation is incapable of processing Large BFD Packets, it could manifest in one of two possible ways:

- o A receiving BFD implementation is incapable of accepting Large BFD Packets. This is identical to the packet being discarded.

- o A receiving BFD implementation is capable of accepting Large BFD Packets, but the Control packet is improperly rejected during validation procedures. This is identical to the packet being discarded.

In each of these cases, the BFD state machine would behave as if it were not receiving Control packets and the implementation would follow normal BFD procedures with regards to not having received Control packets.

### 3.2. Selecting MTU size to be detected

Since the consideration is path MTU, BFD sessions using this feature only need to use a `bfd.PaddedPduSize` appropriate to exercise the path MTU for the desired application. This may be significantly smaller than the system's link MTU; e.g. desired path MTU is 1500 bytes while the interface MTU that BFD with large packets is running on is 9000 bytes.

In the case multiple BFD clients desire to test the same BFD endpoints using different `bfd.PaddedPduSize` parameters, implementations should select the largest `bfd.PaddedPduSize` parameter from the configured sessions. This is similar to how implementations of BFD select the most aggressive timing parameters for multiple sessions to the same endpoint.

### 3.3. Detecting MTU mismatches

The accepted MTU for an interface is impacted by packet encapsulation considerations at a given layer; e.g. layer 2, layer 3, tunnel, etc. A common misconfiguration of interface parameters is inconsistent MTU. In the presence of inconsistent MTU, it is possible for applications to have unidirectional connectivity.

When it is necessary for an application using BFD with Large Packets to test the bi-directional Path MTU, it is necessary to configure the `bfd.PaddedPduSize` parameter on both sides of an interface. E.g., if the desire is to verify a 1500 byte MTU in both directions on an Ethernet or point to point link, each side of the BFD session must have `bfd.PaddedPduSize` set to 1500. In the absence of such consistent configuration, BFD with Large Packets may correctly determine unidirectional connectivity at the tested MTU, but bi-directional MTU may not be properly validated.

It should be noted that some interfaces may intentionally have different MTUs. Setting the `bfd.PaddedPduSize` appropriately for each side of the interface supports such scenarios.

### 3.4. Equal Cost Multiple Paths (ECMP) or other Load Balancing Considerations

Various mechanisms are utilized to increase throughput between two endpoints at various network layers. Such features include Link Aggregate Groups (LAGs) or ECMP forwarding. Such mechanisms balance traffic across multiple physical links while hiding the details of that balancing from the higher networking layers. The details of that balancing are highly implementation specific.

In the presence of such load balancing mechanisms, it is possible to have member links that are not properly forwarding traffic. In such circumstances, this will result in dropped traffic when traffic is chosen to be load balanced across those member links.

Such load balancing mechanisms may not permit all link members to be properly tested by BFD. This is because the BFD Control packets may be forwarded only along links that are up. BFD on LAG, [RFC7130], was developed to help cover one such scenario. However, for testing forwarding over multiple hops, there is no such specified general purpose BFD mechanism for exercising all links in an ECMP. This may result in a BFD session being in the Up state while some traffic may be dropped or otherwise negatively impacted along some component links.

Some BFD implementations utilize their internal understanding of the component links and their resultant forwarding to exercise BFD in such a way to better test the ECMP members and to tie the BFD session state to the health of that ECMP. Due to the implementation specific load balancing, it is not possible to standardize such additional mechanisms for BFD.

Mis-configuration of some member MTUs may lead to Load Balancing that may have an inconsistent Path MTU depending on how the traffic is balanced. While the intent of BFD with Large Packets is to verify path MTU, it is subject to the same considerations above.

### 3.5. S-BFD

This mechanism also can be applied to other forms of BFD, including S-BFD [RFC7880].

## 4. Security Considerations

This document does not change the underlying security considerations of the BFD protocol or its encapsulations.

## 5. IANA Considerations

This document introduces no additional considerations to IANA.

## 6. Acknowledgments

The authors would like to thank Les Ginsberg, Mahesh Jethandani, Robert Raszuk, and Ketan Talaulikar, for their valuable feedback on this proposal.

## 7. References

### 7.1. Normative References

- [RFC0791] Postel, J., "Internet Protocol", STD 5, RFC 791, DOI 10.17487/RFC0791, September 1981, <<https://www.rfc-editor.org/info/rfc791>>.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC5880] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD)", RFC 5880, DOI 10.17487/RFC5880, June 2010, <<https://www.rfc-editor.org/info/rfc5880>>.
- [RFC5881] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD) for IPv4 and IPv6 (Single Hop)", RFC 5881, DOI 10.17487/RFC5881, June 2010, <<https://www.rfc-editor.org/info/rfc5881>>.
- [RFC5883] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD) for Multihop Paths", RFC 5883, DOI 10.17487/RFC5883, June 2010, <<https://www.rfc-editor.org/info/rfc5883>>.
- [RFC7130] Bhatia, M., Ed., Chen, M., Ed., Boutros, S., Ed., Binderberger, M., Ed., and J. Haas, Ed., "Bidirectional Forwarding Detection (BFD) on Link Aggregation Group (LAG) Interfaces", RFC 7130, DOI 10.17487/RFC7130, February 2014, <<https://www.rfc-editor.org/info/rfc7130>>.
- [RFC7880] Pignataro, C., Ward, D., Akiya, N., Bhatia, M., and S. Pallagatti, "Seamless Bidirectional Forwarding Detection (S-BFD)", RFC 7880, DOI 10.17487/RFC7880, July 2016, <<https://www.rfc-editor.org/info/rfc7880>>.

## 7.2. Informative References

- [I-D.haas-xiao-bfd-echo-path-mtu]  
Haas, J. and M. Xiao, "Application of the BFD Echo function for Path MTU Verification or Detection", draft-haas-xiao-bfd-echo-path-mtu-01 (work in progress), July 2011.
- [RFC1191] Mogul, J. and S. Deering, "Path MTU discovery", RFC 1191, DOI 10.17487/RFC1191, November 1990, <<https://www.rfc-editor.org/info/rfc1191>>.
- [RFC3719] Parker, J., Ed., "Recommendations for Interoperable Networks using Intermediate System to Intermediate System (IS-IS)", RFC 3719, DOI 10.17487/RFC3719, February 2004, <<https://www.rfc-editor.org/info/rfc3719>>.

## Appendix A. Related Features

IS-IS [RFC3719] supports a Padding feature for its hellos. This provides the ability to detect inconsistent link MTUs.

## Authors' Addresses

Jeffrey Haas  
Juniper Networks, Inc.  
1133 Innovation Way  
Sunnyvale, CA 94089  
US

Email: [jhaas@juniper.net](mailto:jhaas@juniper.net)

Albert Fu  
Bloomberg L.P.

Email: [afu14@bloomberg.net](mailto:afu14@bloomberg.net)

Network Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: 6 June 2022

E. Chen  
Palo Alto Networks  
N. Shen  
Zededa  
R. Raszuk  
NTT Network Innovations  
R. Rahman  
3 December 2021

Unsolicited BFD for Sessionless Applications  
draft-ietf-bfd-unsolicited-09

Abstract

For operational simplification of "sessionless" applications using BFD, in this document we present procedures for "unsolicited BFD" that allow a BFD session to be initiated by only one side, and be established without explicit per-session configuration or registration by the other side (subject to certain per-interface or per-router policies).

We also introduce a new YANG module to configure and manage "unsolicited BFD". The YANG module in this document conforms to the Network Management Datastore Architecture (NMDA) [RFC8342].

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

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 <https://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 6 June 2022.

#### Copyright Notice

Copyright (c) 2021 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 (<https://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 Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

#### Table of Contents

1. Introduction . . . . .	2
2. Procedures for Unsolicited BFD . . . . .	3
3. State Variables . . . . .	4
4. YANG Data Model . . . . .	5
4.1. Unsolicited BFD Hierarchy . . . . .	5
4.2. Unsolicited BFD Module . . . . .	6
5. IANA Considerations . . . . .	10
6. Acknowledgments . . . . .	10
7. Security Considerations . . . . .	10
7.1. BFD Protocol Security Considerations . . . . .	11
7.2. YANG Module Security Considerations . . . . .	11
8. References . . . . .	12
8.1. Normative References . . . . .	12
8.2. Informative References . . . . .	14
Authors' Addresses . . . . .	14

#### 1. Introduction

The current implementation and deployment practice for BFD ([RFC5880] and [RFC5881]) usually requires BFD sessions be explicitly configured or registered on both sides. This requirement is not an issue when an application like BGP [RFC4271] has the concept of a "session" that involves both sides for its establishment. However, this requirement can be operationally challenging when the prerequisite "session" does not naturally exist between two endpoints in an application. Simultaneous configuration and coordination may be required on both sides for BFD to take effect. For example:

- \* When BFD is used to keep track of the "liveness" of the nexthop of static routes. Although only one side may need the BFD functionality, currently both sides need to be involved in specific configuration and coordination and in some cases static routes are created unnecessarily just for BFD.
- \* When BFD is used to keep track of the "liveness" of the third-party nexthop of BGP routes received from the Route Server [RFC7947] at an Internet Exchange Point (IXP). As the third-party nexthop is different from the peering address of the Route Server, for BFD to work, currently two routers peering with the Route Server need to have routes and nexthops from each other (although indirectly via the Router Server), and the nexthop of each router must be present at the same time. These issues are also discussed in [I-D.ietf-idr-rs-bfd].

Clearly it is beneficial and desirable to reduce or eliminate unnecessary configurations and coordination in these "sessionless" applications using BFD.

In this document we present procedures for "unsolicited BFD" that allow a BFD session to be initiated by only one side, and be established without explicit per-session configuration or registration by the other side (subject to certain per-interface or per-router policies).

With "unsolicited BFD" there is potential risk for excessive resource usage by BFD from "unexpected" remote systems. To mitigate such risks, several mechanisms are recommended in the Security Considerations section.

Compared to the "Seamless BFD" [RFC7880], this proposal involves only minor procedural enhancements to the widely deployed BFD itself. Thus we believe that this proposal is inherently simpler in the protocol itself and deployment. As an example, it does not require the exchange of BFD discriminators over an out-of-band channel before the BFD session bring-up.

When BGP Add-Path [RFC7911] is deployed at an IXP using the Route Server, multiple BGP paths (when exist) can be made available to the clients of the Router Server as described in [RFC7947]. The "unsolicited BFD" can be used in BGP route selection by these clients to eliminate paths with "inaccessible nexthops".

## 2. Procedures for Unsolicited BFD

With "unsolicited BFD", one side takes the "Active role" and the other side takes only the "Passive role" as described in [RFC5880].

On the passive side, the "unsolicited BFD" SHOULD be explicitly configured on an interface or globally (apply to all interfaces). The BFD parameters can be either per-interface or per-router based. It MAY also choose to use the parameters that the active side uses in its BFD Control packets. The "My Discriminator", however, MUST be chosen to allow multiple unsolicited BFD sessions.

The active side starts sending the BFD Control packets as specified in [RFC5880]. The passive side does not send BFD Control packets.

When the passive side receives a BFD Control packet from the active side with 0 as "Your Discriminator" and does not find an existing BFD session, the passive side MAY create a matching BFD session toward the active side, if permitted by local configuration.

It would then start sending the BFD Control packets and perform necessary procedure for bringing up, maintaining and tearing down the BFD session. If the BFD session fails to get established within certain specified time, or if an established BFD session goes down, the passive side would stop sending BFD Control packets and MAY delete the BFD session created until the BFD Control packets is initiated by the active side again.

When an Unsolicited BFD session goes down, an implementation MAY retain the session state for a period of time, which may be configurable. Retaining this state can be useful for operational purposes.

The "Passive role" may change to the "Active role" when a local client registers for the same BFD session, and from the "Active role" to the "Passive role" when there is no longer any locally registered client for the BFD session.

### 3. State Variables

This document defines a new state variable called Unsolicited Role.

bfd.UnsolicitedRole

The operational mode of BFD interface when configured for unsolicited behaviour. Options can be either PASSIVE, ACTIVE or NULL (NULL - not initialized) for unsolicited BFD sessions. Default (not configured for unsolicited behaviour) MUST be set to NULL if present on the interface.

#### 4. YANG Data Model

This section extends the YANG data model for BFD [RFC9127] to cover unsolicited BFD. We import [RFC8349] since the "bfd" container in [RFC9127] is under "control-plane-protocol".

##### 4.1. Unsolicited BFD Hierarchy

Configuration for unsolicited BFD parameters for IP single-hop sessions can be done at 2 levels:

- \* Globally, i.e. for all interfaces. This requires support for the "unsolicited-params-global" feature.
- \* For specific interfaces. This requires support for the "unsolicited-params-per-interface" feature.

For operational data, a new "unsolicited" container has been added for BFD IP single-hop sessions.

The tree diagram below uses the graphical representation of data models, as defined in [RFC8340].

```

module: ietf-bfd-unsolicited

augment /rt:routing/rt:control-plane-protocols
  /rt:control-plane-protocol/bfd:bfd/bfd-ip-sh:ip-sh:
  +--rw unsolicited {bfd-unsol:unsolicited-params-global}?
    +--rw enabled? boolean
    +--rw local-multiplier? multiplier
    +--rw (interval-config-type)?
      +--:(tx-rx-intervals)
        | +--rw desired-min-tx-interval? uint32
        | +--rw required-min-rx-interval? uint32
      +--:(single-interval) {single-minimum-interval}?
        +--rw min-interval? uint32
augment /rt:routing/rt:control-plane-protocols
  /rt:control-plane-protocol/bfd:bfd/bfd-ip-sh:ip-sh
  /bfd-ip-sh:interfaces:
  +--rw unsolicited {bfd-unsol:unsolicited-params-per-interface}?
    +--rw enabled? boolean
    +--rw local-multiplier? multiplier
    +--rw (interval-config-type)?
      +--:(tx-rx-intervals)
        | +--rw desired-min-tx-interval? uint32
        | +--rw required-min-rx-interval? uint32
      +--:(single-interval) {single-minimum-interval}?
        +--rw min-interval? uint32
augment /rt:routing/rt:control-plane-protocols
  /rt:control-plane-protocol/bfd:bfd/bfd-ip-sh:ip-sh
  /bfd-ip-sh:sessions/bfd-ip-sh:session:
  +--ro unsolicited
    +--ro role? bfd-unsol:unsolicited-role

```

#### 4.2. Unsolicited BFD Module

```

<CODE BEGINS> file "ietf-bfd-unsolicited@2021-11-23.yang"
module ietf-bfd-unsolicited {

  yang-version 1.1;

  namespace "urn:ietf:params:xml:ns:yang:ietf-bfd-unsolicited";

  prefix "bfd-unsol";

  // RFC Ed.: replace occurrences of YYYY with actual RFC numbers
  // and remove this note

  import ietf-bfd-types {

```

```
    prefix "bfd-types";
    reference
      "RFC 9127: YANG Data Model for Bidirectional Forwarding Detection
      (BFD)";
  }

  import ietf-bfd {
    prefix "bfd";
    reference
      "RFC 9127: YANG Data Model for Bidirectional Forwarding Detection
      (BFD)";
  }

  import ietf-bfd-ip-sh {
    prefix "bfd-ip-sh";
    reference
      "RFC 9127: YANG Data Model for Bidirectional Forwarding Detection
      (BFD)";
  }

  import ietf-routing {
    prefix "rt";
    reference
      "RFC 8349: A YANG Data Model for Routing Management
      (NMDA version)";
  }

  organization "IETF BFD Working Group";

  contact
    "WG Web: <https://datatracker.ietf.org/wg/bfd/>
    WG List: <rtg-bfd@ietf.org>

    Editors: Enke Chen (enchen@paloaltonetworks.com),
            Naiming Shen (naiming@zededa.com),
            Robert Raszuk (robert@raszuk.net),
            Reshad Rahman (reshad@yahoo.com);

  description
    "This module contains the YANG definition for BFD unsolicited
    as per RFC YYYY.

    Copyright (c) 2021 IETF Trust and the persons
    identified as authors of the code. All rights reserved.

    Redistribution and use in source and binary forms, with or
    without modification, is permitted pursuant to, and subject
    to the license terms contained in, the Simplified BSD License
```

set forth in Section 4.c of the IETF Trust's Legal Provisions  
Relating to IETF Documents  
(<http://trustee.ietf.org/license-info>).

This version of this YANG module is part of RFC YYYY; see  
the RFC itself for full legal notices.";

```
reference "RFC YYYY";

revision 2021-11-23 {
  description
    "Initial revision.";
  reference
    "RFC YYYY: Unsolicited BFD for Sessionless Applications.";
}

/*
 * Feature definitions
 */
feature unsolicited-params-global {
  description
    "This feature indicates that the server supports global
    parameters for unsolicited sessions.";
  reference
    "RFC YYYY: Unsolicited BFD for Sessionless Applications.";
}

feature unsolicited-params-per-interface {
  description
    "This feature indicates that the server supports per-interface
    parameters for unsolicited sessions.";
  reference
    "RFC YYYY: Unsolicited BFD for Sessionless Applications.";
}

/*
 * Type Definitions
 */
typedef unsolicited-role {
  type enumeration {
    enum unsolicited-active {
      description "Active role";
    }
    enum unsolicited-passive {
      description "Passive role";
    }
  }
  description "Unsolicited role";
}
```

```

    }

    /*
    * Augments
    */
    augment "/rt:routing/rt:control-plane-protocols/"
        + "rt:control-plane-protocol/bfd:bfd/bfd-ip-sh:ip-sh" {
        if-feature bfd-unsol:unsolicited-params-global;
        description
            "Augmentation for BFD unsolicited parameters";
        container unsolicited {
            description
                "BFD unsolicited top level container";
            leaf enabled {
                type boolean;
                default false;
                description
                    "BFD unsolicited enabled globally for IP single-hop.";
            }
            uses bfd-types:base-cfg-parms;
        }
    }

    augment "/rt:routing/rt:control-plane-protocols/"
        + "rt:control-plane-protocol/bfd:bfd/bfd-ip-sh:ip-sh/"
        + "bfd-ip-sh:interfaces" {
        if-feature bfd-unsol:unsolicited-params-per-interface;
        description
            "Augmentation for BFD unsolicited on IP single-hop interface";
        container unsolicited {
            description
                "BFD IP single-hop interface unsolicited top level
                container";
            leaf enabled {
                type boolean;
                default false;
                description
                    "BFD unsolicited enabled on this interface.";
            }
            uses bfd-types:base-cfg-parms;
        }
    }

    augment "/rt:routing/rt:control-plane-protocols/"
        + "rt:control-plane-protocol/bfd:bfd/bfd-ip-sh:ip-sh/"
        + "bfd-ip-sh:sessions/bfd-ip-sh:session" {
        description
            "Augmentation for BFD unsolicited on IP single-hop session";
    }

```



```
    container unsolicited {  
        config false;  
        description  
            "BFD IP single-hop session unsolicited top level container";  
        leaf role {  
            type bfd-unsol:unsolicited-role;  
            description "Role.";  
        }  
    }  
}  
}  
<CODE ENDS>
```

## 5. IANA Considerations

This document registers the following namespace URI in the "IETF XML Registry" [RFC3688]:

URI: urn:ietf:params:xml:ns:yang:ietf-bfd-unsolicited

Registrant Contact: The IESG.

XML: N/A; the requested URI is an XML namespace.

This document registers the following YANG module in the "YANG Module Names" registry [RFC6020]:

Name: ietf-bfd-unsolicited

Namespace: urn:ietf:params:xml:ns:yang:ietf-bfd-unsolicited

Prefix: bfd-unsol

Reference: RFC YYYY

## 6. Acknowledgments

Authors would like to thank Acee Lindem, Greg Mirsky, Jeffrey Haas, Raj Chetan and Tom Petch for their review and valuable input.

## 7. Security Considerations

### 7.1. BFD Protocol Security Considerations

The same security considerations and protection measures as those described in [RFC5880] and [RFC5881] normatively apply to this document. With "unsolicited BFD" there is potential risk for excessive resource usage by BFD from "unexpected" remote systems. To mitigate such risks, the following measures are mandatory:

- \* Limit the feature to specific interfaces, and to a single-hop BFD with "TTL=255" [RFC5082]. For numbered interfaces, the source address of an incoming BFD packet should belong to the subnet of the interface on which the BFD packet is received. For unnumbered interfaces the above check should be aligned with routing protocol addresses running on such pair of interfaces.
- \* Apply "policy" to allow BFD packets only from certain subnets or hosts.
- \* Deploy the feature only in certain "trustworthy" environment, e.g., at an IXP, or between a provider and its customers.
- \* Adjust BFD parameters as needed for the particular deployment and scale.
- \* Use BFD authentication.

### 7.2. YANG Module Security Considerations

The YANG module specified in this document defines a schema for data that is designed to be accessed via network management protocols such as NETCONF [RFC6241] or RESTCONF [RFC8040]. The lowest NETCONF layer is the secure transport layer, and the mandatory-to-implement secure transport is Secure Shell (SSH) [RFC6242]. The lowest RESTCONF layer is HTTPS, and the mandatory-to-implement secure transport is TLS [RFC8446].

The NETCONF access control model [RFC8341] provides the means to restrict access for particular NETCONF or RESTCONF users to a preconfigured subset of all available NETCONF or RESTCONF protocol operations and content.

There are a number of data nodes defined in this YANG module that are writable/creatable/deletable (i.e., config true, which is the default). These data nodes may be considered sensitive or vulnerable in some network environments. Write operations (e.g., edit-config) to these data nodes without proper protection can have a negative effect on network operations. These are the subtrees and data nodes and their sensitivity/vulnerability:

```
/routing/control-plane-protocols/control-plane-protocol/bfd/ip-sh  
/unsolicited:
```

- \* data node "enabled" enables creation of unsolicited BFD IP single-hop sessions globally, i.e. on all interfaces. See Section 7.1.
- \* data nodes local-multiplier, desired-min-tx-interval, required-min-rx-interval and min-interval all impact the parameters of the unsolicited BFD IP single-hop sessions.

/routing/control-plane-protocols/control-plane-protocol/bfd/ip-sh  
/interfaces/interface/unsolicited:

- \* data node "enabled" enables creation of unsolicited BFD IP single-hop sessions on a specific interface. See Section 7.1.
- \* data nodes local-multiplier, desired-min-tx-interval, required-min-rx-interval and min-interval all impact the parameters of the unsolicited BFD IP single-hop sessions on the interface.

Some of the readable data nodes in this YANG module may be considered sensitive or vulnerable in some network environments. It is thus important to control read access (e.g., via get, get-config, or notification) to these data nodes. These are the subtrees and data nodes and their sensitivity/vulnerability:

/routing/control-plane-protocols/control-plane-protocol/bfd/ip-sh  
/sessions/session/unsolicited: access to this information discloses the role of the local system in the creation of the unsolicited BFD session.

## 8. References

### 8.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.17487/RFC3688, January 2004, <<https://www.rfc-editor.org/info/rfc3688>>.
- [RFC5082] Gill, V., Heasley, J., Meyer, D., Savola, P., Ed., and C. Pignataro, "The Generalized TTL Security Mechanism (GTSM)", RFC 5082, DOI 10.17487/RFC5082, October 2007, <<https://www.rfc-editor.org/info/rfc5082>>.
- [RFC5880] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD)", RFC 5880, DOI 10.17487/RFC5880, June 2010, <<https://www.rfc-editor.org/info/rfc5880>>.

- [RFC5881] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD) for IPv4 and IPv6 (Single Hop)", RFC 5881, DOI 10.17487/RFC5881, June 2010, <<https://www.rfc-editor.org/info/rfc5881>>.
- [RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, DOI 10.17487/RFC6020, October 2010, <<https://www.rfc-editor.org/info/rfc6020>>.
- [RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <<https://www.rfc-editor.org/info/rfc6241>>.
- [RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure Shell (SSH)", RFC 6242, DOI 10.17487/RFC6242, June 2011, <<https://www.rfc-editor.org/info/rfc6242>>.
- [RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017, <<https://www.rfc-editor.org/info/rfc8040>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [RFC8340] Bjorklund, M. and L. Berger, Ed., "YANG Tree Diagrams", BCP 215, RFC 8340, DOI 10.17487/RFC8340, March 2018, <<https://www.rfc-editor.org/info/rfc8340>>.
- [RFC8341] Bierman, A. and M. Bjorklund, "Network Configuration Access Control Model", STD 91, RFC 8341, DOI 10.17487/RFC8341, March 2018, <<https://www.rfc-editor.org/info/rfc8341>>.
- [RFC8349] Lhotka, L., Lindem, A., and Y. Qu, "A YANG Data Model for Routing Management (NMDA Version)", RFC 8349, DOI 10.17487/RFC8349, March 2018, <<https://www.rfc-editor.org/info/rfc8349>>.
- [RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <<https://www.rfc-editor.org/info/rfc8446>>.

- [RFC9127] Rahman, R., Ed., Zheng, L., Ed., Jethanandani, M., Ed., Pallagatti, S., and G. Mirsky, "YANG Data Model for Bidirectional Forwarding Detection (BFD)", RFC 9127, DOI 10.17487/RFC9127, October 2021, <<https://www.rfc-editor.org/info/rfc9127>>.

## 8.2. Informative References

- [I-D.ietf-idr-rs-bfd]  
Bush, R., Haas, J., Scudder, J. G., Nipper, A., and C. Dietzel, "Making Route Servers Aware of Data Link Failures at IXPs", Work in Progress, Internet-Draft, draft-ietf-idr-rs-bfd-09, 21 September 2020, <<https://www.ietf.org/archive/id/draft-ietf-idr-rs-bfd-09.txt>>.
- [RFC4271] Rekhter, Y., Ed., Li, T., Ed., and S. Hares, Ed., "A Border Gateway Protocol 4 (BGP-4)", RFC 4271, DOI 10.17487/RFC4271, January 2006, <<https://www.rfc-editor.org/info/rfc4271>>.
- [RFC7880] Pignataro, C., Ward, D., Akiya, N., Bhatia, M., and S. Pallagatti, "Seamless Bidirectional Forwarding Detection (S-BFD)", RFC 7880, DOI 10.17487/RFC7880, July 2016, <<https://www.rfc-editor.org/info/rfc7880>>.
- [RFC7911] Walton, D., Retana, A., Chen, E., and J. Scudder, "Advertisement of Multiple Paths in BGP", RFC 7911, DOI 10.17487/RFC7911, July 2016, <<https://www.rfc-editor.org/info/rfc7911>>.
- [RFC7947] Jasinska, E., Hilliard, N., Raszuk, R., and N. Bakker, "Internet Exchange BGP Route Server", RFC 7947, DOI 10.17487/RFC7947, September 2016, <<https://www.rfc-editor.org/info/rfc7947>>.
- [RFC8342] Bjorklund, M., Schoenwaelder, J., Shafer, P., Watsen, K., and R. Wilton, "Network Management Datastore Architecture (NMDA)", RFC 8342, DOI 10.17487/RFC8342, March 2018, <<https://www.rfc-editor.org/info/rfc8342>>.

## Authors' Addresses

Enke Chen  
Palo Alto Networks

Email: [enchen@paloaltonetworks.com](mailto:enchen@paloaltonetworks.com)

Naiming Shen  
Zededa

Email: [naiming@zededa.com](mailto:naiming@zededa.com)

Robert Raszuk  
NTT Network Innovations  
940 Stewart Dr  
Sunnyvale, CA 94085  
United States of America

Email: [robert@raszuk.net](mailto:robert@raszuk.net)

Reshad Rahman  
Canada

Email: [reshad@yahoo.com](mailto:reshad@yahoo.com)

BFD  
Internet-Draft  
Intended status: Informational  
Expires: April 29, 2021

S. Pallagatti, Ed.  
VMware  
G. Mirsky, Ed.  
ZTE Corp.  
S. Paragiri  
Individual Contributor  
V. Govindan  
M. Mudigonda  
Cisco  
October 26, 2020

BFD for VXLAN  
draft-ietf-bfd-vxlan-16

## Abstract

This document describes the use of the Bidirectional Forwarding Detection (BFD) protocol in point-to-point Virtual eXtensible Local Area Network (VXLAN) tunnels used to form an overlay network.

## 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 <https://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 April 29, 2021.

## Copyright Notice

Copyright (c) 2020 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 (<https://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. Conventions Used in this Document . . . . .	3
2.1. Acronyms . . . . .	3
2.2. Requirements Language . . . . .	4
3. Deployment . . . . .	4
4. Use of the Management VNI . . . . .	5
5. BFD Packet Transmission over VXLAN Tunnel . . . . .	6
6. Reception of BFD Packet from VXLAN Tunnel . . . . .	8
7. Echo BFD . . . . .	8
8. IANA Considerations . . . . .	8
9. Security Considerations . . . . .	9
10. Contributors . . . . .	9
11. Acknowledgments . . . . .	9
12. References . . . . .	10
12.1. Normative References . . . . .	10
12.2. Informational References . . . . .	10
Authors' Addresses . . . . .	11

## 1. Introduction

"Virtual eXtensible Local Area Network" (VXLAN) [RFC7348] provides an encapsulation scheme that allows building an overlay network by decoupling the address space of the attached virtual hosts from that of the network.

One use of VXLAN is in data centers interconnecting virtual machines (VMs) of a tenant. VXLAN addresses requirements of the Layer 2 and Layer 3 data center network infrastructure in the presence of VMs in a multi-tenant environment by providing a Layer 2 overlay scheme on a Layer 3 network [RFC7348]. Another use is as an encapsulation for Ethernet VPN [RFC8365].

This document is written assuming the use of VXLAN for virtualized hosts and refers to VMs and VXLAN Tunnel End Points (VTEPs) in hypervisors. However, the concepts are equally applicable to non-virtualized hosts attached to VTEPs in switches.

In the absence of a router in the overlay, a VM can communicate with another VM only if they are on the same VXLAN segment. VMs are unaware of VXLAN tunnels as a VXLAN tunnel is terminated on a VTEP.



VTEPs are responsible for encapsulating and decapsulating frames exchanged among VMs.

The ability to monitor path continuity, i.e., perform proactive continuity check (CC) for point-to-point (p2p) VXLAN tunnels, is important. The asynchronous mode of BFD, as defined in [RFC5880], is used to monitor a p2p VXLAN tunnel.

In the case where a Multicast Service Node (MSN) (as described in Section 3.3 of [RFC8293]) participates in VXLAN, the mechanisms described in this document apply and can, therefore, be used to test the continuity of the path between the source NVE and the MSN.

This document describes the use of Bidirectional Forwarding Detection (BFD) protocol to enable monitoring continuity of the path between VXLAN VTEPs that are performing as Network Virtualization Endpoints, and/or between the source NVE and a replicator MSN using a Management VNI (Section 4). All other uses of the specification to test toward other VXLAN endpoints are out of the scope.

## 2. Conventions Used in this Document

### 2.1. Acronyms

BFD Bidirectional Forwarding Detection

CC Continuity Check

p2p Point-to-point

MSN Multicast Service Node

NVE Network Virtualization Endpoint

VFI Virtual Forwarding Instance

VM Virtual Machine

VNI VXLAN Network Identifier (or VXLAN Segment ID)

VTEP VXLAN Tunnel End Point

VXLAN Virtual eXtensible Local Area Network

## 2.2. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

## 3. Deployment

Figure 1 illustrates the scenario with two servers, each of them hosting two VMs. The servers host VTEPs that terminate two VXLAN tunnels with VXLAN Network Identifier (VNI) number 100 and 200 respectively. Separate BFD sessions can be established between the VTEPs (IP1 and IP2) for monitoring each of the VXLAN tunnels (VNI 100 and 200). Using a BFD session to monitor a set of VXLAN VNIs between the same pair of VTEPs might help to detect and localize problems caused by misconfiguration. An implementation that supports this specification MUST be able to control the number of BFD sessions that can be created between the same pair of VTEPs. This method is applicable whether the VTEP is a virtual or physical device.

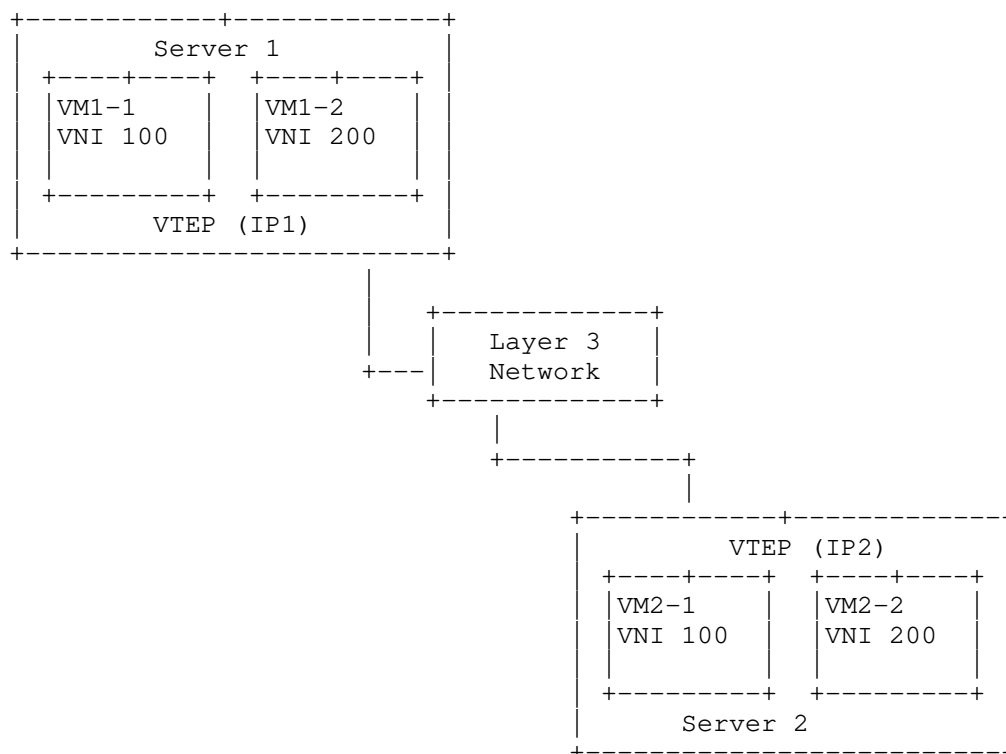


Figure 1: Reference VXLAN Domain

At the same time, a service layer BFD session may be used between the tenants of VTEPs IP1 and IP2 to provide end-to-end fault management (this use case is outside the scope of this document). In such a case, for VTEPs, the BFD Control packets of that session are indistinguishable from data packets.

For BFD Control packets encapsulated in VXLAN (Figure 2), the inner destination IP address SHOULD be set to one of the loopback addresses from 127/8 range for IPv4 or to one of IPv4-mapped IPv6 loopback addresses from `::ffff:127.0.0.0/104` range for IPv6.

#### 4. Use of the Management VNI

In most cases, a single BFD session is sufficient for the given VTEP to monitor the reachability of a remote VTEP, regardless of the number of VNIs. BFD control messages MUST be sent using the Management VNI which acts as the control and management channel between VTEPs. An implementation MAY support operating BFD on

another (non-Management) VNI although the implications of this are outside the scope of this document. The selection of the VNI number of the Management VNI MUST be controlled through a management plane. An implementation MAY use VNI number 1 as the default value for the Management VNI. All VXLAN packets received on the Management VNI MUST be processed locally and MUST NOT be forwarded to a tenant.

## 5. BFD Packet Transmission over VXLAN Tunnel

BFD packets MUST be encapsulated and sent to a remote VTEP as explained in this section. Implementations SHOULD ensure that the BFD packets follow the same forwarding path as VXLAN data packets within the sender system.

BFD packets are encapsulated in VXLAN as described below. The VXLAN packet format is defined in Section 5 of [RFC7348]. The value in the VNI field of the VXLAN header MUST be set to the value selected as the Management VNI. The Outer IP/UDP and VXLAN headers MUST be encoded by the sender as defined in [RFC7348].

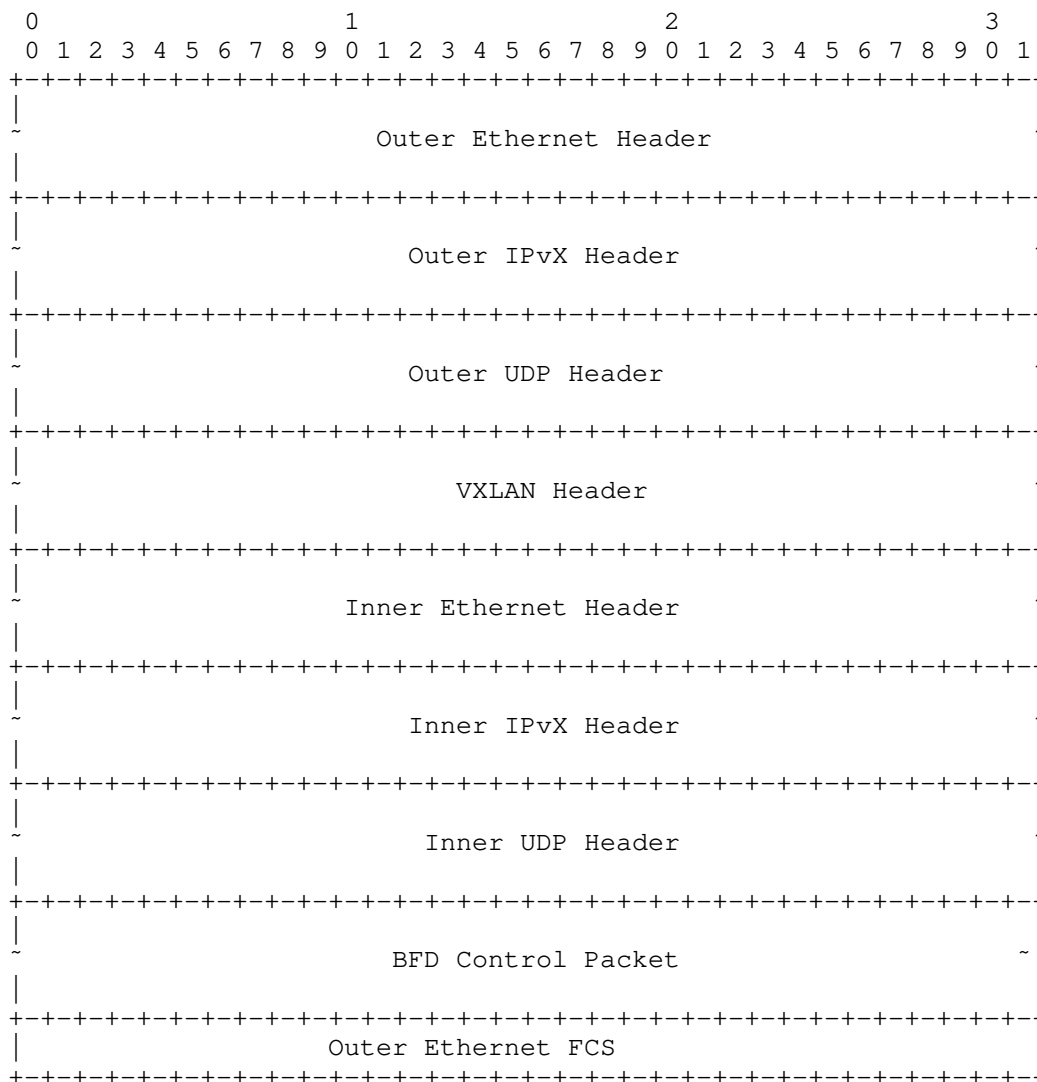


Figure 2: VXLAN Encapsulation of BFD Control Packet

The BFD packet MUST be carried inside the inner Ethernet frame of the VXLAN packet. The choice of Destination MAC and Destination IP addresses for the inner Ethernet frame MUST ensure that the BFD Control packet is not forwarded to a tenant but is processed locally at the remote VTEP. The inner Ethernet frame carrying the BFD Control packet- has the following format:

Ethernet Header:

Destination MAC: A Management VNI, which does not have any tenants, will have no dedicated MAC address for decapsulated traffic. The value (TBD1) SHOULD be used in this field.

Source MAC: MAC address associated with the originating VTEP.

Ethertype: is set to 0x0800 if the inner IP header is IPv4, and is set to 0x86DD if the inner IP header is IPv6.

IP header:

Destination IP: IP address MUST NOT be of one of tenant's IP addresses. The IP address SHOULD be selected from the range 127/8 for IPv4, for IPv6 - from the range ::ffff:127.0.0.0/104. Alternatively, the destination IP address MAY be set to VTEP's IP address.

Source IP: IP address of the originating VTEP.

TTL or Hop Limit: MUST be set to 255 in accordance with [RFC5881].

The fields of the UDP header and the BFD Control packet are encoded as specified in [RFC5881].

## 6. Reception of BFD Packet from VXLAN Tunnel

Once a packet is received, the VTEP MUST validate the packet. If the packet is received on the management VNI and is identified as BFD control packet addressed to the VTEP, and then the packet can be processed further. Processing of BFD control packets received on non-management VNI is outside the scope of this specification.

The received packet's inner IP payload is then validated according to Sections 4 and 5 in [RFC5881].

## 7. Echo BFD

Support for echo BFD is outside the scope of this document.

## 8. IANA Considerations

IANA is requested to assign a single MAC address to the value TBD1 from the "IANA Unicast 48-bit MAC Address" registry from the "Unassigned (small allocations)" block. The Usage field will be "BFD for VXLAN" with a Reference field of this document.

## 9. Security Considerations

Security issues discussed in [RFC5880], [RFC5881], and [RFC7348] apply to this document.

This document recommends using an address from the Internal host loopback addresses 127/8 range for IPv4 or an IP4-mapped IPv6 loopback address from ::ffff:127.0.0.0/104 range for IPv6 as the destination IP address in the inner IP header. Using such an address prevents the forwarding of the encapsulated BFD control message by a transient node in case the VXLAN tunnel is broken as according to [RFC1812].

A router SHOULD NOT forward, except over a loopback interface, any packet that has a destination address on network 127. A router MAY have a switch that allows the network manager to disable these checks. If such a switch is provided, it MUST default to performing the checks.

The use of IPv4-mapped IPv6 addresses has the same property as using the IPv4 network 127/8, moreover, the IPv4-mapped IPv6 addresses prefix is not advertised in any routing protocol.

If the implementation supports establishing multiple BFD sessions between the same pair of VTEPs, there SHOULD be a mechanism to control the maximum number of such sessions that can be active at the same time.

## 10. Contributors

Reshad Rahman  
rrahman@cisco.com  
Cisco

## 11. Acknowledgments

Authors would like to thank Jeff Haas of Juniper Networks for his reviews and feedback on this material.

Authors would also like to thank Nobo Akiya, Marc Binderberger, Shahram Davari, Donald E. Eastlake 3rd, Anoop Ghanwani, Dinesh Dutt, Joel Halpern, and Carlos Pignataro for the extensive reviews and the most detailed and constructive comments.

## 12. References

### 12.1. Normative References

- [RFC1812] Baker, F., Ed., "Requirements for IP Version 4 Routers", RFC 1812, DOI 10.17487/RFC1812, June 1995, <<https://www.rfc-editor.org/info/rfc1812>>.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC5880] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD)", RFC 5880, DOI 10.17487/RFC5880, June 2010, <<https://www.rfc-editor.org/info/rfc5880>>.
- [RFC5881] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD) for IPv4 and IPv6 (Single Hop)", RFC 5881, DOI 10.17487/RFC5881, June 2010, <<https://www.rfc-editor.org/info/rfc5881>>.
- [RFC7348] Mahalingam, M., Dutt, D., Duda, K., Agarwal, P., Kreeger, L., Sridhar, T., Bursell, M., and C. Wright, "Virtual eXtensible Local Area Network (VXLAN): A Framework for Overlaying Virtualized Layer 2 Networks over Layer 3 Networks", RFC 7348, DOI 10.17487/RFC7348, August 2014, <<https://www.rfc-editor.org/info/rfc7348>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.

### 12.2. Informational References

- [RFC8293] Ghanwani, A., Dunbar, L., McBride, M., Bannai, V., and R. Krishnan, "A Framework for Multicast in Network Virtualization over Layer 3", RFC 8293, DOI 10.17487/RFC8293, January 2018, <<https://www.rfc-editor.org/info/rfc8293>>.
- [RFC8365] Sajassi, A., Ed., Drake, J., Ed., Bitar, N., Shekhar, R., Uttaro, J., and W. Henderickx, "A Network Virtualization Overlay Solution Using Ethernet VPN (EVPN)", RFC 8365, DOI 10.17487/RFC8365, March 2018, <<https://www.rfc-editor.org/info/rfc8365>>.



Authors' Addresses

Santosh Pallagatti (editor)  
VMware

Email: santosh.pallagatti@gmail.com

Greg Mirsky (editor)  
ZTE Corp.

Email: gregimirsky@gmail.com

Sudarsan Paragiri  
Individual Contributor

Email: sudarsan.225@gmail.com

Vengada Prasad Govindan  
Cisco

Email: venggovi@cisco.com

Mallik Mudigonda  
Cisco

Email: mmudigon@cisco.com

MPLS Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: April 29, 2021

T. Saad  
Juniper Networks  
K. Raza  
R. Gandhi  
Cisco Systems Inc  
X. Liu  
Volta Networks  
V. Beeram  
Juniper Networks  
October 26, 2020

A YANG Data Model for MPLS Base  
draft-ietf-mpls-base-yang-17

## Abstract

This document contains a specification of the MPLS base YANG data model. The MPLS base YANG data model serves as a base framework for configuring and managing an MPLS switching subsystem on an MPLS-enabled router. It is expected that other MPLS YANG data models (e.g. MPLS Label Switched Path (LSP) Static, LDP or RSVP-TE YANG models) will augment the MPLS base YANG data model.

## 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 <https://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 April 29, 2021.

## Copyright Notice

Copyright (c) 2020 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

(<https://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
1.1. Terminology . . . . .	3
1.2. Acronyms and Abbreviations . . . . .	3
2. MPLS Base Model . . . . .	4
2.1. Model Overview . . . . .	4
2.2. Model Organization . . . . .	4
2.3. Model Design . . . . .	6
2.4. Model Tree Diagram . . . . .	8
2.5. Model YANG Module . . . . .	9
3. IANA Considerations . . . . .	19
4. Security Considerations . . . . .	20
5. Acknowledgement . . . . .	21
6. Appendix A. Data Tree Instance Example . . . . .	21
7. Contributors . . . . .	27
8. References . . . . .	27
8.1. Normative References . . . . .	27
8.2. Informative References . . . . .	29
Authors' Addresses . . . . .	29

## 1. Introduction

A core routing YANG data model is defined in [RFC8349], and it provides a basis for the development of routing data models for specific Address Families (AFs). Specifically, [RFC8349] defines a model for a generic Routing Information Base (RIB) that is Address-Family (AF) agnostic. [RFC8349] also defines two instances of RIBs based on the generic RIB model for IPv4 and IPv6 AFs.

The MPLS base model that is defined in this document augments the generic RIB model defined in [RFC8349] with additional data that enables MPLS forwarding for the specific destination prefix(es) present in the AF RIB(s) as described in the MPLS architecture document [RFC3031].

The MPLS base model also defines a new instance of the generic RIB YANG data model as defined in [RFC8349] to store native MPLS routes. The native MPLS RIB instance stores route(s) that are not associated with other AF instance RIBs (such as IPv4, or IPv6 instance RIB(s)),

but are enabled for MPLS forwarding. Examples of such native MPLS routes are routes programmed by RSVP on transit MPLS router(s) along the path of a Label Switched Path (LSP). Other example(s) are MPLS routes that cross-connect to specific Layer-2 adjacencies, such as Layer-2 Attachment Circuit(s) (ACs)), or Layer-3 adjacencies, such as Segment-Routing (SR) Adjacency Segments (Adj-SIDs) described in [RFC8402].

The MPLS base YANG data model serves as a basis for future development of MPLS YANG data models covering more-sophisticated MPLS feature(s) and sub-system(s). The main purpose is to provide essential building blocks for other YANG data models involving different control-plane protocols, and MPLS functions.

To this end, it is expected that the MPLS base data model will be augmented by a number of other YANG modules developed at IETF (e.g. by TEAS and MPLS working groups).

The YANG module in this document conforms to the Network Management Datastore Architecture (NMDA) [RFC8342].

### 1.1. Terminology

The terminology for describing YANG data models is found in [RFC7950].

### 1.2. Acronyms and Abbreviations

MPLS: Multiprotocol Label Switching

RIB: Routing Information Base

LSP: Label Switched Path

LSR: Label Switching Router

LER: Label Edge Router

FEC: Forwarding Equivalence Class

NHLFE: Next Hop Label Forwarding Entry

ILM: Incoming Label Map

## 2. MPLS Base Model

This document describes the 'ietf-mpls' YANG module that provides base components of the MPLS data model. It is expected that other MPLS YANG modules will augment 'ietf-mpls' YANG module for other MPLS extension to provision Label Switched Paths (LSPs) (e.g. MPLS Static, MPLS LDP or MPLS RSVP-TE LSP(s)).

### 2.1. Model Overview

This document models MPLS labeled routes as an augmentation of the generic routing RIB data model as defined in [RFC8349]. For example, IP prefix routes (e.g. routes stored in IPv4 or IPv6 RIBs) are augmented to carry additional data to enable it for MPLS forwarding.

This document also defines a new instance of the generic RIB defined in [RFC8349] to store native MPLS route(s) (described further in Section 2.3) by extending the identity 'address-family' defined in [RFC8349] with a new "mpls" identity as suggested in Section 3 of [RFC8349].

### 2.2. Model Organization

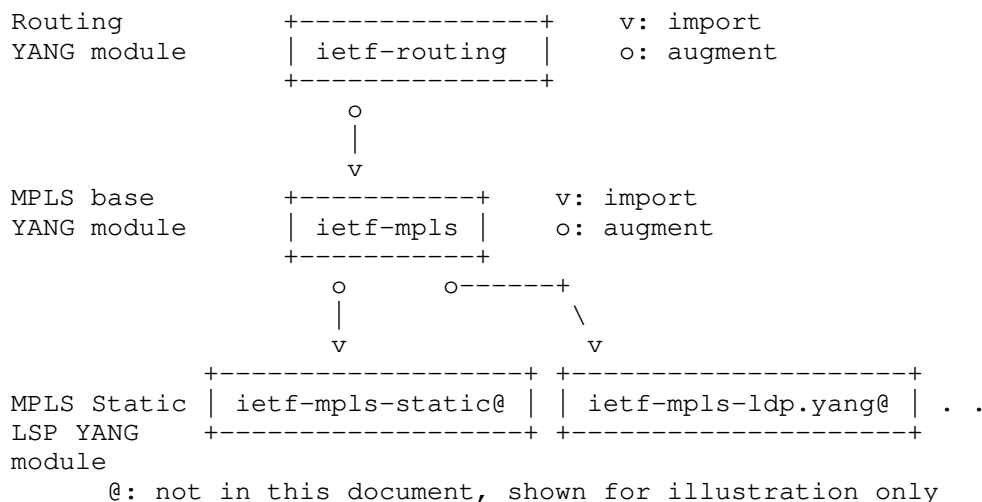


Figure 1: Relationship between MPLS modules

The 'ietf-mpls' YANG module defines the following identities:

mpls:

This identity extends the 'address-family' identity for RIB instance(s) identity as defined in [RFC8349] to represent the native MPLS RIB instance.

#### label-block-alloc-mode:

A base YANG identity for supported label block allocation mode(s).

The 'ietf-mpls' YANG module contains the following high-level types and groupings:

#### mpls-operations-type:

An enumeration type that represents support for possible MPLS operation types (impose-and-forward, pop-and-forward, pop-impose-and-forward, and pop-and-lookup)

#### nhlfe-role:

An enumeration type that represents the role of the NHLFE entry.

#### nhlfe-single-contents:

A YANG grouping that describes single Next Hop Label Forwarding Entry (NHLFE) and its associated parameters as described in the MPLS architecture document [RFC3031]. This grouping is specific to the case when a single next-hop is associated with the route.

The NHLFE is used when forwarding labeled packet. It contains the following information:

1. the packet's next hop. For 'nhlfe-single-contents' only a single next hop is expected, while for 'nhlfe-multiple-contents' multiple next hops are possible.
2. the operation to perform on the packet's label stack; this can be one of the following operations: a) replace the label at the top of the label stack with one or more specified new label b) pop the label stack c) replace the label at the top of the label stack with a specified new label, and then push one or more specified new labels onto the label stack. d) push one or more label(s) on an unlabeled packet

It may also contain:

- d) the data link encapsulation to use when transmitting the packet
- e) the way to encode the label stack when transmitting the packet
- f) any other information needed in order to properly dispose of the packet.

#### nhlfe-multiple-contents:

A YANG grouping that describes a set of NHLFE(s) and their associated parameters as described in the MPLS architecture document [RFC3031]. This grouping is used when multiple next-hops are associated with the route.

#### interfaces-mpls:

A YANG grouping that describes the list of MPLS enabled interfaces on a device.

#### label-blocks:

A YANG grouping that describes the list of assigned MPLS label blocks and their properties.

#### rib-mpls-properties:

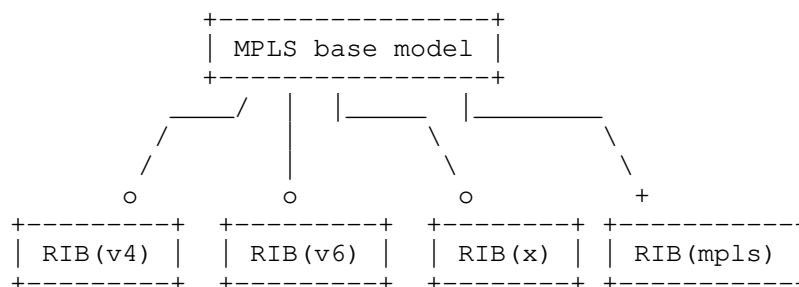
A YANG grouping for the augmentation of the generic RIB with MPLS label forwarding data as defined in [RFC3031].

#### rib-active-route-mpls-input:

A YANG grouping for the augmentation to the 'active-route' RPC that is specific to the MPLS RIB instance.

### 2.3. Model Design

The MPLS routing model is based on the core routing data model defined in [RFC8349]. Figure 2 shows the extensions introduced by the MPLS base model on defined RIB(s).



+: created by the MPLS base model  
 o: augmented by the MPLS base model

Figure 2: Relationship between MPLS model and RIB instances

As shown in Figure 2, the MPLS base YANG data model augments defined instance(s) of AF RIB(s) with additional data that enables MPLS forwarding for destination prefix(es) store in such RIB(s). For example, an IPv4 prefix stored in RIB(v4) is augmented to carry a MPLS local label and per next-hop remote label(s) to enable MPLS forwarding for such prefix.

The MPLS base model also creates a separate instance of the generic RIB model defined in [RFC8349] to store MPLS native route(s) that are enabled for MPLS forwarding, but not stored in other AF RIB(s).

Some examples of such native MPLS routes are:

- o routes programmed by RSVP on Label Switched Router(s) (LSRs) along the path of a Label Switched Path (LSP),
- o routes that cross-connect an MPLS local label to a Layer-2, or Layer-3 VRF,
- o routes that cross-connect an MPLS local label to a specific Layer-2 adjacency or interface, such as Layer-2 Attachment Circuit(s) (ACs), or
- o routes that cross-connect an MPLS local label to a Layer-3 adjacency or interface - such as MPLS Segment-Routing (SR) Adjacency Segments (Adj-SIDs), SR MPLS Binding SIDs, etc. as defined in [RFC8402].



## 2.4. Model Tree Diagram

The MPLS base tree diagram that follows the notation defined in [RFC8340] is shown in Figure 3.

```

module: ietf-mpls
  augment /rt:routing:
    +--rw mpls
      +--rw ttl-propagate?          boolean
      +--rw mpls-label-blocks
        +--rw mpls-label-block* [index]
          +--rw index                string
          +--rw start-label?         rt-types:mpls-label
          +--rw end-label?           rt-types:mpls-label
          +--rw block-allocation-mode? identityref
          +--ro inuse-labels-count?  yang:gauge32
      +--rw interfaces
        +--rw interface* [name]
          +--rw name                  if:interface-ref
          +--rw mpls-enabled?         boolean
          +--rw maximum-labeled-packet? uint32
  augment /rt:routing/rt:ribs/rt:rib/rt:routes/rt:route:
    +--ro mpls-enabled?              boolean
    +--ro mpls-local-label?          rt-types:mpls-label
    +--ro destination-prefix?        -> ../mpls-local-label
    +--ro route-context?             string
  augment /rt:routing/rt:ribs/rt:rib/rt:routes/rt:route/rt:next-hop
    /rt:next-hop-options/rt:simple-next-hop:
    +--ro mpls-label-stack
      +--ro entry* [id]
        +--ro id                     uint8
        +--ro label?                 rt-types:mpls-label
        +--ro ttl?                   uint8
        +--ro traffic-class?         uint8
  augment /rt:routing/rt:ribs/rt:rib/rt:routes/rt:route/rt:next-hop
    /rt:next-hop-options/rt:next-hop-list/rt:next-hop-list
    /rt:next-hop:
    +--ro index?                     string
    +--ro backup-index?              string
    +--ro loadshare?                 uint16
    +--ro role?                      nhlfe-role
    +--ro mpls-label-stack
      +--ro entry* [id]
        +--ro id                     uint8
        +--ro label?                 rt-types:mpls-label
        +--ro ttl?                   uint8
        +--ro traffic-class?         uint8
  augment /rt:routing/rt:ribs/rt:rib/rt:active-route/rt:input:

```

```

    +---w destination-address?    -> ../mpls-local-label
    +---w mpls-local-label?       rt-types:mpls-label
augment /rt:routing/rt:ribs/rt:rib/rt:active-route/rt:output
    /rt:route/rt:next-hop/rt:next-hop-options
    /rt:simple-next-hop:
+--- mpls-label-stack
    +--- entry* [id]
        +--- id                  uint8
        +--- label?              rt-types:mpls-label
        +--- ttl?                uint8
        +--- traffic-class?      uint8
augment /rt:routing/rt:ribs/rt:rib/rt:active-route/rt:output
    /rt:route/rt:next-hop/rt:next-hop-options
    /rt:next-hop-list/rt:next-hop-list/rt:next-hop:
+--- index?                      string
+--- backup-index?              string
+--- loadshare?                 uint16
+--- role?                      nhlfe-role
+--- mpls-label-stack
    +--- entry* [id]
        +--- id                  uint8
        +--- label?              rt-types:mpls-label
        +--- ttl?                uint8
        +--- traffic-class?      uint8

```

Figure 3: MPLS Base tree diagram

## 2.5. Model YANG Module

This section describes the 'ietf-mpls' YANG module that provides base components of the MPLS data model. Other YANG module(s) may import and augment the base MPLS module to add feature specific data.

The ietf-mpls YANG module imports the following YANG modules:

- o ietf-routing defined in [RFC8349]
- o ietf-routing-types defined in [RFC8294]
- o ietf-interfaces defined in [RFC8343]

This YANG module also references the following RFCs in defining the types and YANG grouping of the YANG module: [RFC3032], [RFC3031], and [RFC7424].

```

<CODE BEGINS> file "ietf-mpls@2020-10-26.yang"
module ietf-mpls {
  yang-version 1.1;

```

```
namespace "urn:ietf:params:xml:ns:yang:ietf-mpls";

/* Replace with IANA when assigned */

prefix mpls;

import ietf-routing {
  prefix rt;
  reference
    "RFC8349: A YANG Data Model for Routing Management";
}
import ietf-routing-types {
  prefix rt-types;
  reference
    "RFC8294: Common YANG Data Types for the Routing Area";
}
import ietf-yang-types {
  prefix yang;
  reference
    "RFC6991: Common YANG Data Types";
}
import ietf-interfaces {
  prefix if;
  reference
    "RFC8343: A YANG Data Model for Interface Management";
}

organization
  "IETF MPLS Working Group";
contact
  "WG Web:    <http://tools.ietf.org/wg/mpls/>

  WG List:    <mailto:mpls@ietf.org>

  Editor:     Tarek Saad
               <mailto:tsaad@juniper.net>

  Editor:     Kamran Raza
               <mailto:skraza@cisco.com>

  Editor:     Rakesh Gandhi
               <mailto:rgandhi@cisco.com>

  Editor:     Xufeng Liu
               <mailto: xufeng.liu.ietf@gmail.com>

  Editor:     Vishnu Pavan Beeram
               <mailto:vbeeram@juniper.net>";
```

## description

"This YANG module defines the essential components for the management of the MPLS subsystem. The model fully conforms to the Network Management Datastore Architecture (NMDA).

Copyright (c) 2018 IETF Trust and the persons identified as authors of the code. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, is permitted pursuant to, and subject to the license terms contained in, the Simplified BSD License set forth in Section 4.c of the IETF Trust's Legal Provisions Relating to IETF Documents

(<https://trustee.ietf.org/license-info>).

This version of this YANG module is part of RFC XXXX; see the RFC itself for full legal notices.";

// RFC Ed.: replace XXXX with actual RFC number and remove this  
// note.  
// RFC Ed.: update the date below with the date of RFC publication  
// and remove this note.

revision 2020-10-26 {

description

"Initial revision.";

reference

"RFC XXXX: A YANG Data Model for base MPLS";

}

/\* Identities \*/

identity mpls {

base rt:address-family;

description

"This identity represents the MPLS address family.";

}

identity mpls-unicast {

base mpls:mpls;

description

"This identity represents the MPLS unicast address family.";

}

identity label-block-alloc-mode {

description

"Base identity for label-block allocation mode.";

}

```
identity label-block-alloc-mode-manager {
  base label-block-alloc-mode;
  description
    "Label block allocation on reserved block
    is managed by label manager.";
}

identity label-block-alloc-mode-application {
  base label-block-alloc-mode;
  description
    "Label block allocation on reserved block
    is managed by application.";
}

/**
 * Typedefs
 */

typedef mpls-operations-type {
  type enumeration {
    enum impose-and-forward {
      description
        "Operation impose outgoing label(s) and forward to
        next-hop.";
    }
    enum pop-and-forward {
      description
        "Operation pop incoming label and forward to next-hop.";
    }
    enum pop-impose-and-forward {
      description
        "Operation pop incoming label, impose one or more
        outgoing label(s) and forward to next-hop.";
    }
    enum swap-and-forward {
      description
        "Operation swap incoming label, with outgoing label and
        forward to next-hop.";
    }
    enum pop-and-lookup {
      description
        "Operation pop incoming label and perform a lookup.";
    }
  }
  description
    "MPLS operations types.";
}
```

```
typedef nhlfe-role {
  type enumeration {
    enum primary {
      description
        "Next-hop acts as primary for carrying traffic.";
    }
    enum backup {
      description
        "Next-hop acts as backup.";
    }
    enum primary-and-backup {
      description
        "Next-hop acts as primary and backup simultaneously
        for carry traffic.";
    }
  }
  description
    "The next-hop role.";
}

grouping nhlfe-single-contents {
  description
    "A grouping that describes single Next Hop Label Forwarding
    Entry (NHLFE) and its associated parameters as described in
    the MPLS architecture. This grouping is specific to the case
    when a single next-hop is associated with the route.";
  uses rt-types:mpls-label-stack;
}

grouping nhlfe-multiple-contents {
  description
    "A grouping that describes a set of NHLFE(s) and their
    associated parameters as described in the MPLS architecture.
    This grouping is used when multiple next-hops are associated
    with the route.";
  leaf index {
    type string;
    description
      "A user-specified identifier utilised to uniquely
      reference the next-hop entry in the next-hop list.
      The value of this index has no semantic meaning
      other than for referencing the entry.";
  }
  leaf backup-index {
    type string;
    description
      "A user-specified identifier utilised to uniquely
      reference the backup next-hop entry in the NHLFE list."
  }
}
```

```
        The value of this index has no semantic meaning
        other than for referencing the entry.";
    reference
        "RFC4090 and RFC5714";
}
leaf loadshare {
    type uint16;
    default "1";
    description
        "This value is used to compute a loadshare to perform un-equal
        load balancing when multiple outgoing next-hop(s) are
        specified. A share is computed as a ratio of this number to the
        total under all next-hops(s).";
    reference
        "RFC7424, section 5.4,
        RFC3031, section 3.11 and 3.12.";
}
leaf role {
    type nhlfe-role;
    description
        "NHLFE role.";
}
uses nhlfe-single-contents;
}

grouping interfaces-mpls {
    description
        "List of MPLS interfaces.";
    container interfaces {
        description
            "List of MPLS enabled interaces.";
        list interface {
            key "name";
            description
                "MPLS enabled interface entry.";
            leaf name {
                type if:interface-ref;
                description
                    "A reference to the name of a interface in the system that
                    is to be enabled for MPLS.";
            }
            leaf mpls-enabled {
                type boolean;
                default "false";
                description
                    "'true' if mpls encapsulation is enabled on the interface.
                    'false' if mpls encapsulation is disabled on the
                    interface.";
            }
        }
    }
}
```

```
    }
    leaf maximum-labeled-packet {
        type uint32;
        units "octets";
        description
            "Maximum labeled packet size.";
        reference
            "RFC3032, section 3.2.";
    }
}
}
}

grouping globals {
    description
        "MPLS global configuration grouping.";
    leaf ttl-propagate {
        type boolean;
        default "true";
        description
            "Propagate TTL between IP and MPLS.";
    }
}

grouping label-blocks {
    description
        "Label-block allocation grouping.";
    container mpls-label-blocks {
        description
            "Label-block allocation container.";
        list mpls-label-block {
            key "index";
            description
                "List of MPLS label-blocks.";
            leaf index {
                type string;
                description
                    "A user-specified identifier utilised to uniquely
                     reference an MPLS label block.";
            }
            leaf start-label {
                type rt-types:mpls-label;
                must '. <= ../end-label' {
                    error-message
                        "The start-label must be less than or equal "
                        + "to end-label";
                }
                description

```



```
        "Label-block start.";
    }
    leaf end-label {
        type rt-types:mpls-label;
        must '.. >= ../start-label' {
            error-message
                "The end-label must be greater than or equal "
                + "to start-label";
        }
        description
            "Label-block end.";
    }
    leaf block-allocation-mode {
        type identityref {
            base label-block-alloc-mode;
        }
        description
            "Label-block allocation mode.";
    }
    leaf inuse-labels-count {
        when "derived-from-or-self(../block-allocation-mode, "
            + "'mpls:label-block-alloc-mode-manager')";
        type yang:gauge32;
        config false;
        description
            "Label-block inuse labels count.";
    }
}

grouping rib-mpls-properties {
    description
        "A grouping of native MPLS RIB properties.";
    leaf destination-prefix {
        type leafref {
            path "../mpls-local-label";
        }
        description
            "MPLS destination prefix.";
    }
    leaf route-context {
        type string;
        description
            "A context associated with the native MPLS route.";
    }
}
```

```
grouping rib-active-route-mpls-input {
  description
    "A grouping applicable to native MPLS RIB 'active-route'
    RPC input augmentation.";
  leaf destination-address {
    type leafref {
      path "../mpls-local-label";
    }
    description
      "MPLS native active route destination.";
  }
  leaf mpls-local-label {
    type rt-types:mpls-label;
    description
      "MPLS local label.";
  }
}

augment "/rt:routing" {
  description
    "MPLS augmentation.";
  container mpls {
    description
      "MPLS container, to be used as an augmentation target node
      other MPLS sub-features config, e.g. MPLS static LSP, MPLS
      LDP LSPs, and Traffic Engineering MPLS LSP Tunnels, etc.";
    uses globals;
    uses label-blocks;
    uses interfaces-mpls;
  }
}

/* MPLS routes augmentation */

augment "/rt:routing/rt:ribs/rt:rib/rt:routes/rt:route" {
  description
    "This augmentation is applicable to all MPLS routes.";
  leaf mpls-enabled {
    type boolean;
    default "false";
    description
      "Indicates whether MPLS is enabled for this route.";
  }
  leaf mpls-local-label {
    when "../mpls-enabled = 'true'";
    type rt-types:mpls-label;
    description
      "MPLS local label associated with the route.";
  }
}
```

```
    }
    uses rib-mpls-properties {
      /* MPLS AF augmentation to native MPLS RIB */
      when "derived-from-or-self(..../rt:address-family, "
        + "'mpls:mpls')" {
        description
          "This augment is valid only for routes of native MPLS
          RIB.";
      }
    }
  }
}

/* MPLS simple-next-hop augmentation */

augment "/rt:routing/rt:ribs/rt:rib/rt:routes/rt:route/"
  + "rt:next-hop/rt:next-hop-options/rt:simple-next-hop" {
  description
    "Augment 'simple-next-hop' case in IP unicast routes.";
  uses nhlfe-single-contents {
    when "/rt:routing/rt:ribs/rt:rib/rt:routes/rt:route"
      + "/mpls:mpls-enabled = 'true'";
  }
}

/* MPLS next-hop-list augmentation */

augment "/rt:routing/rt:ribs/rt:rib/rt:routes/rt:route/"
  + "rt:next-hop/rt:next-hop-options/rt:next-hop-list/"
  + "rt:next-hop-list/rt:next-hop" {
  description
    "This leaf augments the 'next-hop-list' case of IP unicast
    routes.";
  uses nhlfe-multiple-contents {
    when "/rt:routing/rt:ribs/rt:rib/rt:routes/rt:route"
      + "/mpls:mpls-enabled = 'true'";
  }
}

/* MPLS RPC input augmentation */

augment "/rt:routing/rt:ribs/rt:rib/rt:active-route/rt:input" {
  description
    "Input MPLS augmentation for the 'active-route' action
    statement.";
  uses rib-active-route-mpls-input {
    /* MPLS AF augmentation to native MPLS RIB */
    when "derived-from-or-self(..../rt:address-family, "
      + "'mpls:mpls')" {

```

```

        description
            "This augment is valid only for routes of native MPLS
            RIB.";
    }
}

/* MPLS RPC output augmentation */

augment "/rt:routing/rt:ribs/rt:rib/rt:active-route/"
    + "rt:output/rt:route/"
    + "rt:next-hop/rt:next-hop-options/rt:simple-next-hop" {
    description
        "Output MPLS augmentation for the 'active-route' action
        statement.";
    uses nhlfe-single-contents;
}

augment "/rt:routing/rt:ribs/rt:rib/rt:active-route/"
    + "rt:output/rt:route/"
    + "rt:next-hop/rt:next-hop-options/rt:next-hop-list/"
    + "rt:next-hop-list/rt:next-hop" {
    description
        "Output MPLS augmentation for the 'active-route' action
        statement.";
    uses nhlfe-multiple-contents;
}
}
<CODE ENDS>

```

Figure 4: MPLS base YANG module.

### 3. IANA Considerations

This document registers the following URIs in the 'ns' sub-registry of the IETF XML registry [RFC3688]. Following the format in [RFC3688], the following registration is requested to be made.

URI: urn:ietf:params:xml:ns:yang:ietf-mpls  
 Registrant Contact: The MPLS WG of the IETF.  
 XML: N/A, the requested URI is an XML namespace.

This document registers a YANG module in the YANG Module Names registry [RFC6020].

```
name:      ietf-mpls
namespace: urn:ietf:params:xml:ns:yang:ietf-mpls
prefix:    mpls
// RFC Ed.: replace XXXX with RFC number and remove this note
reference:  RFCXXXX
```

#### 4. Security Considerations

The YANG module specified in this document define a schema for data that is designed to be accessed via network management protocols such as NETCONF [RFC6241] or RESTCONF [RFC8040]. The lowest NETCONF layer is the secure transport layer, and the mandatory-to-implement secure transport is Secure Shell (SSH) [RFC6242]. The lowest RESTCONF layer is HTTPS, and the mandatory-to-implement secure transport is TLS [RFC8446].

The NETCONF access control model [RFC8341] provides the means to restrict access for particular NETCONF or RESTCONF users to a preconfigured subset of all available NETCONF or RESTCONF protocol operations and content.

There are a number of data nodes defined in this YANG module that are writable/creatable/deletable (i.e., config true, which is the default). These data nodes may be considered sensitive or vulnerable in some network environments. Write operations (e.g., edit-config) to these data nodes without proper protection can have a negative effect on network operations. These are the subtrees and data nodes and their sensitivity/vulnerability:

`"/rt:routing/mpls:mpls/mpls:label-blocks"`: there are data nodes under this path that are writeable such as 'start-label' and 'end-label'. Write operations to those data nodes may cause disruptive action to existing traffic.

Some of the readable data nodes in these YANG module may be considered sensitive or vulnerable in some network environments. It is thus important to control read access (e.g., via get, get-config, or notification) to these data nodes. These are the subtrees and data nodes and their sensitivity/vulnerability:

`"/rt:routing/rt:ribs/rt:rib/rt:routes/rt:route/rt:next-hop/rt:next-hop-options/rt:next-hop-list/rt:next-hop-list/rt:next-hop"` and `"/rt:routing/rt:ribs/rt:rib/rt:active-route/rt:output/rt:route/rt:next-hop/rt:next-hop-options/rt:simple-next-hop"`: these two paths are augmented by additional MPLS leaf(s) defined in this model. Access to this information may disclose the next-hop or path per prefix and/or other information.

Some of the RPC operations in this YANG module may be considered sensitive or vulnerable in some network environments. It is thus important to control access to these operations. These are the operations and their sensitivity/vulnerability:

`"/rt:routing/rt:ribs/rt:rib/rt:active-route/rt:input"` and `"/rt:routing/rt:ribs/rt:rib/rt:active-route/rt:output/rt:route"`: these two paths are augmented by additional MPLS data node(s) that are defined in this model. Access to those path(s) may disclose information about per prefix route and/or other information and that may be further used for further attack(s).

The security considerations spelled out in [RFC3031] and [RFC3032] apply for this document as well.

## 5. Acknowledgement

The authors would like to thank Xia Chen for her contributions to the early revisions of this document.

## 6. Appendix A. Data Tree Instance Example

A simple network setup is shown in Figure 5. R1 runs the ISIS routing protocol, and learns reachability about two IPv4 prefixes: P1: 198.51.100.1/32 and P2: 198.51.100.1/32, and two IPv6 prefixes P3: 2001:db8:0:10::1/64 and P4: 2001:db8:0:10::1/64. We also assume that R1 learns about local and remote MPLS label bindings for each prefix using ISIS (e.g. using Segment-Routing (SR) extensions).

```

State on R1:
=====
      IPv4 Prefix      MPLS Label
P1: 198.51.100.1/32    16001
P2: 198.51.100.2/32    16002

      IPv6 Prefix      MPLS Label
P3: 2001:db8:0:10::1/64 16003
P4: 2001:db8:0:10::2/64 16004

RSVP MPLS LSPv4-Tunnel:
Source:      198.51.100.3
Destination: 198.51.100.4
Tunnel-ID:   10
LSP-ID:      1

                        192.0.2.5/30
                        2001:db8:0:1::1/64
                        eth0
                        +---+
                        /
          +-----+
          |  R1  |
          +-----+
            \
              +---+
              eth1
              192.0.2.13/30
              2001:db8:0:2::1/64

```

Figure 5: Example of network configuration.

The instance data tree could then be as follows:

```

{
  "ietf-routing:routing":{
    "ribs":{
      "rib":[
        {
          "name":"RIB-V4",
          "address-family":
            "ietf-ipv4-unicast-routing:v4ur:ipv4-unicast",
          "routes":{
            "route":[
              {
                "next-hop":{
                  "outgoing-interface":"eth0",
                  "ietf-mpls:mpls-label-stack":{

```

```
    "entry":[
      {
        "id":1,
        "label":16001,
        "ttl":255
      }
    ]
  },
  "ietf-ipv4-unicast-routing:next-hop-address":
    "192.0.2.5"
},
"source-protocol":"isis:isis",
"ietf-mpls:mpls-enabled":true,
"ietf-mpls:mpls-local-label":16001,
"ietf-ipv4-unicast-routing:destination-prefix":
"198.51.100.1/32",
"ietf-mpls:route-context":"SID-IDX:1"
},
{
  "next-hop":{
    "next-hop-list":{
      "next-hop":[
        {
          "outgoing-interface":"eth0",
          "ietf-mpls:index":"1",
          "ietf-mpls:backup-index":"2",
          "ietf-mpls:role":"primary-and-backup",
          "ietf-mpls:mpls-label-stack":{
            "entry":[
              {
                "id":1,
                "label":16002,
                "ttl":255
              }
            ]
          }
        },
        {
          "outgoing-interface":"eth1",
          "ietf-mpls:index":"2",
          "ietf-mpls:backup-index":"1",
          "ietf-mpls:role":"primary-and-backup",
          "ietf-mpls:mpls-label-stack":{
            "entry":[
              {
                "id":1,
                "label":16002,
```



```

        "ttl":255
      }
    ]
  },
  "ietf-ipv4-unicast-routing:address":"192.0.2.13"
}
]
}
},
"source-protocol":"isis:isis",
"ietf-mpls:mpls-enabled":true,
"ietf-mpls:mpls-local-label":16002,
"ietf-ipv4-unicast-routing:destination-prefix":
"198.51.100.2/32",
"ietf-mpls:route-context":"SID-IDX:2"
}
]
}
},
{
  "name":"RIB-V6",
  "address-family":
  "ietf-ipv6-unicast-routing:v6ur:ipv6-unicast",
  "routes":{
    "route":[
      {
        "next-hop":{
          "outgoing-interface":"eth0",
          "ietf-mpls:mpls-label-stack":{
            "entry":[
              {
                "id":1,
                "label":16003,
                "ttl":255
              }
            ]
          },
          "ietf-ipv6-unicast-routing:next-hop-address":
            "2001:db8:0:1::1"
        },
        "source-protocol":"isis:isis",
        "ietf-mpls:mpls-enabled":true,
        "ietf-mpls:mpls-local-label":16001,
        "ietf-ipv6-unicast-routing:destination-prefix":
        "2001:db8:0:10::1/6",
        "ietf-mpls:route-context":"SID-IDX:1"
      },
      {

```

```

    "next-hop":{
      "next-hop-list":{
        "next-hop":[
          {
            "outgoing-interface":"eth0",
            "ietf-mpls:index":"1",
            "ietf-mpls:backup-index":"2",
            "ietf-mpls:role":"primary-and-backup",
            "ietf-mpls:mpls-label-stack":{
              "entry":[
                {
                  "id":1,
                  "label":16004,
                  "ttl":255
                }
              ]
            },
            "ietf-ipv6-unicast-routing:address":
              "2001:db8:0:1::1"
          },
          {
            "outgoing-interface":"eth1",
            "ietf-mpls:index":"2",
            "ietf-mpls:backup-index":"1",
            "ietf-mpls:role":"primary-and-backup",
            "ietf-mpls:mpls-label-stack":{
              "entry":[
                {
                  "id":1,
                  "label":16004,
                  "ttl":255
                }
              ]
            },
            "ietf-ipv6-unicast-routing:address":
              "2001:db8:0:2::1"
          }
        ]
      }
    },
    "source-protocol":"isis:isis",
    "ietf-mpls:mpls-enabled":true,
    "ietf-mpls:mpls-local-label":16004,
    "ietf-ipv6-unicast-routing:destination-prefix":
      "2001:db8:0:10::2/64",
    "ietf-mpls:route-context":"SID-IDX:2"
  }
]

```

```

    }
  },
  {
    "name": "RIB-MPLS",
    "address-family": "ietf-mpls:mpls:mpls",
    "routes": {
      "route": [
        {
          "next-hop": {
            "outgoing-interface": "eth0",
            "ietf-mpls:mpls-label-stack": {
              "entry": [
                {
                  "id": 1,
                  "label": 24002,
                  "ttl": 255
                }
              ]
            }
          },
          "ietf-ipv4-unicast-routing:next-hop-address":
            "192.0.2.5"
        },
        "source-protocol": "rsvp:rsvp",
        "ietf-mpls:mpls-enabled": true,
        "ietf-mpls:mpls-local-label": 24001,
        "ietf-mpls:destination-prefix": "24001",
        "ietf-mpls:route-context":
          "RSVP Src:198.51.100.3,Dst:198.51.100.4,T:10,L:1"
      ]
    }
  }
}

]
},
"ietf-mpls:mpls": {
  "mpls-label-blocks": {
    "mpls-label-block": [
      {
        "index": "mpls-srgb-label-block",
        "start-label": 16000,
        "end-label": 16500,
        "block-allocation-mode": "mpls:label-block-alloc-mode-manager"
      }
    ]
  }
},
"interfaces": {
  "interface": [
    {

```

```
        "name":"eth0",
        "mpls-enabled":true,
        "maximum-labeled-packet":1488
      },
      {
        "name":"eth1",
        "mpls-enabled":true,
        "maximum-labeled-packet":1488
      }
    ]
  }
}
```

Figure 6: Foo bar.

## 7. Contributors

Igor Bryskin  
Huawei Technologies  
email: i\_bryskin@yahoo.com

Himanshu Shah  
Ciena  
email: hshah@ciena.com

## 8. References

### 8.1. Normative References

- [RFC3032] Rosen, E., Tappan, D., Fedorkow, G., Rekhter, Y., Farinacci, D., Li, T., and A. Conta, "MPLS Label Stack Encoding", RFC 3032, DOI 10.17487/RFC3032, January 2001, <<https://www.rfc-editor.org/info/rfc3032>>.
- [RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.17487/RFC3688, January 2004, <<https://www.rfc-editor.org/info/rfc3688>>.
- [RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, DOI 10.17487/RFC6020, October 2010, <<https://www.rfc-editor.org/info/rfc6020>>.

- [RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <<https://www.rfc-editor.org/info/rfc6241>>.
- [RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure Shell (SSH)", RFC 6242, DOI 10.17487/RFC6242, June 2011, <<https://www.rfc-editor.org/info/rfc6242>>.
- [RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", RFC 7950, DOI 10.17487/RFC7950, August 2016, <<https://www.rfc-editor.org/info/rfc7950>>.
- [RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017, <<https://www.rfc-editor.org/info/rfc8040>>.
- [RFC8294] Liu, X., Qu, Y., Lindem, A., Hopps, C., and L. Berger, "Common YANG Data Types for the Routing Area", RFC 8294, DOI 10.17487/RFC8294, December 2017, <<https://www.rfc-editor.org/info/rfc8294>>.
- [RFC8340] Bjorklund, M. and L. Berger, Ed., "YANG Tree Diagrams", BCP 215, RFC 8340, DOI 10.17487/RFC8340, March 2018, <<https://www.rfc-editor.org/info/rfc8340>>.
- [RFC8341] Bierman, A. and M. Bjorklund, "Network Configuration Access Control Model", STD 91, RFC 8341, DOI 10.17487/RFC8341, March 2018, <<https://www.rfc-editor.org/info/rfc8341>>.
- [RFC8342] Bjorklund, M., Schoenwaelder, J., Shafer, P., Watsen, K., and R. Wilton, "Network Management Datastore Architecture (NMDA)", RFC 8342, DOI 10.17487/RFC8342, March 2018, <<https://www.rfc-editor.org/info/rfc8342>>.
- [RFC8343] Bjorklund, M., "A YANG Data Model for Interface Management", RFC 8343, DOI 10.17487/RFC8343, March 2018, <<https://www.rfc-editor.org/info/rfc8343>>.
- [RFC8349] Lhotka, L., Lindem, A., and Y. Qu, "A YANG Data Model for Routing Management (NMDA Version)", RFC 8349, DOI 10.17487/RFC8349, March 2018, <<https://www.rfc-editor.org/info/rfc8349>>.

- [RFC8402] Filsfils, C., Ed., Previdi, S., Ed., Ginsberg, L., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing Architecture", RFC 8402, DOI 10.17487/RFC8402, July 2018, <<https://www.rfc-editor.org/info/rfc8402>>.
- [RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <<https://www.rfc-editor.org/info/rfc8446>>.

## 8.2. Informative References

- [RFC3031] Rosen, E., Viswanathan, A., and R. Callon, "Multiprotocol Label Switching Architecture", RFC 3031, DOI 10.17487/RFC3031, January 2001, <<https://www.rfc-editor.org/info/rfc3031>>.
- [RFC7424] Krishnan, R., Yong, L., Ghanwani, A., So, N., and B. Khasnabish, "Mechanisms for Optimizing Link Aggregation Group (LAG) and Equal-Cost Multipath (ECMP) Component Link Utilization in Networks", RFC 7424, DOI 10.17487/RFC7424, January 2015, <<https://www.rfc-editor.org/info/rfc7424>>.

## Authors' Addresses

Tarek Saad  
Juniper Networks  
  
Email: [tsaad@juniper.net](mailto:tsaad@juniper.net)

Kamran Raza  
Cisco Systems Inc  
  
Email: [skraza@cisco.com](mailto:skraza@cisco.com)

Rakesh Gandhi  
Cisco Systems Inc  
  
Email: [rgandhi@cisco.com](mailto:rgandhi@cisco.com)

Xufeng Liu  
Volta Networks  
  
Email: [xufeng.liu.ietf@gmail.com](mailto:xufeng.liu.ietf@gmail.com)

Vishnu Pavan Beeram  
Juniper Networks

Email: [vbeeram@juniper.net](mailto:vbeeram@juniper.net)

TEAS Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: 11 August 2022

T. Saad  
Juniper Networks  
R. Gandhi  
Cisco Systems Inc  
X. Liu  
Volta Networks  
V.P. Beeram  
Juniper Networks  
I. Bryskin  
Individual  
O. Gonzalez de Dios  
Telefonica  
7 February 2022

A YANG Data Model for Traffic Engineering Tunnels, Label Switched Paths  
and Interfaces  
draft-ietf-teas-yang-te-29

Abstract

This document defines a YANG data model for the provisioning and management of Traffic Engineering (TE) tunnels, Label Switched Paths (LSPs), and interfaces. The model is divided into YANG modules that classify data into generic, device-specific, technology agnostic, and technology-specific elements.

This model covers data for configuration, operational state, remote procedural calls, and event notifications.

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 <https://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 11 August 2022.



## Copyright Notice

Copyright (c) 2022 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 (<https://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 Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

## Table of Contents

1. Introduction . . . . .	3
2. Requirements Language . . . . .	3
2.1. Prefixes in Data Node Names . . . . .	4
2.2. Model Tree Diagrams . . . . .	4
3. Design Considerations . . . . .	5
3.1. State Data Organization . . . . .	5
4. Model Overview . . . . .	6
4.1. Module Relationship . . . . .	6
5. TE YANG Model . . . . .	7
5.1. Module Structure . . . . .	7
5.1.1. TE Globals . . . . .	9
5.1.2. TE Tunnels . . . . .	12
5.1.3. TE LSPs . . . . .	19
5.2. Tree Diagram . . . . .	19
5.3. YANG Module . . . . .	60
6. TE Device YANG Model . . . . .	98
6.1. Module Structure . . . . .	99
6.1.1. TE Interfaces . . . . .	99
6.2. Tree Diagram . . . . .	100
6.3. YANG Module . . . . .	102
7. Notifications . . . . .	116
8. TE Generic and Helper YANG Modules . . . . .	117
9. IANA Considerations . . . . .	117
10. Security Considerations . . . . .	117
11. Acknowledgement . . . . .	119
12. Contributors . . . . .	119
13. Appendix A: Data Tree Examples . . . . .	119
13.1. Basic Tunnel Setup . . . . .	120
13.2. Global Named Path Constraints . . . . .	121
13.3. Tunnel with Global Path Constraint . . . . .	121
13.4. Tunnel with Per-tunnel Path Constraint . . . . .	122
13.5. Tunnel State . . . . .	123

14. References . . . . .	124
14.1. Normative References . . . . .	124
14.2. Informative References . . . . .	127
Authors' Addresses . . . . .	128

## 1. Introduction

YANG [RFC6020] and [RFC7950] is a data modeling language that was introduced to define the contents of a conceptual data store that allows networked devices to be managed using NETCONF [RFC6241]. YANG has proved relevant beyond its initial confines, as bindings to other interfaces (e.g. RESTCONF [RFC8040]) and encoding other than XML (e.g. JSON) are being defined. Furthermore, YANG data models can be used as the basis of implementation for other interfaces, such as CLI and programmatic APIs.

This document describes YANG data model for Traffic Engineering (TE) tunnels, Label Switched Paths (LSPs), and interfaces. The model covers data applicable to generic or device-independent, device-specific, and Multiprotocol Label Switching (MPLS) technology specific.

The document describes a high-level relationship between the modules defined in this document, as well as other external protocol YANG modules. The TE generic YANG data model does not include any data specific to a signaling protocol. It is expected other data plane technology model(s) will augment the TE generic YANG data model.

Also, it is expected other YANG module(s) that model TE signaling protocols, such as RSVP-TE ([RFC3209], [RFC3473]), or Segment-Routing TE (SR-TE) [I-D.ietf-spring-segment-routing-policy] will augment the generic TE YANG module.

## 2. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

The following terms are defined in [RFC6241] and are used in this specification:

- \* client
- \* configuration data

\* state data

This document also makes use of the following terminology introduced in the YANG Data Modeling Language [RFC7950]:

\* augment

\* data model

\* data node

## 2.1. Prefixes in Data Node Names

In this document, names of data nodes and other data model objects are prefixed using the standard prefix associated with the corresponding YANG imported modules, as shown in Table 1.

Prefix	YANG module	Reference
yang	ietf-yang-types	[RFC6991]
inet	ietf-inet-types	[RFC6991]
rt-types	ietf-routing-types	[RFC8294]
te-types	ietf-te-types	[RFC8776]
te-packet-types	ietf-te-packet-types	[RFC8776]
te	ietf-te	this document
te-dev	ietf-te-device	this document

Table 1: Prefixes and corresponding YANG modules

## 2.2. Model Tree Diagrams

The tree diagrams extracted from the module(s) defined in this document are given in subsequent sections as per the syntax defined in [RFC8340].

### 3. Design Considerations

This document describes a generic TE YANG data model that is independent of any dataplane technology. One of the design objectives is to allow specific data plane technology models to reuse the TE generic data model and possibly augment it with technology specific data.

The elements of the generic TE YANG data model, including TE Tunnels, LSPs, and interfaces have leaf(s) that identify the technology layer where they reside. For example, the LSP encoding type can identify the technology associated with a TE Tunnel or LSP.

Also, the generic TE YANG data model does not cover signaling protocol data. The signaling protocol used to instantiate TE LSPs are outside the scope of this document and expected to be covered by augmentations defined in other document(s).

The following other design considerations are taken into account with respect data organization:

- \* The generic TE YANG data model 'ietf-te' contains device independent data and can be used to model data off a device (e.g. on a TE controller). The device-specific TE data is defined in module 'ietf-te-device' as shown in Figure 1,
- \* In general, minimal elements in the model are designated as "mandatory" to allow freedom to vendors to adapt the data model to their specific product implementation.
- \* Suitable defaults are specified for all configurable elements.
- \* The model declares a number of TE functions as features that can be optionally supported.

#### 3.1. State Data Organization

The Network Management Datastore Architecture (NMDA) [RFC8342] addresses modeling state data for ephemeral objects. This document adopts the NMDA model for configuration and state data representation as per IETF guidelines for new IETF YANG models.

#### 4. Model Overview

The data models defined in this document cover the core TE features that are commonly supported by different vendor implementations. The support of extended or vendor specific TE feature(s) is expected to be in either augmentations, or deviations to the model defined in this document.

##### 4.1. Module Relationship

The generic TE YANG data model that is defined in "ietf-te.yang" covers the building blocks that are device independent and agnostic of any specific technology or control plane instances. The TE device model defined in "ietf-te-device.yang" augments the generic TE YANG data model and covers data that is specific to a device - for example, attributes of TE interfaces, or TE timers that are local to a TE node.

The TE data model for specific instances of data plane technology exist in a separate YANG module(s) that augment the generic TE YANG data model. For example, the MPLS-TE module "ietf-te-mpls.yang" is defined in another document and augments the TE generic model as shown in Figure 1.

The TE data model for specific instances of signaling protocol are outside the scope of this document and are defined in other documents. For example, the RSVP-TE YANG model augmentation of the TE model is covered in [I-D.ietf-teas-yang-rsvp].

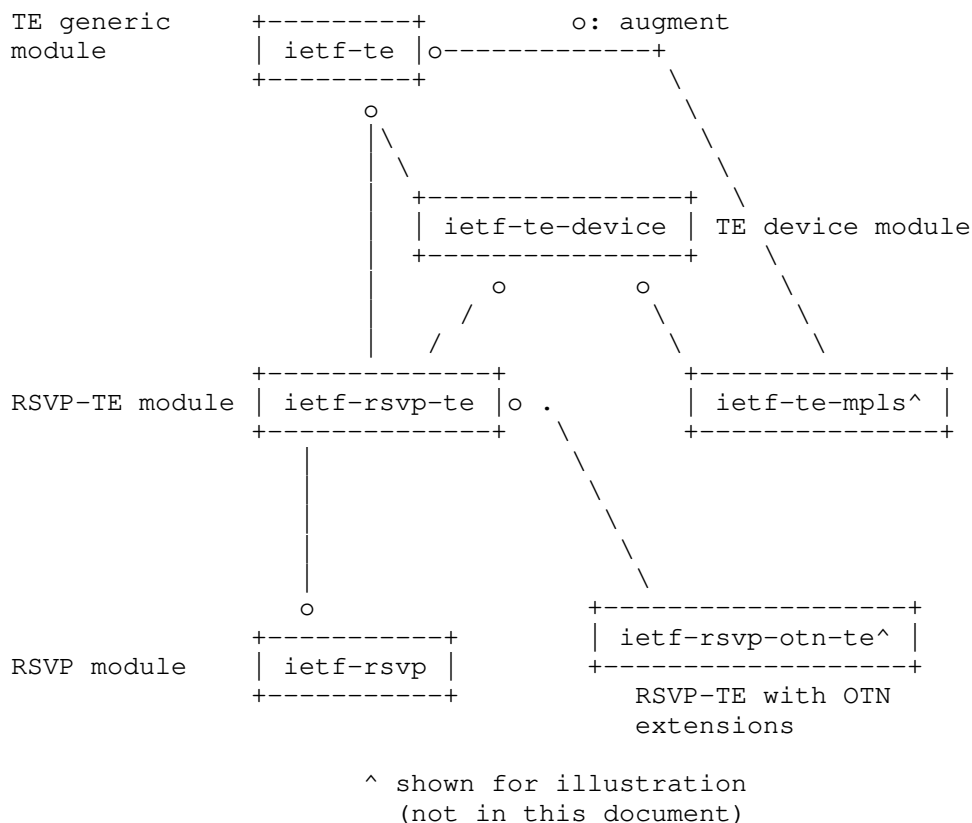


Figure 1: Relationship of TE module(s) with signaling protocol modules

## 5. TE YANG Model

The generic TE YANG module ('ietf-te') is meant to manage and operate a TE network. This includes creating, modifying and retrieving TE Tunnels, LSPs, and interfaces and their associated attributes (e.g. Administrative-Groups, SRLGs, etc.).

The detailed tree structure is provided in Figure 2.

### 5.1. Module Structure

The 'ietf-te' uses three main containers grouped under the main 'te' container (see Figure 2). The 'te' container is the top level container in the data model. The presence of the 'te' container enables TE function system wide. Below provides further descriptions of containers that exist under the 'te' top level container.

**globals:**

The 'globals' container maintains the set of global TE attributes that can be applicable to TE Tunnel(s) and interface(s).

**tunnels:**

The 'tunnels' container includes the list of TE Tunnels that are instantiated. Refer to Section 5.1.2 for further details on the properties of a TE Tunnel.

**lsps:**

The 'lsps' container includes the list of TE LSP(s) that are instantiated for TE Tunnels. Refer to Section 5.1.3 for further details on the properties of a TE LSP.

**tunnels-path-compute:**

A Remote Procedure Call (RPC) to request path computation for a specific TE Tunnel. The RPC allows requesting path computation using atomic and stateless operation. A tunnel may also be configured in 'compute-only' mode to provide stateful path updates - see Section 5.1.2 for further details.

**tunnels-action:**

An RPC to request a specific action (e.g. reoptimize, or tear-and-setup) to be taken on a specific tunnel or all tunnels.

```
module: ietf-te
  +--rw te!
    +--rw globals
      .
      .
    +--rw tunnels
      .
      .
    +-- lsps
```

```
rpcs:
  +---x tunnels-path-compute
  +---x tunnels-action
```

Figure 2: TE Tunnel model high-level YANG tree view

### 5.1.1. TE Globals

The 'globals' container covers properties that control TE features behavior system-wide, and its respective state (see Figure 3). The TE globals configuration include:

```

+--rw globals
|   +--rw named-admin-groups
|   |   +--rw named-admin-group* [name]
|   ..
|   +--rw named-srlgs
|   |   +--rw named-srlg* [name] {te-types:named-srlg-groups}?
|   ..
|   +--rw named-path-constraints
|   |   +--rw named-path-constraint* [name]
|   ..

```

Figure 3: TE globals YANG subtree high-level structure

#### named-admin-groups:

A YANG container for the list of named (extended) administrative groups that may be applied to TE links.

#### named-srlgs:

A YANG container for the list named Shared Risk Link Groups (SRLGs) that may be applied to TE links.

#### named-path-constraints:

A YANG container for a list of named path constraints. Each named path constraint is composed of a set of constraints that can be applied during path computation. A named path constraint can be applied to multiple TE Tunnels. Path constraints may also be specified directly under the TE Tunnel. The path constraint specified under the TE Tunnel take precedence over the path constraints derived from the referenced named path constraint. A named path constraint entry can be formed up of the following path constraints:



```

|   +--rw named-path-constraints
|       +--rw named-path-constraint* [name]
|           +--rw name                               string
|           +--rw te-bandwidth
| // ...
|       +--rw link-protection?                       identityref
|       +--rw setup-priority?                         uint8
|       +--rw hold-priority?                         uint8
|       +--rw signaling-type?                       identityref
|       +--rw path-metric-bounds
| // ...
|       +--rw path-affinities-values
| // ...
|       +--rw path-affinity-names
| // ...
|       +--rw path-srlgs-lists
| // ...
|       +--rw path-srlgs-names
| // ...
|       +--rw disjointness?
|           |   te-path-disjointness
| // ...
|       +--rw explicit-route-objects-always
| // ...
|       |   +--rw route-object-exclude-always* [index]
|       |   +--rw route-object-include-exclude* [index]

```

Figure 4: Named path constraints YANG subtree

- o te-bandwidth: A YANG container that holds the technology agnostic TE bandwidth constraint.
- o link-protection: A YANG leaf that holds the link protection type constraint required for the links to be included in the computed path.
- o setup/hold priority: A YANG leaf that holds the LSP setup and hold admission priority as defined in [RFC3209].
- o signaling-type: A YANG leaf that holds the LSP setup type, such as RSVP-TE or SR.
- o path-metric-bounds: A YANG container that holds the set of metric bounds applicable on the computed TE tunnel path.

- o `path-affinities-values`: A YANG container that holds the set of affinity values and mask to be used during path computation.
- o `path-affinity-names`: A YANG container that holds the set of named affinity constraints and corresponding inclusion or exclusions instruction for each to be used during path computation.
- o `path-srlgs-lists`: A YANG container that holds the set of SRLG values and corresponding inclusion or exclusions instruction to be used during path computation.
- o `path-srlgs-names`: A YANG container that holds the set of named SRLG constraints and corresponding inclusion or exclusions instruction for each to be used during path computation.
- o `disjointness`: The level of resource disjointness constraint that the secondary path of a TE tunnel has to adhere to.
- o `explicit-route-objects-always`: A YANG container that contains two route objects lists:
  - + `'route-object-exclude-always'`: a list of route entries to always exclude from the path computation.
  - + `'route-object-include-exclude'`: a list of route entries to include or exclude in the path computation.

The `'route-object-include-exclude'` is used to configure constraints on which route objects (e.g., nodes, links) are included or excluded in the path computation.

The interpretation of an empty `'route-object-include-exclude'` list depends on the TE Tunnel (end-to-end or Tunnel Segment) and on the specific path, according to the following rules:

1. An empty `'route-object-include-exclude'` list for the primary path of an end-to-end TE Tunnel indicates that there are no route objects to be included or excluded in the path computation.
2. An empty `'route-object-include-exclude'` list for the primary path of a TE Tunnel Segment indicates that no primary LSP is required for that TE Tunnel.

3. An empty 'route-object-include-exclude' list for a reverse path means it always follows the forward path (i.e., the TE Tunnel is co-routed). When the 'route-object-include-exclude' list is not empty, the reverse path is routed independently of the forward path.
4. An empty 'route-object-include-exclude' list for the secondary (forward) path indicates that the secondary path has the same endpoints as the primary path.

#### 5.1.2. TE Tunnels

The 'tunnels' container holds the list of TE Tunnels that are provisioned on devices in the network (see Figure 5).

A TE Tunnel in the list is uniquely identified by a name. When the model is used to manage a specific device, the 'tunnels' list contains the TE Tunnels originating from the specific device. When the model is used to manage a TE controller, the 'tunnels' list contains all TE Tunnels and TE tunnel segments originating from device(s) that the TE controller manages.

The TE Tunnel model allows the configuration and management of the following TE tunnel related objects:

##### TE Tunnel:

A YANG container of one or more LSPs established between the source and destination TE Tunnel termination points. A TE Tunnel LSP is a connection-oriented service provided by the network layer for the delivery of client data between a source and the destination of the TE Tunnel termination points.

##### TE Tunnel Segment:

A part of a multi-domain TE Tunnel that is within a specific network domain.

```

+--rw tunnels
|   +--rw tunnel* [name]
|   |   +--rw name                               string
|   |   +--rw alias?                             string
|   |   +--rw identifier?                         uint32
|   |   +--rw color?                             uint32
|   |   +--rw description?                       string
|   |   +--ro operational-state?                 identityref
|   |   +--rw encoding?                         identityref
|   |   +--rw switching-type?                   identityref
|   |   +--rw admin-state?                     identityref
|   |   +--rw reoptimize-timer?                 uint16
|   |   +--rw source?                           te-types:te-node-id
|   |   +--rw destination?                     te-types:te-node-id
|   |   +--rw src-tunnel-tp-id?                 binary
|   |   +--rw dst-tunnel-tp-id?                 binary
|   |   +--rw controller
|   |   |   +--rw protocol-origin?               identityref
|   |   |   +--rw controller-entity-id?         string
|   |   +--rw bidirectional?                   boolean
|   |   +--rw association-objects
|   |   |   +--rw association-object* [association-key]
|   |
|   |   // ..
|   |   |
|   |   +--rw protection
|   |
|   |   // ..
|   |   |
|   |   +--rw restoration
|   |
|   |   // ..
|   |   |
|   |   +--rw te-topology-identifier
|   |
|   |   // ..
|   |   |
|   |   +--rw hierarchy
|   |
|   |   // ..

```

Figure 5: TE Tunnel list YANG subtree structure

The TE Tunnel has a number of attributes that are set directly under the tunnel (see Figure 5). The main attributes of a TE Tunnel are described below:

**operational-state:**

A YANG leaf that holds the operational state of the tunnel.

**name:**

A YANG leaf that holds the name of a TE Tunnel. The name of the TE Tunnel uniquely identifies the tunnel within the TE tunnel list. The name of the TE Tunnel can be formatted as a Uniform

Resource Indicator (URI) by including the namespace to ensure uniqueness of the name amongst all the TE Tunnels present on devices and controllers.

alias:

A YANG leaf that holds an alternate name to the TE tunnel. Unlike the TE tunnel name, the alias can be modified at any time during the lifetime of the TE tunnel.

identifier:

A YANG leaf that holds an identifier of the tunnel. This identifier is unique amongst tunnels originated from the same ingress device.

color:

A YANG leaf that holds the color associated with the TE tunnel. The color is used to map or steer services that carry matching color on to the TE tunnel as described in [RFC9012].

encoding/switching:

The 'encoding' and 'switching-type' are YANG leafs that define the specific technology in which the tunnel operates in as described in [RFC3945].

reoptimize-timer:

A YANG leaf to set the interval period for tunnel reoptimization.

source/destination:

YANG leafs that define the tunnel source and destination node endpoints.

src-tunnel-tp-id/dst-tunnel-tp-id:

YANG leafs that hold the identifiers of source and destination TE Tunnel Termination Points (TTPs) [RFC8795] residing on the source and destination nodes. The TTP identifiers are optional on nodes that have a single TTP per node. For example, TTP identifiers are optional for packet (IP/MPLS) routers.

controller:

A YANG container that holds tunnel data relevant to an optional external TE controller that may initiate or control a tunnel. This target node may be augmented by external module(s), for example, to add data for PCEP initiated and/or delegated tunnels.

bidirectional:

A YANG leaf that when present indicates the LSPs of a TE Tunnel are bidirectional and co-routed.

association-objects:

A YANG container that holds the set of associations of the TE Tunnel to other TE Tunnels. Associations at the TE Tunnel level apply to all paths of the TE Tunnel. The TE tunnel associations can be overridden by associations configured directly under the TE Tunnel path.

protection:

A YANG container that holds the TE Tunnel protection properties.

restoration:

A YANG container that holds the TE Tunnel restoration properties.

te-topology-identifier:

A YANG container that holds the topology identifier associated with the topology where paths for the TE tunnel are computed.

```

+--rw hierarchy
|   +--rw dependency-tunnels
|   |   +--rw dependency-tunnel* [name]
|   |   |   +--rw name
|   |   |   |   -> ../../../../tunnels/tunnel/name
|   |   |   +--rw encoding?          identityref
|   |   |   +--rw switching-type?    identityref
|   |   +--rw hierarchical-link
|   |   |   +--rw local-te-node-id?    te-types:te-node-id
|   |   |   +--rw local-te-link-tp-id? te-types:te-tp-id
|   |   |   +--rw remote-te-node-id?   te-types:te-node-id
|   |   +--rw te-topology-identifier
|   |   |   +--rw provider-id?    te-global-id
|   |   |   +--rw client-id?      te-global-id
|   |   |   +--rw topology-id?    te-topology-id

```

Figure 6: TE Tunnel hierarchy YANG subtree

#### hierarchy:

A YANG container that holds hierarchy related properties of the TE Tunnel (see Figure 6. A TE LSP can be set up in MPLS or Generalized MPLS (GMPLS) networks to be used as a TE links to carry traffic in other (client) networks [RFC6107]. In this case, the model introduces the TE Tunnel hierarchical link endpoint parameters to identify the specific link in the client layer that the underlying TE Tunnel is associated with. The hierarchy container includes the following:

- o dependency-tunnels: A set of hierarchical TE Tunnels provisioned or to be provisioned in the immediate lower layer that this TE tunnel depends on for multi-layer path computation. A dependency TE Tunnel is provisioned if and only if it is used (selected by path computation) at least by one client layer TE Tunnel. The TE link in the client layer network topology supported by a dependent TE Tunnel is dynamically created only when the dependency TE Tunnel is actually provisioned.
- o hierarchical-link: A YANG container that holds the identity of the hierarchical link (in the client layer) that is supported by this TE Tunnel. The endpoints of the hierarchical link are defined by TE tunnel source and destination node endpoints. The hierarchical link can be identified by its source and destination link termination point identifiers.

#### 5.1.2.1. TE Tunnel Paths

The TE Tunnel can be configured with a set of paths that define the tunnel forward and reverse paths as described in Figure 7. Moreover, a primary path can be specified a set of candidate secondary paths that can be visited to support path protection. The following describe further the list of paths associated with a TE Tunnel.

```

|      +--rw primary-paths
|      |   +--rw primary-path* [name]
|      |   |   +--rw name
|      |   |   |   string
|  // ..
|      |   +
|      |   +--rw primary-reverse-path
|      |   |   +--rw name?
|      |   |   |   string
|  // ..
|      |   |
|      |   |   +--rw candidate-secondary-reverse-paths
|      |   |   |   +--rw candidate-secondary-reverse-path*
|      |   |   |   |   [secondary-path]
|      |   |   |   |   +--rw secondary-path
|      |   |   |   |   |   leafref
|      |   |   +--rw candidate-secondary-paths
|      |   |   |   +--rw candidate-secondary-path* [secondary-path]
|      |   |   |   |   +--rw secondary-path
|      |   |   |   |   |   leafref
|      |   |   |   +--ro active?
|      |   |   |   |   boolean
|
|      +--rw secondary-paths
|      |   +--rw secondary-path* [name]
|      |   |   +--rw name
|      |   |   |   string
|  // ..
|      |   +--rw secondary-reverse-paths
|      |   |   +--rw secondary-reverse-path* [name]
|      |   |   |   +--rw name
|      |   |   |   |   string

```

Figure 7: TE Tunnel paths YANG tree structure

**primary-paths:**

A YANG container that holds the list of primary paths. A primary path is identified by 'name'. A primary path is selected from the list to instantiate a primary forwarding LSP for the tunnel. The list of primary paths is visited by order of preference. A primary path has the following attributes:

- primary-reverse-path: A YANG container that holds properties of the primary reverse path. The reverse path is applicable to bidirectional TE Tunnels.
- candidate-secondary-paths: A YANG container that holds a list of candidate secondary paths which may be used for the primary path to support path protection. The candidate secondary path(s) reference path(s) from the tunnel secondary paths list. The preference of the secondary paths is specified within the list and dictates the order of visiting the secondary path from the list. The attributes of a secondary path can be defined



separately from the primary path. The attributes of a secondary path will be inherited from the associated 'active' primary when not explicitly defined for the secondary path.

#### secondary-paths:

A YANG container that holds the set of secondary paths. A secondary path is identified by 'name'. A secondary path can be referenced from the TE Tunnel's 'candidate-secondary-path' list. A secondary path contains attributes similar to a primary path.

#### secondary-reverse-paths:

A YANG container that holds the set of secondary reverse paths. A secondary reverse path is identified by 'name'. A secondary reverse path can be referenced from the TE Tunnel's 'candidate-secondary-reverse-paths' list. A secondary reverse path contains attributes similar to a primary path.

The following set common path attributes are shared for primary forward and reverse primary and secondary paths:

#### compute-only:

A path of TE Tunnel is, by default, provisioned so that it can be instantiated in forwarding to carry traffic as soon as a valid path is computed. In some cases, a TE path may be provisioned for the only purpose of computing a path and reporting it without the need to instantiate the LSP or commit any resources. In such a case, the path is configured in 'compute-only' mode to distinguish it from the default behavior. A 'compute-only' path is configured as a usual with the associated per path constraint(s) and properties on a device or TE controller. The device or TE controller computes the feasible path(s) subject to configured constraints. A client may query the 'compute-only' computed path properties 'on-demand', or alternatively, can subscribe to be notified of computed path(s) and whenever the path properties change.

#### use-path-computation:

A YANG leaf that indicates whether or not path computation is to be used for a specified path.

#### lockdown:

A YANG leaf that when set indicates the existing path should not be reoptimized after a failure on any of its traversed links.

te-topology-identifier:

A YANG container that holds the topology identifier associated with the tunnel.

optimizations:

a YANG container that holds the optimization objectives that path computation will use to select a path.

computed-paths-properties: > A YANG container that holds properties for the list of computed paths.

computed-path-error-infos:

A YANG container that holds a list of errors related to the path.

lsps:

a YANG container that holds a list of LSPs that are instantiated for this specific path.

#### 5.1.3. TE LSPs

The 'lsps' container includes the set of TE LSP(s) that are instantiated. A TE LSP is identified by a 3-tuple ('tunnel-name', 'node', 'lsp-id').

When the model is used to manage a specific device, the 'lsps' list contains all TE LSP(s) that traverse the device (including ingressing, transiting and egressing the device).

When the model is used to manage a TE controller, the 'lsps' list contains all TE LSP(s) that traverse all network devices (including ingressing, transiting and egressing the device) that the TE controller manages.

#### 5.2. Tree Diagram

Figure 8 shows the tree diagram of the generic TE YANG model defined in modules 'ietf-te.yang'.

```

module: ietf-te
+--rw te!
  +--rw globals
    +--rw named-admin-groups
      +--rw named-admin-group* [name]
        {te-types:extended-admin-groups,te-types:named-extend
ed-admin-groups}?
        +--rw name string
        +--rw bit-position? uint32
    +--rw named-srlgs
      +--rw named-srlg* [name] {te-types:named-srlg-groups}?
        +--rw name string
        +--rw value? te-types:srlg
        +--rw cost? uint32
    +--rw named-path-constraints
      +--rw named-path-constraint* [name]
        {te-types:named-path-constraints}?
        +--rw name string
        +--rw te-bandwidth
          +--rw (technology)?
            +--:(generic)
              +--rw generic? te-bandwidth
        +--rw link-protection? identityref
        +--rw setup-priority? uint8
        +--rw hold-priority? uint8
        +--rw signaling-type? identityref
        +--rw path-metric-bounds
          +--rw path-metric-bound* [metric-type]
            +--rw metric-type identityref
            +--rw upper-bound? uint64
        +--rw path-affinities-values
          +--rw path-affinities-value* [usage]
            +--rw usage identityref
            +--rw value? admin-groups
        +--rw path-affinity-names
          +--rw path-affinity-name* [usage]
            +--rw usage identityref
            +--rw affinity-name* [name]
              +--rw name string
        +--rw path-srlgs-lists
          +--rw path-srlgs-list* [usage]
            +--rw usage identityref
            +--rw values* srlg
        +--rw path-srlgs-names
          +--rw path-srlgs-name* [usage]
            +--rw usage identityref
            +--rw names* string
        +--rw disjointness?

```

```

    te-path-disjointness
+--rw explicit-route-objects-always
+--rw route-object-exclude-always* [index]
+--rw index                               uint32
+--rw (type)?
+--:(numbered-node-hop)
+--rw numbered-node-hop
+--rw node-id         te-node-id
+--rw hop-type?       te-hop-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
+--:(as-number)
+--rw as-number-hop
+--rw as-number        inet:as-number
+--rw hop-type?       te-hop-type
+--:(label)
+--rw label-hop
+--rw te-label
+--rw (technology)?
+--:(generic)
+--rw generic?
+--rw direction?      rt-types:generalized-label
+--rw direction?      te-label-direction
+--rw route-object-include-exclude* [index]
+--rw explicit-route-usage?           identityref
+--rw index                           uint32
+--rw (type)?
+--:(numbered-node-hop)
+--rw numbered-node-hop
+--rw node-id         te-node-id
+--rw hop-type?       te-hop-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
+---:(as-number)
    +---rw as-number-hop
    +---rw as-number       inet:as-number
    +---rw hop-type?       te-hop-type
+---:(label)
    +---rw label-hop
    +---rw te-label
    +---rw (technology)?
    |   +---:(generic)
    |       +---rw generic?
    |           rt-types:generalized-label
    +---rw direction?
    |       te-label-direction
+---:(srlg)
    +---rw srlg
    +---rw srlg?          uint32
+---rw path-in-segment!
    +---rw label-restrictions
    +---rw label-restriction* [index]
    +---rw restriction?     enumeration
    +---rw index            uint32
    +---rw label-start
    |   +---rw te-label
    |       +---rw (technology)?
    |           +---:(generic)
    |               +---rw generic?
    |                   rt-types:generalized-label
    +---rw direction?
    |       te-label-direction
+---rw label-end
    +---rw te-label
    +---rw (technology)?
    |   +---:(generic)
    |       +---rw generic?
    |           rt-types:generalized-label
    +---rw direction?
    |       te-label-direction
+---rw label-step
    |   +---rw (technology)?
    |       +---:(generic)
    |           +---rw generic?      int32
    +---rw range-bitmap?    yang:hex-string
+---rw path-out-segment!
    +---rw label-restrictions

```

```

        +---rw label-restriction* [index]
            +---rw restriction?    enumeration
            +---rw index           uint32
            +---rw label-start
                +---rw te-label
                    +---rw (technology)?
                        +---:(generic)
                            +---rw generic?
                                rt-types:generalized-label
                    +---rw direction?
                        te-label-direction
            +---rw label-end
                +---rw te-label
                    +---rw (technology)?
                        +---:(generic)
                            +---rw generic?
                                rt-types:generalized-label
                    +---rw direction?
                        te-label-direction
            +---rw label-step
                +---rw (technology)?
                    +---:(generic)
                        +---rw generic?    int32
            +---rw range-bitmap?    yang:hex-string
+---rw tunnels
    +---rw tunnel* [name]
        +---rw name                string
        +---rw alias?              string
        +---rw identifier?         uint32
        +---rw color?              uint32
        +---rw description?        string
        +---rw admin-state?        identityref
        +---ro operational-state?   identityref
        +---rw encoding?           identityref
        +---rw switching-type?     identityref
        +---rw source?             te-types:te-node-id
        +---rw destination?        te-types:te-node-id
        +---rw src-tunnel-tp-id?   binary
        +---rw dst-tunnel-tp-id?   binary
        +---rw bidirectional?      boolean
        +---rw controller
            +---rw protocol-origin? identityref
            +---rw controller-entity-id? string
        +---rw reoptimize-timer?   uint16
        +---rw association-objects
            +---rw association-object* [association-key]
                +---rw association-key    string
                +---rw type?              identityref

```

```

+--rw id?                               uint16
+--rw source
  +--rw id?      te-gen-node-id
  +--rw type?    enumeration
+--rw association-object-extended* [association-key]
  +--rw association-key    string
  +--rw type?              identityref
  +--rw id?                uint16
  +--rw source
    +--rw id?      te-gen-node-id
    +--rw type?    enumeration
  +--rw global-source?    uint32
  +--rw extended-id?      yang:hex-string
+--rw protection
  +--rw enable?                                boolean
  +--rw protection-type?                        identityref
  +--rw protection-reversion-disable?          boolean
  +--rw hold-off-time?                          uint32
  +--rw wait-to-revert?                         uint16
  +--rw aps-signal-id?                          uint8
+--rw restoration
  +--rw enable?                                boolean
  +--rw restoration-type?                      identityref
  +--rw restoration-scheme?                    identityref
  +--rw restoration-reversion-disable?          boolean
  +--rw hold-off-time?                          uint32
  +--rw wait-to-restore?                        uint16
  +--rw wait-to-revert?                        uint16
+--rw te-topology-identifier
  +--rw provider-id?    te-global-id
  +--rw client-id?      te-global-id
  +--rw topology-id?    te-topology-id
+--rw te-bandwidth
  +--rw (technology)?
    +--:(generic)
      +--rw generic?    te-bandwidth
+--rw link-protection?                identityref
+--rw setup-priority?                  uint8
+--rw hold-priority?                   uint8
+--rw signaling-type?                  identityref
+--rw hierarchy
  +--rw dependency-tunnels
    +--rw dependency-tunnel* [name]
      +--rw name
        -> /te/tunnels/tunnel/name
      +--rw encoding?              identityref
      +--rw switching-type?        identityref
+--rw hierarchical-link

```

```

+--rw local-te-node-id?          te-types:te-node-id
+--rw local-te-link-tp-id?       te-types:te-tp-id
+--rw remote-te-node-id?         te-types:te-node-id
+--rw te-topology-identifier
  +--rw provider-id?             te-global-id
  +--rw client-id?               te-global-id
  +--rw topology-id?             te-topology-id
+--rw primary-paths
  +--rw primary-path* [name]
    +--rw name                    string
    +--rw path-computation-method? identityref
    +--rw path-computation-server
      +--rw id?                   te-gen-node-id
      +--rw type?                 enumeration
    +--rw compute-only?           empty
    +--rw use-path-computation?   boolean
    +--rw lockdown?               empty
    +--rw path-scope?             identityref
    +--rw preference?             uint8
    +--rw k-requested-paths?      uint8
    +--rw association-objects
      +--rw association-object* [association-key]
        +--rw association-key     string
        +--rw type?               identityref
        +--rw id?                 uint16
        +--rw source
          +--rw id?               te-gen-node-id
          +--rw type?             enumeration
      +--rw association-object-extended*
        [association-key]
        +--rw association-key     string
        +--rw type?               identityref
        +--rw id?                 uint16
        +--rw source
          +--rw id?               te-gen-node-id
          +--rw type?             enumeration
        +--rw global-source?      uint32
        +--rw extended-id?        yang:hex-string
+--rw optimizations
  +--rw (algorithm)?
    +--:(metric) {path-optimization-metric}?
      +--rw optimization-metric* [metric-type]
        +--rw metric-type
          identityref
        +--rw weight?
          uint8
        +--rw explicit-route-exclude-objects
        +--rw route-object-exclude-object*

```



```

[index]
+--rw index
|
|   uint32
+--rw (type)?
|
|   +--:(numbered-node-hop)
|   |
|   |   +--rw numbered-node-hop
|   |   |
|   |   |   +--rw node-id
|   |   |   |
|   |   |   |   te-node-id
|   |   |   |
|   |   |   +--rw hop-type?
|   |   |   |
|   |   |   |   te-hop-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
|   |   +--:(as-number)
|   |   |
|   |   |   +--rw as-number-hop
|   |   |   |
|   |   |   |   +--rw as-number
|   |   |   |   |
|   |   |   |   |   inet:as-number
|   |   |   |   |
|   |   |   |   +--rw hop-type?
|   |   |   |   |
|   |   |   |   |   te-hop-type
|   |   +--:(label)
|   |   |
|   |   |   +--rw label-hop
|   |   |   |
|   |   |   |   +--rw te-label
|   |   |   |   |
|   |   |   |   |   +--rw (technology)?
|   |   |   |   |   |
|   |   |   |   |   |   +--:(generic)
|   |   |   |   |   |   |
|   |   |   |   |   |   |   +--rw generic?
|   |   |   |   |   |   |
|   |   |   |   |   |   |   rt-types:ge
|   |   |   |   |   |   |
|   |   |   |   |   |   +--rw direction?
|   |   |   |   |   |   |
|   |   |   |   |   |   |   te-label-directio
|   |   +--:(srlg)
|   |   |
|   |   |   +--rw srlg
|   |   |   |
|   |   |   |   +--rw srlg?   uint32

```

```

+---rw explicit-route-include-objects
+---rw route-object-include-object*
    [index]
+---rw index
    |   uint32
+---rw (type)?
+---:(numbered-node-hop)
    +---rw numbered-node-hop
        +---rw node-id
            |   te-node-id
        +---rw hop-type?
            |   te-hop-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
+---:(as-number)
    +---rw as-number-hop
        +---rw as-number
            |   inet:as-number
        +---rw hop-type?
            |   te-hop-type
+---:(label)
    +---rw label-hop
        +---rw te-label
            +---rw (technology)?
                +---:(generic)
                    +---rw generic?
                        |   rt-types:ge
neralized-label
n
+---rw tiebreakers

```

```

        +---rw tiebreaker* [tiebreaker-type]
        +---rw tiebreaker-type identityref
    +---:(objective-function)
        {path-optimization-objective-function}?
        +---rw objective-function
        +---rw objective-function-type?
            identityref
+---rw named-path-constraint? leafref
    {te-types:named-path-constraints}?
+---rw te-bandwidth
    +---rw (technology)?
    +---:(generic)
        +---rw generic? te-bandwidth
+---rw link-protection? identityref
+---rw setup-priority? uint8
+---rw hold-priority? uint8
+---rw signaling-type? identityref
+---rw path-metric-bounds
    +---rw path-metric-bound* [metric-type]
    +---rw metric-type identityref
    +---rw upper-bound? uint64
+---rw path-affinities-values
    +---rw path-affinities-value* [usage]
    +---rw usage identityref
    +---rw value? admin-groups
+---rw path-affinity-names
    +---rw path-affinity-name* [usage]
    +---rw usage identityref
    +---rw affinity-name* [name]
    +---rw name string
+---rw path-srlgs-lists
    +---rw path-srlgs-list* [usage]
    +---rw usage identityref
    +---rw values* srlg
+---rw path-srlgs-names
    +---rw path-srlgs-name* [usage]
    +---rw usage identityref
    +---rw names* string
+---rw disjointness?
    te-path-disjointness
+---rw explicit-route-objects-always
    +---rw route-object-exclude-always* [index]
    +---rw index uint32
    +---rw (type)?
    +---:(numbered-node-hop)
        +---rw numbered-node-hop
        +---rw node-id te-node-id
        +---rw hop-type? te-hop-type

```

bel

```

+---:(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
+---:(as-number)
|   +---rw as-number-hop
|       +---rw as-number      inet:as-number
|       +---rw hop-type?      te-hop-type
+---:(label)
|   +---rw label-hop
|       +---rw te-label
|           +---rw (technology)?
|               +---:(generic)
|                   +---rw generic?
|                       rt-types:generalized-la
|
|       +---rw direction?
|           te-label-direction
+---rw route-object-include-exclude* [index]
+---rw explicit-route-usage?      identityref
+---rw index                      uint32
+---rw (type)?
+---:(numbered-node-hop)
|   +---rw numbered-node-hop
|       +---rw node-id      te-node-id
|       +---rw hop-type?    te-hop-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
+---:(as-number)
|   +---rw as-number-hop
|       +---rw as-number      inet:as-number
|       +---rw hop-type?      te-hop-type

```

bel

```

+---:(label)
|   +---rw label-hop
|       +---rw te-label
|           +---rw (technology)?
|               +---:(generic)
|                   +---rw generic?
|                       rt-types:generalized-la
|
|       +---rw direction?
|           te-label-direction
+---:(srlg)
|   +---rw srlg
|       +---rw srlg?    uint32
+---rw path-in-segment!
+---rw label-restrictions
+---rw label-restriction* [index]
+---rw restriction?      enumeration
+---rw index              uint32
+---rw label-start
|   +---rw te-label
|       +---rw (technology)?
|           +---:(generic)
|               +---rw generic?
|                   rt-types:generalized-label
|       +---rw direction?
|           te-label-direction
+---rw label-end
|   +---rw te-label
|       +---rw (technology)?
|           +---:(generic)
|               +---rw generic?
|                   rt-types:generalized-label
|       +---rw direction?
|           te-label-direction
+---rw label-step
|   +---rw (technology)?
|       +---:(generic)
|           +---rw generic?    int32
+---rw range-bitmap?    yang:hex-string
+---rw path-out-segment!
+---rw label-restrictions
+---rw label-restriction* [index]
+---rw restriction?      enumeration
+---rw index              uint32
+---rw label-start
|   +---rw te-label
|       +---rw (technology)?
|           +---:(generic)

```

```

|         |         +---rw generic?
|         |             rt-types:generalized-label
|         +---rw direction?
|             te-label-direction
+---rw label-end
|     +---rw te-label
|         +---rw (technology)?
|             +---:(generic)
|                 +---rw generic?
|                     rt-types:generalized-label
|         +---rw direction?
|             te-label-direction
+---rw label-step
|     +---rw (technology)?
|         +---:(generic)
|             +---rw generic?    int32
+---rw range-bitmap?    yang:hex-string
+---ro computed-paths-properties
+---ro computed-path-properties* [k-index]
+---ro k-index            uint8
+---ro path-properties
+---ro path-metric* [metric-type]
|     +---ro metric-type        identityref
|     +---ro accumulative-value? uint64
+---ro path-affinities-values
|     +---ro path-affinities-value* [usage]
|         +---ro usage          identityref
|         +---ro value?         admin-groups
+---ro path-affinity-names
|     +---ro path-affinity-name* [usage]
|         +---ro usage          identityref
|         +---ro affinity-name* [name]
|             +---ro name       string
+---ro path-srlgs-lists
|     +---ro path-srlgs-list* [usage]
|         +---ro usage          identityref
|         +---ro values*      srlg
+---ro path-srlgs-names
|     +---ro path-srlgs-name* [usage]
|         +---ro usage          identityref
|         +---ro names*       string
+---ro path-route-objects
|     +---ro path-route-object* [index]
|         +---ro index
|             |           uint32
|         +---ro (type)?
|             +---:(numbered-node-hop)
|                 +---ro numbered-node-hop

```

```

+--ro node-id      te-node-id
+--ro hop-type?
    te-hop-type
+--:(numbered-link-hop)
+--ro numbered-link-hop
+--ro link-tp-id    te-tp-id
+--ro hop-type?
    |
    te-hop-type
+--ro direction?
    te-link-direction
+--:(unnumbered-link-hop)
+--ro unnumbered-link-hop
+--ro link-tp-id    te-tp-id
+--ro node-id
    |
    te-node-id
+--ro hop-type?
    |
    te-hop-type
+--ro direction?
    te-link-direction
+--:(as-number)
+--ro as-number-hop
+--ro as-number
    |
    inet:as-number
+--ro hop-type?
    te-hop-type
+--:(label)
+--ro label-hop
+--ro te-label
    +--ro (technology)?
        |
        +--:(generic)
        |
        +--ro generic?
            rt-types:gener
+--ro direction?
    te-label-direction
+--ro te-bandwidth
+--ro (technology)?
    +--:(generic)
    +--ro generic?
        te-bandwidth
+--ro disjointness-type?
    te-types:te-path-disjointness
+--ro computed-path-error-infos
+--ro computed-path-error-info* []
+--ro error-description?
    string
+--ro error-timestamp?
    yang:date-and-time
+--ro error-reason?
    identityref
+--ro lsp-provisioning-error-infos
+--ro lsp-provisioning-error-info* []

```

```

+--ro error-description?  string
+--ro error-timestamp?    yang:date-and-time
+--ro error-node-id?      te-types:te-node-id
+--ro error-link-id?      te-types:te-tp-id
+--ro lsp-id?             uint16
+--ro lsps
|
|   +--ro lsp* [node lsp-id]
|   |   +--ro tunnel-name?
|   |   |   -> /te/lsps/lsp/tunnel-name
|   |   +--ro node          -> /te/lsps/lsp/node
|   |   +--ro lsp-id        -> /te/lsps/lsp/lsp-id
|
+--rw primary-reverse-path
|
|   +--rw name?                                string
|   +--rw path-computation-method?
|   |   identityref
|   +--rw path-computation-server
|   |   +--rw id?      te-gen-node-id
|   |   +--rw type?    enumeration
|   +--rw compute-only?                                empty
|   +--rw use-path-computation?
|   |   boolean
|   +--rw lockdown?                                empty
|   +--ro path-scope?
|   |   identityref
|   +--rw association-objects
|   |   +--rw association-object* [association-key]
|   |   |   +--rw association-key    string
|   |   |   +--rw type?              identityref
|   |   |   +--rw id?                uint16
|   |   |   +--rw source
|   |   |   |   +--rw id?      te-gen-node-id
|   |   |   |   +--rw type?    enumeration
|   |   +--rw association-object-extended*
|   |   |   [association-key]
|   |   |   +--rw association-key    string
|   |   |   +--rw type?              identityref
|   |   |   +--rw id?                uint16
|   |   |   +--rw source
|   |   |   |   +--rw id?      te-gen-node-id
|   |   |   |   +--rw type?    enumeration
|   |   +--rw global-source?        uint32
|   |   +--rw extended-id?          yang:hex-string
|   +--rw optimizations
|   |   +--rw (algorithm)?
|   |   |   +--:(metric) {path-optimization-metric}?
|   |   |   |   +--rw optimization-metric* [metric-type]
|   |   |   |   |   +--rw metric-type
|   |   |   |   |   |   identityref

```



```

+---rw weight?
|   uint8
+---rw explicit-route-exclude-objects
|   +---rw route-object-exclude-object*
|       [index]
|       +---rw index
|           |   uint32
|       +---rw (type)?
|           +---:(numbered-node-hop)
|               +---rw numbered-node-hop
|                   +---rw node-id
|                       |   te-node-id
|                   +---rw hop-type?
|                       |   te-hop-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
|           +---:(as-number)
|               +---rw as-number-hop
|                   +---rw as-number
|                       |   inet:as-number
|                   +---rw hop-type?
|                       |   te-hop-type
|           +---:(label)
|               +---rw label-hop
|                   +---rw te-label
|                       +---rw (technology)?
|                           +---:(generic)
|                               +---rw generic?
|                                   rt-types
:generalized-label
+---rw direction?
    te-label-direc

```

tion

```

+--:(srlg)
+--rw srlg
+--rw srlg?   uint32
+--rw explicit-route-include-objects
+--rw route-object-include-object*
    [index]
+--rw index
    |
    |   uint32
+--rw (type)?
+--:(numbered-node-hop)
    |
    |   +--rw numbered-node-hop
    |       |
    |       |   +--rw node-id
    |       |       |
    |       |       |   te-node-id
    |       |   +--rw hop-type?
    |       |       |
    |       |       |   te-hop-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
+--:(as-number)
    |
    |   +--rw as-number-hop
    |       |
    |       |   +--rw as-number
    |       |       |
    |       |       |   inet:as-number
    |       |   +--rw hop-type?
    |       |       |
    |       |       |   te-hop-type
+--:(label)
    |
    |   +--rw label-hop
    |       |
    |       |   +--rw te-label
    |       |       |
    |       |       |   +--rw (technology)?
    |       |       |       |
    |       |       |       |   +--:(generic)
    |       |       |       |       |
    |       |       |       |       |   +--rw generic?
    |       |       |       |       |       |
    |       |       |       |       |       |   rt-types
:generalized-label

```

```

tion
}

    +--rw direction?
        te-label-direct

    +--rw tiebreakers
        +--rw tiebreaker* [tiebreaker-type]
            +--rw tiebreaker-type
                identityref
    +--:(objective-function)
        {path-optimization-objective-function

    +--rw objective-function
        +--rw objective-function-type?
            identityref
    +--rw named-path-constraint?
        {te-types:named-path-constraints}?
    +--rw te-bandwidth
        +--rw (technology)?
            +--:(generic)
                +--rw generic?
                    te-bandwidth
    +--rw link-protection?
        identityref
    +--rw setup-priority?
        uint8
    +--rw hold-priority?
        uint8
    +--rw signaling-type?
        identityref
    +--rw path-metric-bounds
        +--rw path-metric-bound* [metric-type]
            +--rw metric-type
                identityref
            +--rw upper-bound?
                uint64
    +--rw path-affinities-values
        +--rw path-affinities-value* [usage]
            +--rw usage
                identityref
            +--rw value?
                admin-groups
    +--rw path-affinity-names
        +--rw path-affinity-name* [usage]
            +--rw usage
                identityref
            +--rw affinity-name* [name]
                +--rw name
                    string
    +--rw path-srlgs-lists
        +--rw path-srlgs-list* [usage]
            +--rw usage
                identityref
            +--rw values*
                srlg
    +--rw path-srlgs-names
        +--rw path-srlgs-name* [usage]
            +--rw usage
                identityref
            +--rw names*
                string
    +--rw disjointness?
        te-path-disjointness

```

```

+---rw explicit-route-objects-always
+---rw route-object-exclude-always* [index]
+---rw index                                uint32
+---rw (type)?
+---:(numbered-node-hop)
+---rw numbered-node-hop
+---rw node-id          te-node-id
+---rw hop-type?        te-hop-type
+---:(numbered-link-hop)
+---rw numbered-link-hop
+---rw link-tp-id        te-tp-id
+---rw hop-type?        te-hop-type
+---rw direction?
+---rw                      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?
+---rw                      te-link-direction
+---:(as-number)
+---rw as-number-hop
+---rw as-number          inet:as-number
+---rw hop-type?          te-hop-type
+---:(label)
+---rw label-hop
+---rw te-label
+---rw (technology)?
+---:(generic)
+---rw generic?
+---rw                      rt-types:generalized
- label
+---rw direction?
+---rw                      te-label-direction
+---rw route-object-include-exclude* [index]
+---rw explicit-route-usage?
+---rw identityref
+---rw index                                uint32
+---rw (type)?
+---:(numbered-node-hop)
+---rw numbered-node-hop
+---rw node-id          te-node-id
+---rw hop-type?        te-hop-type
+---:(numbered-link-hop)
+---rw numbered-link-hop
+---rw link-tp-id        te-tp-id
+---rw hop-type?        te-hop-type

```

				<pre> +---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 +---:(as-number) +---rw as-number-hop +---rw as-number      inet:as-number +---rw hop-type?      te-hop-type +---:(label) +---rw label-hop +---rw te-label +---rw (technology)?     +---:(generic)         +---rw generic?             rt-types:generalized </pre>
-label				<pre> +---rw direction?     te-label-direction +---:(srlg) +---rw srlg +---rw srlg?      uint32 +---rw path-in-segment! +---rw label-restrictions +---rw label-restriction* [index] +---rw restriction?      enumeration +---rw index              uint32 +---rw label-start +---rw te-label +---rw (technology)?     +---:(generic)         +---rw generic?             rt-types:generalized-la </pre>
bel				<pre> +---rw direction?     te-label-direction +---rw label-end +---rw te-label +---rw (technology)?     +---:(generic)         +---rw generic?             rt-types:generalized-la </pre>
bel				<pre> +---rw direction? </pre>

					<pre> te-label-direction +--rw label-step     +--rw (technology)?         +--:(generic)             +--rw generic?    int32 +--rw range-bitmap?    yang:hex-string +--rw path-out-segment!     +--rw label-restrictions         +--rw label-restriction* [index]             +--rw restriction?    enumeration             +--rw index          uint32 +--rw label-start     +--rw te-label         +--rw (technology)?             +--:(generic)                 +--rw generic?                     rt-types:generalized-la </pre>
bel					
					<pre> +--rw direction?     te-label-direction +--rw label-end     +--rw te-label         +--rw (technology)?             +--:(generic)                 +--rw generic?                     rt-types:generalized-la </pre>
bel					
					<pre> +--rw direction?     te-label-direction +--rw label-step     +--rw (technology)?         +--:(generic)             +--rw generic?    int32 +--rw range-bitmap?    yang:hex-string +--ro computed-paths-properties     +--ro computed-path-properties* [k-index]         +--ro k-index          uint8 +--ro path-properties     +--ro path-metric* [metric-type]         +--ro metric-type             identityref         +--ro accumulative-value?    uint64 +--ro path-affinities-values     +--ro path-affinities-value* [usage]         +--ro usage          identityref         +--ro value?    admin-groups +--ro path-affinity-names     +--ro path-affinity-name* [usage] </pre>

```

+---ro usage          identityref
+---ro affinity-name* [name]
    +---ro name      string
+---ro path-srlgs-lists
    +---ro path-srlgs-list* [usage]
        +---ro usage    identityref
        +---ro values*   srlg
+---ro path-srlgs-names
    +---ro path-srlgs-name* [usage]
        +---ro usage    identityref
        +---ro names*   string
+---ro path-route-objects
    +---ro path-route-object* [index]
        +---ro index
            |
            uint32
        +---ro (type)?
            +---:(numbered-node-hop)
                +---ro numbered-node-hop
                    +---ro node-id
                        |
                        te-node-id
                    +---ro hop-type?
                        |
                        te-hop-type
            +---:(numbered-link-hop)
                +---ro numbered-link-hop
                    +---ro link-tp-id
                        |
                        te-tp-id
                    +---ro hop-type?
                        |
                        te-hop-type
                    +---ro direction?
                        |
                        te-link-direction
            +---:(unnumbered-link-hop)
                +---ro unnumbered-link-hop
                    +---ro link-tp-id
                        |
                        te-tp-id
                    +---ro node-id
                        |
                        te-node-id
                    +---ro hop-type?
                        |
                        te-hop-type
                    +---ro direction?
                        |
                        te-link-direction
            +---:(as-number)
                +---ro as-number-hop
                    +---ro as-number
                        |
                        inet:as-number
                    +---ro hop-type?
                        |
                        te-hop-type
            +---:(label)
                +---ro label-hop

```

```

+---ro te-label
+---ro (technology)?
|   +---:(generic)
|       +---ro generic?
|           rt-types:ge
neralized-label
+---ro direction?
te-label-directio
n
+---ro te-bandwidth
|   +---ro (technology)?
|       +---:(generic)
|           +---ro generic?   te-bandwidth
+---ro disjointness-type?
te-types:te-path-disjointness
+---ro computed-path-error-infos
+---ro computed-path-error-info* []
+---ro error-description?   string
+---ro error-timestamp?
|   yang:date-and-time
+---ro error-reason?       identityref
+---ro lsp-provisioning-error-infos
+---ro lsp-provisioning-error-info* []
+---ro error-description?   string
+---ro error-timestamp?
|   yang:date-and-time
+---ro error-node-id?
|   te-types:te-node-id
+---ro error-link-id?
|   te-types:te-tp-id
+---ro lsp-id?             uint16
+---ro lsps
|   +---ro lsp* [node lsp-id]
|       +---ro tunnel-name?
|           -> /te/lsps/lsp/tunnel-name
|       +---ro node         -> /te/lsps/lsp/node
|       +---ro lsp-id       -> /te/lsps/lsp/lsp-id
+---rw candidate-secondary-reverse-paths
+---rw candidate-secondary-reverse-path*
|   [secondary-path]
+---rw secondary-path      leafref
+---rw candidate-secondary-paths
+---rw candidate-secondary-path* [secondary-path]
+---rw secondary-path      leafref
+---ro active?             boolean
+---rw secondary-paths
+---rw secondary-path* [name]
+---rw name                string

```



```

+--rw path-computation-method?          identityref
+--rw path-computation-server
|   +--rw id?        te-gen-node-id
|   +--rw type?      enumeration
+--rw compute-only?                      empty
+--rw use-path-computation?              boolean
+--rw lockdown?                          empty
+--ro path-scope?                        identityref
+--rw preference?                        uint8
+--rw association-objects
|   +--rw association-object* [association-key]
|   |   +--rw association-key      string
|   |   +--rw type?                identityref
|   |   +--rw id?                  uint16
|   |   +--rw source
|   |   |   +--rw id?        te-gen-node-id
|   |   |   +--rw type?      enumeration
|   +--rw association-object-extended*
|   |   [association-key]
|   |   +--rw association-key      string
|   |   +--rw type?                identityref
|   |   +--rw id?                  uint16
|   |   +--rw source
|   |   |   +--rw id?        te-gen-node-id
|   |   |   +--rw type?      enumeration
|   |   +--rw global-source?      uint32
|   |   +--rw extended-id?        yang:hex-string
+--rw optimizations
|   +--rw (algorithm)?
|   |   +--:(metric) {path-optimization-metric}?
|   |   |   +--rw optimization-metric* [metric-type]
|   |   |   |   +--rw metric-type
|   |   |   |   |   identityref
|   |   |   |   +--rw weight?
|   |   |   |   |   uint8
|   |   |   +--rw explicit-route-exclude-objects
|   |   |   |   +--rw route-object-exclude-object*
|   |   |   |   |   [index]
|   |   |   |   |   +--rw index
|   |   |   |   |   |   uint32
|   |   |   |   +--rw (type)?
|   |   |   |   |   +--:(numbered-node-hop)
|   |   |   |   |   |   +--rw numbered-node-hop
|   |   |   |   |   |   |   +--rw node-id
|   |   |   |   |   |   |   |   te-node-id
|   |   |   |   |   |   |   +--rw hop-type?
|   |   |   |   |   |   |   |   te-hop-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
+--:(as-number)
+--rw as-number-hop
+--rw as-number
|   inet:as-number
+--rw hop-type?
|   te-hop-type
+--:(label)
+--rw label-hop
+--rw te-label
|   +--rw (technology)?
|   |   +--:(generic)
|   |   +--rw generic?
|   |   rt-types:ge
+--rw direction?
|   te-label-direction
+--:(srlg)
+--rw srlg
|   +--rw srlg?   uint32
+--rw explicit-route-include-objects
+--rw route-object-include-object*
|   [index]
+--rw index
|   uint32
+--rw (type)?
+--:(numbered-node-hop)
+--rw numbered-node-hop
|   +--rw node-id
|   |   te-node-id
+--rw hop-type?

```

						te-hop-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
					+++:(as-number)	
					+++rw	as-number-hop
					+++rw	as-number
						inet:as-number
					+++rw	hop-type?
						te-hop-type
					+++:(label)	
					+++rw	label-hop
					+++rw	te-label
					+++rw	(technology)?
						+++:(generic)
						+++rw generic?
						rt-types:ge
neralized-label						
						+++rw direction?
						te-label-directio
n						
					+++rw	tiebreakers
					+++rw	tiebreaker* [tiebreaker-type]
					+++rw	tiebreaker-type identityref
					+++:(objective-function)	
					{path-optimization-objective-function}?	
					+++rw	objective-function
					+++rw	objective-function-type?
					identityref	
					+++rw	named-path-constraint? leafref
					{te-types:named-path-constraints}?	
					+++rw	te-bandwidth
						+++rw (technology)?

```

+---:(generic)
+---rw generic?      te-bandwidth
+---rw link-protection?      identityref
+---rw setup-priority?      uint8
+---rw hold-priority?      uint8
+---rw signaling-type?      identityref
+---rw path-metric-bounds
+---rw path-metric-bound* [metric-type]
+---rw metric-type      identityref
+---rw upper-bound?      uint64
+---rw path-affinities-values
+---rw path-affinities-value* [usage]
+---rw usage      identityref
+---rw value?      admin-groups
+---rw path-affinity-names
+---rw path-affinity-name* [usage]
+---rw usage      identityref
+---rw affinity-name* [name]
+---rw name      string
+---rw path-srlgs-lists
+---rw path-srlgs-list* [usage]
+---rw usage      identityref
+---rw values*      srlg
+---rw path-srlgs-names
+---rw path-srlgs-name* [usage]
+---rw usage      identityref
+---rw names*      string
+---rw disjointness?
+---rw te-path-disjointness
+---rw explicit-route-objects-always
+---rw route-object-exclude-always* [index]
+---rw index      uint32
+---rw (type)?
+---:(numbered-node-hop)
+---rw numbered-node-hop
+---rw node-id      te-node-id
+---rw hop-type?      te-hop-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

```

				<pre> +--:(as-number)     +--rw as-number-hop         +--rw as-number      inet:as-number         +--rw hop-type?      te-hop-type +--:(label)     +--rw label-hop         +--rw te-label             +--rw (technology)?                 +--:(generic)                     +--rw generic?                         rt-types:generalized-la </pre>
bel				<pre> +--rw direction?     te-label-direction +--rw route-object-include-exclude* [index] +--rw explicit-route-usage?      identityref +--rw index                      uint32 +--rw (type)?     +--:(numbered-node-hop)         +--rw numbered-node-hop             +--rw node-id      te-node-id             +--rw hop-type?    te-hop-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     +--:(as-number)         +--rw as-number-hop             +--rw as-number      inet:as-number             +--rw hop-type?      te-hop-type     +--:(label)         +--rw label-hop             +--rw te-label                 +--rw (technology)?                     +--:(generic)                         +--rw generic?                             rt-types:generalized-la </pre>
bel				<pre> +--rw direction?     te-label-direction +--:(srlg) </pre>

```

        +---rw srlg
            +---rw srlg?   uint32
+---rw path-in-segment!
    +---rw label-restrictions
        +---rw label-restriction* [index]
            +---rw restriction?   enumeration
            +---rw index          uint32
            +---rw label-start
                +---rw te-label
                    +---rw (technology)?
                        +---:(generic)
                            +---rw generic?
                                rt-types:generalized-label
                    +---rw direction?
                        te-label-direction
            +---rw label-end
                +---rw te-label
                    +---rw (technology)?
                        +---:(generic)
                            +---rw generic?
                                rt-types:generalized-label
                    +---rw direction?
                        te-label-direction
            +---rw label-step
                +---rw (technology)?
                    +---:(generic)
                        +---rw generic?   int32
            +---rw range-bitmap?   yang:hex-string
+---rw path-out-segment!
    +---rw label-restrictions
        +---rw label-restriction* [index]
            +---rw restriction?   enumeration
            +---rw index          uint32
            +---rw label-start
                +---rw te-label
                    +---rw (technology)?
                        +---:(generic)
                            +---rw generic?
                                rt-types:generalized-label
                    +---rw direction?
                        te-label-direction
            +---rw label-end
                +---rw te-label
                    +---rw (technology)?
                        +---:(generic)
                            +---rw generic?
                                rt-types:generalized-label
                    +---rw direction?

```

```

|                                     te-label-direction
|   +---rw label-step
|   |   +---rw (technology)?
|   |   |   +---:(generic)
|   |   |   +---rw generic?    int32
|   +---rw range-bitmap?    yang:hex-string
+---rw protection
|   +---rw enable?                                boolean
|   +---rw protection-type?                        identityref
|   +---rw protection-reversion-disable?          boolean
|   +---rw hold-off-time?                          uint32
|   +---rw wait-to-revert?                        uint16
|   +---rw aps-signal-id?                          uint8
+---rw restoration
|   +---rw enable?                                boolean
|   +---rw restoration-type?
|   |   identityref
|   +---rw restoration-scheme?
|   |   identityref
|   +---rw restoration-reversion-disable?          boolean
|   +---rw hold-off-time?                          uint32
|   +---rw wait-to-restore?                        uint16
|   +---rw wait-to-revert?                        uint16
+---ro computed-paths-properties
|   +---ro computed-path-properties* [k-index]
|   |   +---ro k-index                          uint8
|   |   +---ro path-properties
|   |   |   +---ro path-metric* [metric-type]
|   |   |   |   +---ro metric-type              identityref
|   |   |   |   +---ro accumulative-value?      uint64
|   |   |   +---ro path-affinities-values
|   |   |   |   +---ro path-affinities-value* [usage]
|   |   |   |   |   +---ro usage                identityref
|   |   |   |   |   +---ro value?              admin-groups
|   |   |   +---ro path-affinity-names
|   |   |   |   +---ro path-affinity-name* [usage]
|   |   |   |   |   +---ro usage                identityref
|   |   |   |   |   +---ro affinity-name* [name]
|   |   |   |   |   |   +---ro name              string
|   |   +---ro path-srlgs-lists
|   |   |   +---ro path-srlgs-list* [usage]
|   |   |   |   +---ro usage                identityref
|   |   |   |   +---ro values*              srlg
|   |   +---ro path-srlgs-names
|   |   |   +---ro path-srlgs-name* [usage]
|   |   |   |   +---ro usage                identityref
|   |   |   |   +---ro names*              string
|   +---ro path-route-objects

```

				+--ro path-route-object* [index]
				+--ro index
				uint32
				+--ro (type)?
				+--:(numbered-node-hop)
				+--ro numbered-node-hop
				+--ro node-id     te-node-id
				+--ro hop-type?
				te-hop-type
				+--:(numbered-link-hop)
				+--ro numbered-link-hop
				+--ro link-tp-id   te-tp-id
				+--ro hop-type?
				te-hop-type
				+--ro direction?
				te-link-direction
				+--:(unnumbered-link-hop)
				+--ro unnumbered-link-hop
				+--ro link-tp-id   te-tp-id
				+--ro node-id
				te-node-id
				+--ro hop-type?
				te-hop-type
				+--ro direction?
				te-link-direction
				+--:(as-number)
				+--ro as-number-hop
				+--ro as-number
				inet:as-number
				+--ro hop-type?
				te-hop-type
				+--:(label)
				+--ro label-hop
				+--ro te-label
				+--ro (technology)?
				+--:(generic)
				+--ro generic?
				rt-types:gener
				+--ro direction?
				te-label-direction
				+--ro te-bandwidth
				+--ro (technology)?
				+--:(generic)
				+--ro generic?   te-bandwidth
				+--ro disjointness-type?
				te-types:te-path-disjointness
				+--ro computed-path-error-infos

alized-label



```

    +---ro computed-path-error-info* []
    |   +---ro error-description?    string
    |   +---ro error-timestamp?     yang:date-and-time
    |   +---ro error-reason?        identityref
+---ro lsp-provisioning-error-infos
|   +---ro lsp-provisioning-error-info* []
|   |   +---ro error-description?    string
|   |   +---ro error-timestamp?     yang:date-and-time
|   |   +---ro error-node-id?       te-types:te-node-id
|   |   +---ro error-link-id?       te-types:te-tp-id
|   |   +---ro lsp-id?              uint16
+---ro lsps
|   +---ro lsp* [node lsp-id]
|   |   +---ro tunnel-name?
|   |   |   -> /te/lsps/lsp/tunnel-name
|   |   +---ro node                 -> /te/lsps/lsp/node
|   |   +---ro lsp-id               -> /te/lsps/lsp/lsp-id
+---rw secondary-reverse-paths
|   +---rw secondary-reverse-path* [name]
|   |   +---rw name                  string
|   |   +---rw path-computation-method? identityref
|   |   +---rw path-computation-server
|   |   |   +---rw id?               te-gen-node-id
|   |   |   +---rw type?             enumeration
|   |   +---rw compute-only?         empty
|   |   +---rw use-path-computation? boolean
|   |   +---rw lockdown?             empty
|   |   +---ro path-scope?           identityref
|   |   +---rw preference?           uint8
+---rw association-objects
|   +---rw association-object* [association-key]
|   |   +---rw association-key        string
|   |   +---rw type?                 identityref
|   |   +---rw id?                   uint16
|   |   +---rw source
|   |   |   +---rw id?               te-gen-node-id
|   |   |   +---rw type?             enumeration
+---rw association-object-extended*
|   [association-key]
|   +---rw association-key            string
|   +---rw type?                     identityref
|   +---rw id?                       uint16
|   +---rw source
|   |   +---rw id?                   te-gen-node-id
|   |   +---rw type?                 enumeration
+---rw global-source?                uint32
+---rw extended-id?                  yang:hex-string
+---rw optimizations

```

```

+--rw (algorithm)?
+--:(metric) {path-optimization-metric}?
+--rw optimization-metric* [metric-type]
+--rw metric-type
|   identityref
+--rw weight?
|   uint8
+--rw explicit-route-exclude-objects
+--rw route-object-exclude-object*
|   [index]
+--rw index
|   uint32
+--rw (type)?
+--:(numbered-node-hop)
+--rw numbered-node-hop
+--rw node-id
|   te-node-id
+--rw hop-type?
|   te-hop-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
+--:(as-number)
+--rw as-number-hop
+--rw as-number
|   inet:as-number
+--rw hop-type?
|   te-hop-type
+--:(label)
+--rw label-hop
+--rw te-label
+--rw (technology)?
|   +--:(generic)

```

```

n
neralized-label

rt-types:generic
direction?
te-label-direction

+--:(srlg)
+--rw srlg
+--rw srlg? uint32
+--rw explicit-route-include-objects
+--rw route-object-include-object*
[index]
+--rw index
|
uint32
+--rw (type)?
+--:(numbered-node-hop)
+--rw numbered-node-hop
+--rw node-id
|
te-node-id
+--rw hop-type?
te-hop-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
+--:(as-number)
+--rw as-number-hop
+--rw as-number
|
inet:as-number
+--rw hop-type?
te-hop-type
+--:(label)
+--rw label-hop
+--rw te-label

```

```

+---rw (technology)?
+---:(generic)
+---rw generic?
rt-types:ge

neralized-label

+---rw direction?
te-label-directio

n

+---rw tiebreakers
+---rw tiebreaker* [tiebreaker-type]
+---rw tiebreaker-type identityref
+---:(objective-function)
{path-optimization-objective-function}?
+---rw objective-function
+---rw objective-function-type?
identityref
+---rw named-path-constraint? leafref
{te-types:named-path-constraints}?
+---rw te-bandwidth
+---rw (technology)?
+---:(generic)
+---rw generic? te-bandwidth
+---rw link-protection? identityref
+---rw setup-priority? uint8
+---rw hold-priority? uint8
+---rw signaling-type? identityref
+---rw path-metric-bounds
+---rw path-metric-bound* [metric-type]
+---rw metric-type identityref
+---rw upper-bound? uint64
+---rw path-affinities-values
+---rw path-affinities-value* [usage]
+---rw usage identityref
+---rw value? admin-groups
+---rw path-affinity-names
+---rw path-affinity-name* [usage]
+---rw usage identityref
+---rw affinity-name* [name]
+---rw name string
+---rw path-srlgs-lists
+---rw path-srlgs-list* [usage]
+---rw usage identityref
+---rw values* srlg
+---rw path-srlgs-names
+---rw path-srlgs-name* [usage]
+---rw usage identityref
+---rw names* string
+---rw disjointness?

```

				te-path-disjointness
			+--rw	explicit-route-objects-always
			+--rw	route-object-exclude-always* [index]
			+--rw	index uint32
			+--rw	(type)?
			+--:	(numbered-node-hop)
			+--rw	numbered-node-hop
			+--rw	node-id te-node-id
			+--rw	hop-type? te-hop-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
			+--:	(as-number)
			+--rw	as-number-hop
			+--rw	as-number inet:as-number
			+--rw	hop-type? te-hop-type
			+--:	(label)
			+--rw	label-hop
			+--rw	te-label
			+--rw	(technology)?
			+--:	(generic)
			+--rw	generic?
				rt-types:generalized-la
			+--rw	direction?
				te-label-direction
			+--rw	route-object-include-exclude* [index]
			+--rw	explicit-route-usage? identityref
			+--rw	index uint32
			+--rw	(type)?
			+--:	(numbered-node-hop)
			+--rw	numbered-node-hop
			+--rw	node-id te-node-id
			+--rw	hop-type? te-hop-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)

				<pre> +--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 +--:(as-number) +--rw as-number-hop +--rw as-number        inet:as-number +--rw hop-type?        te-hop-type +--:(label) +--rw label-hop +--rw te-label +--rw (technology)? +--:(generic) +--rw generic? rt-types:generalized-la </pre>
bel				<pre> +--rw direction? te-label-direction +--:(srlg) +--rw srlg +--rw srlg?   uint32 +--rw path-in-segment! +--rw label-restrictions +--rw label-restriction* [index] +--rw restriction?       enumeration +--rw index               uint32 +--rw label-start +--rw te-label +--rw (technology)? +--:(generic) +--rw generic? rt-types:generalized-label +--rw direction? te-label-direction +--rw label-end +--rw te-label +--rw (technology)? +--:(generic) +--rw generic? rt-types:generalized-label +--rw direction? te-label-direction +--rw label-step +--rw (technology)? +--:(generic) +--rw generic?   int32 +--rw range-bitmap?   yang:hex-string </pre>

```

+--rw path-out-segment!
  +--rw label-restrictions
    +--rw label-restriction* [index]
      +--rw restriction?    enumeration
      +--rw index           uint32
      +--rw label-start
        +--rw te-label
          +--rw (technology)?
            +--:(generic)
              +--rw generic?
                rt-types:generalized-label
          +--rw direction?
            te-label-direction
      +--rw label-end
        +--rw te-label
          +--rw (technology)?
            +--:(generic)
              +--rw generic?
                rt-types:generalized-label
          +--rw direction?
            te-label-direction
      +--rw label-step
        +--rw (technology)?
          +--:(generic)
            +--rw generic?    int32
        +--rw range-bitmap?  yang:hex-string
+--rw protection
  +--rw enable?                boolean
  +--rw protection-type?       identityref
  +--rw protection-reversion-disable? boolean
  +--rw hold-off-time?         uint32
  +--rw wait-to-revert?        uint16
  +--rw aps-signal-id?         uint8
+--rw restoration
  +--rw enable?                boolean
  +--rw restoration-type?
    | identityref
  +--rw restoration-scheme?
    | identityref
  +--rw restoration-reversion-disable? boolean
  +--rw hold-off-time?         uint32
  +--rw wait-to-restore?       uint16
  +--rw wait-to-revert?        uint16
+--ro computed-paths-properties
  +--ro computed-path-properties* [k-index]
    +--ro k-index              uint8
    +--ro path-properties
      +--ro path-metric* [metric-type]

```

```

|   +---ro metric-type          identityref
|   +---ro accumulative-value?  uint64
+---ro path-affinities-values
|   +---ro path-affinities-value* [usage]
|   +---ro usage                identityref
|   +---ro value?               admin-groups
+---ro path-affinity-names
|   +---ro path-affinity-name* [usage]
|   +---ro usage                identityref
|   +---ro affinity-name* [name]
|   +---ro name                 string
+---ro path-srlgs-lists
|   +---ro path-srlgs-list* [usage]
|   +---ro usage                identityref
|   +---ro values*             srlg
+---ro path-srlgs-names
|   +---ro path-srlgs-name* [usage]
|   +---ro usage                identityref
|   +---ro names*              string
+---ro path-route-objects
|   +---ro path-route-object* [index]
|   +---ro index
|   |   uint32
|   +---ro (type)?
|   |   +---:(numbered-node-hop)
|   |   |   +---ro numbered-node-hop
|   |   |   |   +---ro node-id      te-node-id
|   |   |   |   +---ro hop-type?
|   |   |   |   |   te-hop-type
|   |   |   +---:(numbered-link-hop)
|   |   |   |   +---ro numbered-link-hop
|   |   |   |   |   +---ro link-tp-id  te-tp-id
|   |   |   |   |   +---ro hop-type?
|   |   |   |   |   |   te-hop-type
|   |   |   |   |   +---ro direction?
|   |   |   |   |   |   te-link-direction
|   |   |   +---:(unnumbered-link-hop)
|   |   |   |   +---ro unnumbered-link-hop
|   |   |   |   |   +---ro link-tp-id  te-tp-id
|   |   |   |   |   +---ro node-id
|   |   |   |   |   |   te-node-id
|   |   |   |   |   +---ro hop-type?
|   |   |   |   |   |   te-hop-type
|   |   |   |   |   +---ro direction?
|   |   |   |   |   |   te-link-direction
|   |   +---:(as-number)
|   |   |   +---ro as-number-hop
|   |   |   +---ro as-number

```



```

    |      inet:as-number
    |      +---ro hop-type?
    |          te-hop-type
    |      +---:(label)
    |          +---ro label-hop
    |              +---ro te-label
    |                  +---ro (technology)?
    |                      +---:(generic)
    |                          +---ro generic?
    |                              rt-types:gener
alized-label
    |      +---ro direction?
    |          te-label-direction
    |      +---ro te-bandwidth
    |          +---ro (technology)?
    |              +---:(generic)
    |                  +---ro generic?   te-bandwidth
    |      +---ro disjointness-type?
    |          te-types:te-path-disjointness
+---ro computed-path-error-infos
|   +---ro computed-path-error-info* []
|       +---ro error-description?     string
|       +---ro error-timestamp?        yang:date-and-time
|       +---ro error-reason?           identityref
+---ro lsp-provisioning-error-infos
|   +---ro lsp-provisioning-error-info* []
|       +---ro error-description?     string
|       +---ro error-timestamp?        yang:date-and-time
|       +---ro error-node-id?         te-types:te-node-id
|       +---ro error-link-id?         te-types:te-tp-id
|       +---ro lsp-id?                uint16
+---ro lsps
|   +---ro lsp* [node lsp-id]
|       +---ro tunnel-name?
|           | -> /te/lsp/<lsp/tunnel-name
|       +---ro node                    -> /te/lsp/<lsp/node
|       +---ro lsp-id                 -> /te/lsp/<lsp/lsp-id
+---x tunnel-action
|   +---w input
|       | +----w action-type?     identityref
|   +---ro output
|       +---ro action-result?     identityref
+---x protection-external-commands
|   +---w input
|       +---w protection-external-command?
|           | identityref
|   +---w protection-group-ingress-node-id?
|       | te-types:te-node-id
```

```

|         +---w protection-group-egress-node-id?
|         |         te-types:te-node-id
|         +---w path-ref?                                path-ref
|         +---w traffic-type?
|         |         enumeration
|         +---w extra-traffic-tunnel-ref?                tunnel-ref
+---ro lsp
+---ro lsp* [tunnel-name lsp-id node]
+---ro tunnel-name                                string
+---ro lsp-id                                    uint16
+---ro node
|         te-types:te-node-id
+---ro source?
|         te-types:te-node-id
+---ro destination?
|         te-types:te-node-id
+---ro tunnel-id?                                uint16
+---ro extended-tunnel-id?                       yang:dotted-quad
+---ro operational-state?                       identityref
+---ro signaling-type?                         identityref
+---ro origin-type?                           enumeration
+---ro lsp-resource-status?                   enumeration
+---ro lockout-of-normal?                     boolean
+---ro freeze?                               boolean
+---ro lsp-protection-role?                   enumeration
+---ro lsp-protection-state?                 identityref
+---ro protection-group-ingress-node-id?
|         te-types:te-node-id
+---ro protection-group-egress-node-id?
|         te-types:te-node-id
+---ro lsp-record-route-information
+---ro lsp-record-route-information* [index]
+---ro index                                    uint32
+---ro (type)?
+---:(numbered-node-hop)
|         +---ro numbered-node-hop
|         |         +---ro node-id      te-node-id
|         |         +---ro flags*      path-attribute-flags
+---:(numbered-link-hop)
|         +---ro numbered-link-hop
|         |         +---ro link-tp-id   te-tp-id
|         |         +---ro flags*      path-attribute-flags
+---:(unnumbered-link-hop)
|         +---ro unnumbered-link-hop
|         |         +---ro link-tp-id   te-tp-id
|         |         +---ro node-id?    te-node-id
|         |         +---ro flags*      path-attribute-flags
+---:(label)

```

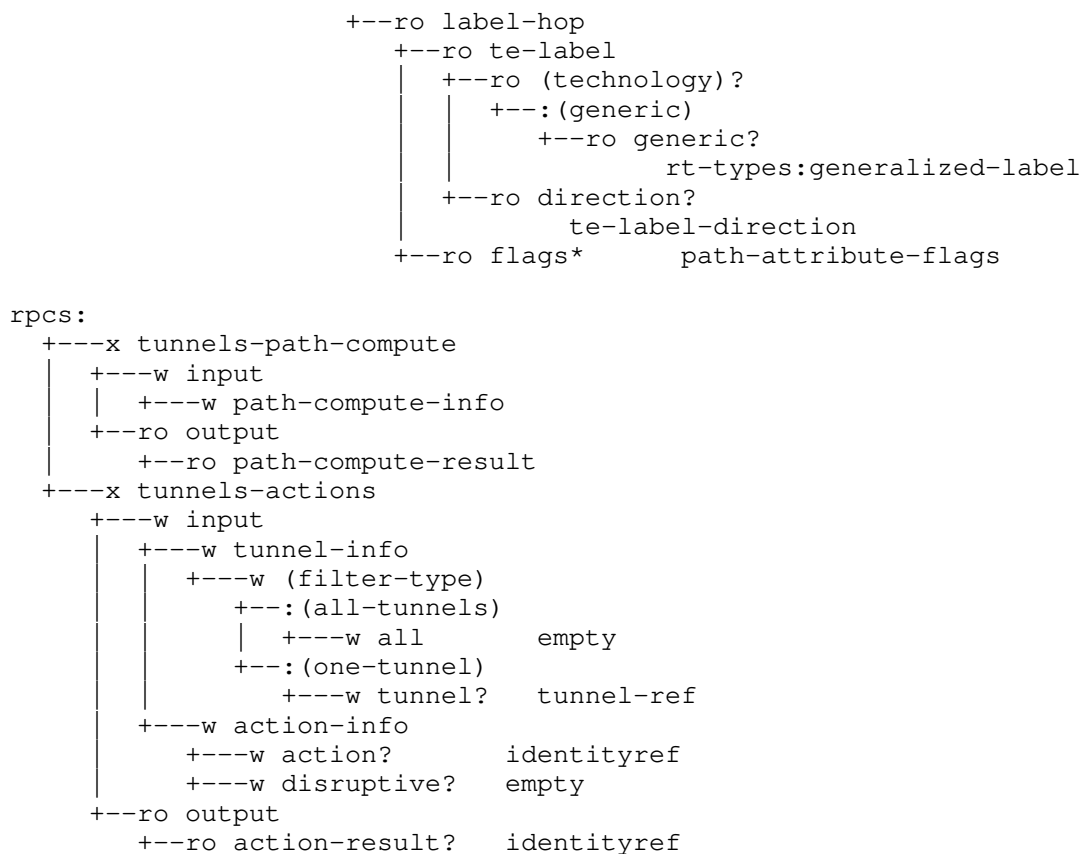


Figure 8: TE Tunnel generic model YANG tree diagram

### 5.3. YANG Module

The generic TE YANG module 'ietf-te' imports the following modules:

- ```
* ietf-yang-types and ietf-inet-types defined in [RFC6991]
* ietf-te-types defined in [RFC8776]
```

This module references the following documents: [RFC6991], [RFC4875], [RFC7551], [RFC4206], [RFC4427], [RFC4872], [RFC3945], [RFC3209], [RFC6780], [RFC8800], and [RFC7308].

```
<CODE BEGINS> file "ietf-te@2021-10-22.yang"
module ietf-te {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-te";

  /* Replace with IANA when assigned */

  prefix te;

  /* Import TE generic types */

  import ietf-te-types {
    prefix te-types;
    reference
      "RFC8776: Common YANG Data Types for Traffic Engineering.";
  }
  import ietf-inet-types {
    prefix inet;
    reference
      "RFC6991: Common YANG Data Types.";
  }
  import ietf-yang-types {
    prefix yang;
    reference
      "RFC6991: Common YANG Data Types.";
  }
}

organization
  "IETF Traffic Engineering Architecture and Signaling (TEAS)
  Working Group.";
contact
  "WG Web:  <http://tools.ietf.org/wg/teas/>
  WG List:  <mailto:teas@ietf.org>

  Editor:   Tarek Saad
            <mailto:tsaad@juniper.net>

  Editor:   Rakesh Gandhi
            <mailto:rgandhi@cisco.com>

  Editor:   Vishnu Pavan Beeram
            <mailto:vbeeram@juniper.net>

  Editor:   Himanshu Shah
            <mailto:hshah@ciena.com>

  Editor:   Xufeng Liu
            <mailto:xufeng.liu.ietf@gmail.com>
```

```
Editor:   Igor Bryskin
         <mailto:i_bryskin@yahoo.com>;

description
  "YANG data module for TE configuration, state, and RPCs.
  The model fully conforms to the Network Management
  Datastore Architecture (NMDA).

  Copyright (c) 2019 IETF Trust and the persons
  identified as authors of the code.  All rights reserved.

  Redistribution and use in source and binary forms, with or
  without modification, is permitted pursuant to, and subject
  to the license terms contained in, the Simplified BSD License
  set forth in Section 4.c of the IETF Trust's Legal Provisions
  Relating to IETF Documents
  (https://trustee.ietf.org/license-info).
  This version of this YANG module is part of RFC XXXX; see
  the RFC itself for full legal notices.";

// RFC Ed.: replace XXXX with actual RFC number and remove this
// note.
// RFC Ed.: update the date below with the date of RFC publication
// and remove this note.

revision 2021-10-22 {
  description
    "Latest update to TE generic YANG module.";
  reference
    "RFCXXXX: A YANG Data Model for Traffic Engineering Tunnels
    and Interfaces.";
}

identity path-computation-error-reason {
  description
    "Base identity for path computation error reasons.";
}

identity path-computation-error-no-topology {
  base path-computation-error-reason;
  description
    "Path computation has failed because there is no topology
    with the provided topology-identifier.";
}

identity path-computation-error-no-dependent-server {
  base path-computation-error-reason;
  description
    "Path computation has failed because one or more dependent
```

```
        path computation servers are unavailable.
        The dependent path computation server could be
        a Backward-Recursive Path Computation (BRPC) downstream
        PCE or a child PCE.";
    reference
        "RFC5441, RFC8685";
}

identity path-computation-error-pce-unavailable {
    base path-computation-error-reason;
    description
        "Path computation has failed because PCE is not available.";
    reference
        "RFC5440";
}

identity path-computation-error-no-inclusion-hop {
    base path-computation-error-reason;
    description
        "Path computation has failed because there is no
        node or link provided by one or more inclusion hops.";
    reference
        "RFC8685";
}

identity path-computation-error-destination-unknown-in-domain {
    base path-computation-error-reason;
    description
        "Path computation has failed because the destination node is
        unknown in indicated destination domain.";
    reference
        "RFC8685";
}

identity path-computation-error-no-resource {
    base path-computation-error-reason;
    description
        "Path computation has failed because there is no
        available resource in one or more domains.";
    reference
        "RFC8685";
}

identity path-computation-error-child-pce-unresponsive {
    base path-computation-error-reason;
    description
        "Path computation has failed because child PCE is not
        responsive.";
```

```
        reference
        "RFC8685";
    }

    identity path-computation-error-destination-domain-unknown {
        base path-computation-error-reason;
        description
            "Path computation has failed because the destination domain
            was unknown.";
        reference
            "RFC8685";
    }

    identity path-computation-error-p2mp {
        base path-computation-error-reason;
        description
            "Path computation has failed because of P2MP reachability
            problem.";
        reference
            "RFC8306";
    }

    identity path-computation-error-no-gco-migration {
        base path-computation-error-reason;
        description
            "Path computation has failed because of no Global Concurrent
            Optimization (GCO) migration path found.";
        reference
            "RFC5557";
    }

    identity path-computation-error-no-gco-solution {
        base path-computation-error-reason;
        description
            "Path computation has failed because of no GCO solution
            found.";
        reference
            "RFC5557";
    }

    identity path-computation-error-path-not-found {
        base path-computation-error-reason;
        description
            "Path computation no path found error reason.";
        reference
            "RFC5440";
    }
}
```

```
identity path-computation-error-pks-expansion {
  base path-computation-error-reason;
  description
    "Path computation has failed because of Path-Key Subobject
    (PKS) expansion failure.";
  reference
    "RFC5520";
}

identity path-computation-error-brpc-chain-unavailable {
  base path-computation-error-reason;
  description
    "Path computation has failed because PCE BRPC chain
    unavailable.";
  reference
    "RFC5441";
}

identity path-computation-error-source-unknown {
  base path-computation-error-reason;
  description
    "Path computation has failed because source node is unknown.";
  reference
    "RFC5440";
}

identity path-computation-error-destination-unknown {
  base path-computation-error-reason;
  description
    "Path computation has failed because destination node is
    unknown.";
  reference
    "RFC5440";
}

identity path-computation-error-no-server {
  base path-computation-error-reason;
  description
    "Path computation has failed because path computation
    server is unavailable.";
  reference
    "RFC5440";
}

identity tunnel-actions-type {
  description
    "TE tunnel actions type.";
}
```



```
identity tunnel-action-reoptimize {
  base tunnel-actions-type;
  description
    "Reoptimize tunnel action type.";
}

identity tunnel-admin-auto {
  base te-types:tunnel-admin-state-type;
  description
    "Tunnel administrative auto state. The administrative status
    in state datastore transitions to 'tunnel-admin-up' when the
    tunnel used by the client layer, and to 'tunnel-admin-down'
    when it is not used by the client layer.";
}

identity association-type-diversity {
  base te-types:association-type;
  description
    "Association Type diversity used to associate LSPs whose paths
    are to be diverse from each other.";
  reference
    "RFC8800";
}

identity protocol-origin-type {
  description
    "Base identity for protocol origin type.";
}

identity protocol-origin-api {
  base protocol-origin-type;
  description
    "Protocol origin is via Application Programmable Interface
    (API).";
}

identity protocol-origin-pcep {
  base protocol-origin-type;
  description
    "Protocol origin is Path Computation Engine Protocol (PCEP).";
  reference "RFC5440";
}

identity protocol-origin-bgp {
  base protocol-origin-type;
  description
    "Protocol origin is Border Gateway Protocol (BGP).";
  reference "RFC5512";
}

typedef tunnel-ref {
```

```
    type leafref {
      path "/te:te/te:tunnels/te:tunnel/te:name";
    }
    description
      "This type is used by data models that need to reference
       configured TE tunnel.";
  }

  typedef path-ref {
    type union {
      type leafref {
        path "/te:te/te:tunnels/te:tunnel/"
          + "te:primary-paths/te:primary-path/te:name";
      }
      type leafref {
        path "/te:te/te:tunnels/te:tunnel/"
          + "te:secondary-paths/te:secondary-path/te:name";
      }
    }
    description
      "This type is used by data models that need to reference
       configured primary or secondary path of a TE tunnel.";
  }

  typedef te-gen-node-id {
    type union {
      type te-types:te-node-id;
      type inet:ip-address;
    }
    description
      "Generic type that identifies a node in a TE topology.";
  }

  /**
   * TE tunnel generic groupings
   */

  grouping te-generic-node-id {
    description
      "A reusable grouping for a TE generic node identifier.";
    leaf id {
      type te-gen-node-id;
      description
        "The identifier of the node. Can be represented as IP
         address or dotted quad address.";
    }
    leaf type {
      type enumeration {
```

```
    enum ip {
      description
        "IP address representation of the node identifier.";
    }
    enum dotted-quad {
      description
        "Dotted quad address representation of the node
        identifier.";
    }
  }
  description
    "Type of node identifier representation.";
}

grouping primary-path {
  description
    "The tunnel primary path properties.";
  uses path-common-properties;
  uses path-preference;
  uses k-requested-paths;
  uses path-compute-info;
  uses path-state;
}

grouping primary-reverse-path {
  description
    "The tunnel primary reverse path properties.";
  reference
    "RFC7551";
  uses path-common-properties;
  uses path-compute-info;
  uses path-state;
}

grouping secondary-path {
  description
    "The tunnel secondary path properties.";
  uses path-common-properties;
  uses path-preference;
  uses path-compute-info;
  uses protection-restoration-properties;
  uses path-state;
}

grouping secondary-reverse-path {
  description
    "The tunnel secondary reverse path properties.";
```

```
    uses path-common-properties;
    uses path-preference;
    uses path-compute-info;
    uses protection-restoration-properties;
    uses path-state;
}

grouping path-common-properties {
  description
    "Common path attributes.";
  leaf name {
    type string;
    description
      "TE path name.";
  }
  leaf path-computation-method {
    type identityref {
      base te-types:path-computation-method;
    }
    default "te-types:path-locally-computed";
    description
      "The method used for computing the path, either
       locally computed, queried from a server or not
       computed at all (explicitly configured).";
  }
  container path-computation-server {
    when "derived-from-or-self(..path-computation-method, "
      + "'te-types:path-externally-queried') " {
      description
        "The path-computation server when the path is
         externally queried.";
    }
    uses te-generic-node-id;
    description
      "Address of the external path computation
       server.";
  }
  leaf compute-only {
    type empty;
    description
      "When set, the path is computed and updated whenever
       the topology is updated. No resources are committed
       or reserved in the network.";
  }
  leaf use-path-computation {
    when "derived-from-or-self(..path-computation-method, "
      + "'te-types:path-locally-computed') " {
      type boolean;
    }
  }
}
```

```
    default "true";
    description
        "When 'true' indicates the path is dynamically computed
        and/or validated against the Traffic-Engineering Database
        (TED), and when 'false' indicates no validation against
        the TED is required.";
}
leaf lockdown {
    type empty;
    description
        "Indicates no reoptimization to be attempted for this path.";
}
leaf path-scope {
    type identityref {
        base te-types:path-scope-type;
    }
    default "te-types:path-scope-end-to-end";
    config false;
    description
        "Path scope if segment or an end-to-end path.";
}
}

/* This grouping will be re-used in path-computation rpc */

grouping path-compute-info {
    description
        "Attributes used for path computation request.";
    uses tunnel-associations-properties;
    uses te-types:generic-path-optimization;
    leaf named-path-constraint {
        if-feature "te-types:named-path-constraints";
        type leafref {
            path "/te:te/te:globals/te:named-path-constraints/"
                + "te:named-path-constraint/te:name";
        }
        description
            "Reference to a globally defined named path constraint set.";
    }
    uses path-constraints-common;
}

/* This grouping will be re-used in path-computation rpc */

grouping path-preference {
    description
        "The path preference.";
    leaf preference {
```

```
    type uint8 {
      range "1..255";
    }
    default "1";
    description
      "Specifies a preference for this path. The lower the number
       higher the preference.";
  }
}

/* This grouping will be re-used in path-computation rpc */

grouping k-requested-paths {
  description
    "The k-shortest paths requests.";
  leaf k-requested-paths {
    type uint8;
    default "1";
    description
      "The number of k-shortest-paths requested from the path
       computation server and returned sorted by its optimization
       objective. The value 0 all possible paths.";
  }
}

grouping path-properties {
  description
    "TE computed path properties grouping.";
  uses te-types:generic-path-properties {
    augment "path-properties" {
      description
        "additional path properties returned by path computation.";
      uses te-types:te-bandwidth;
      leaf disjointness-type {
        type te-types:te-path-disjointness;
        config false;
        description
          "The type of resource disjointness.
           When reported for a primary path, it represents the
           minimum level of disjointness of all the secondary
           paths.
           When reported for a secondary path, it represents the
           disjointness of the secondary path.";
      }
    }
  }
}
```

```
grouping path-state {
  description
    "TE per path state parameters.";
  uses path-computation-response;
  uses lsp-provisioning-error-info {
    augment "lsp-provisioning-error-infos/"
      + "lsp-provisioning-error-info" {
      description
        "Augmentation of LSP provisioning information under a
        specific path.";
      leaf lsp-id {
        type uint16;
        description
          "The LSP-ID for which path computation was performed.";
      }
    }
  }
}
container lsps {
  config false;
  description
    "The TE LSPs container.";
  list lsp {
    key "node lsp-id";
    description
      "List of LSPs associated with the tunnel.";
    leaf tunnel-name {
      type leafref {
        path "/te:te/te:lsps/te:lsp/te:tunnel-name";
      }
      description "TE tunnel name.";
    }
    leaf node {
      type leafref {
        path "/te:te/te:lsps/te:lsp/te:node";
      }
      description "The node where the LSP state resides on.";
    }
    leaf lsp-id {
      type leafref {
        path "/te:te/te:lsps/te:lsp/te:lsp-id";
      }
      description "The TE LSP identifier.";
    }
  }
}

/* This grouping will be re-used in path-computation rpc */
```

```
grouping path-computation-response {
  description
    "Attributes reported by path computation response.";
  container computed-paths-properties {
    config false;
    description
      "Computed path properties container.";
    list computed-path-properties {
      key "k-index";
      description
        "List of computed paths.";
      leaf k-index {
        type uint8;
        description
          "The k-th path returned from the computation server.
          A lower k value path is more optimal than higher k
          value path(s)";
      }
      uses path-properties {
        description
          "The TE path computed properties.";
      }
    }
  }
}
container computed-path-error-infos {
  config false;
  description
    "Path computation information container.";
  list computed-path-error-info {
    description
      "List of path computation info entries.";
    leaf error-description {
      type string;
      description
        "Textual representation of the error occurred during
        path computation.";
    }
    leaf error-timestamp {
      type yang:date-and-time;
      description
        "Timestamp of last path computation attempt.";
    }
    leaf error-reason {
      type identityref {
        base path-computation-error-reason;
      }
      description
        "Reason for the path computation error.";
    }
  }
}
```



```
    }
  }
}

grouping lsp-provisioning-error-info {
  description
    "Grouping for LSP provisioning error information.";
  container lsp-provisioning-error-infos {
    config false;
    description
      "LSP provisioning error information.";
    list lsp-provisioning-error-info {
      description
        "List of LSP provisioning error info entries.";
      leaf error-description {
        type string;
        description
          "Textual representation of the error occurred during
          path computation.";
      }
      leaf error-timestamp {
        type yang:date-and-time;
        description
          "Timestamp of when the reported error occurred.";
      }
      leaf error-node-id {
        type te-types:te-node-id;
        default "0.0.0.0";
        description
          "Node identifier of node where error occurred.";
      }
      leaf error-link-id {
        type te-types:te-tp-id;
        default "0";
        description
          "Link ID where the error occurred.";
      }
    }
  }
}

grouping protection-restoration-properties-state {
  description
    "Protection parameters grouping.";
  leaf lockout-of-normal {
    type boolean;
    default "false";
  }
}
```

```
description
  "When set to 'True', it represents a lockout of normal
  traffic external command. When set to 'False', it
  represents a clear lockout of normal traffic external
  command. The lockout of normal traffic command applies
  to this Tunnel.";
reference
  "RFC4427";
}
leaf freeze {
  type boolean;
  default "false";
  description
    "When set to 'True', it represents a freeze external command.
    When set to 'False', it represents a clear freeze external
    command. The freeze command applies to all the Tunnels which
    are sharing the protection resources with this Tunnel.";
  reference
    "RFC4427";
}
leaf lsp-protection-role {
  type enumeration {
    enum working {
      description
        "A working LSP must be a primary LSP whilst a protecting
        LSP can be either a primary or a secondary LSP. Also,
        known as protected LSPs when working LSPs are associated
        with protecting LSPs.";
    }
    enum protecting {
      description
        "A secondary LSP is an LSP that has been provisioned
        in the control plane only; e.g. resource allocation
        has not been committed at the data plane.";
    }
  }
  default "working";
  description
    "LSP role type.";
  reference
    "RFC4872, section 4.2.1";
}
leaf lsp-protection-state {
  type identityref {
    base te-types:lsp-protection-state;
  }
  default "te-types:normal";
  description
```

```
        "The state of the APS state machine controlling which
        tunnels is using the resources of the protecting LSP.";
    }
    leaf protection-group-ingress-node-id {
        type te-types:te-node-id;
        default "0.0.0.0";
        description
            "Indicates the te-node-id of the protection group
            ingress node when the APS state represents an external
            command (LoP, SF, MS) applied to it or a WTR timer
            running on it. If the external command is not applied to
            the ingress node or the WTR timer is not running on it,
            this attribute is not specified. A value 0.0.0.0 is used
            when the te-node-id of the protection group ingress node is
            unknown (e.g., because the ingress node is outside the scope
            of control of the server)";
    }
    leaf protection-group-egress-node-id {
        type te-types:te-node-id;
        default "0.0.0.0";
        description
            "Indicates the te-node-id of the protection group egress node
            when the APS state represents an external command (LoP, SF,
            MS) applied to it or a WTR timer running on it. If the
            external command is not applied to the ingress node or
            the WTR timer is not running on it, this attribute is not
            specified. A value 0.0.0.0 is used when the te-node-id of
            the protection group ingress node is unknown (e.g., because
            the ingress node is outside the scope of control of the
            server)";
    }
}

grouping protection-restoration-properties {
    description
        "Protection and restoration parameters.";
    container protection {
        description
            "Protection parameters.";
        leaf enable {
            type boolean;
            default "false";
            description
                "A flag to specify if LSP protection is enabled.";
            reference
                "RFC4427";
        }
        leaf protection-type {
```

```
    type identityref {
      base te-types:lsp-protection-type;
    }
    default "te-types:lsp-protection-unprotected";
    description
      "LSP protection type.";
  }
  leaf protection-reversion-disable {
    type boolean;
    default "false";
    description
      "Disable protection reversion to working path.";
  }
  leaf hold-off-time {
    type uint32;
    units "milli-seconds";
    default "0";
    description
      "The time between the declaration of an SF or SD condition
       and the initialization of the protection switching
       algorithm.";
    reference
      "RFC4427";
  }
  leaf wait-to-revert {
    type uint16;
    units "seconds";
    description
      "Time to wait before attempting LSP reversion.";
    reference
      "RFC4427";
  }
  leaf aps-signal-id {
    type uint8 {
      range "1..255";
    }
    default "1";
    description
      "The APS signal number used to reference the traffic of
       this tunnel. The default value for normal traffic is 1.
       The default value for extra-traffic is 255. If not
       specified, non-default values can be assigned by the
       server, if and only if, the server controls both
       endpoints.";
    reference
      "RFC4427";
  }
}
```

```
container restoration {
  description
    "Restoration parameters.";
  leaf enable {
    type boolean;
    default "false";
    description
      "A flag to specify if LSP restoration is enabled.";
    reference
      "RFC4427";
  }
  leaf restoration-type {
    type identityref {
      base te-types:lsp-restoration-type;
    }
    default "te-types:lsp-restoration-restore-any";
    description
      "LSP restoration type.";
  }
  leaf restoration-scheme {
    type identityref {
      base te-types:restoration-scheme-type;
    }
    default "te-types:restoration-scheme-preconfigured";
    description
      "LSP restoration scheme.";
  }
  leaf restoration-reversion-disable {
    type boolean;
    default "false";
    description
      "Disable restoration reversion to working path.";
  }
  leaf hold-off-time {
    type uint32;
    units "milli-seconds";
    description
      "The time between the declaration of an SF or SD condition
       and the initialization of the protection switching
       algorithm.";
    reference
      "RFC4427";
  }
  leaf wait-to-restore {
    type uint16;
    units "seconds";
    description
      "Time to wait before attempting LSP restoration.";
```

```
        reference
          "RFC4427";
      }
      leaf wait-to-revert {
        type uint16;
        units "seconds";
        description
          "Time to wait before attempting LSP reversion.";
        reference
          "RFC4427";
      }
    }
  }

  grouping tunnel-associations-properties {
    description
      "TE tunnel association grouping.";
    container association-objects {
      description
        "TE tunnel associations.";
      list association-object {
        key "association-key";
        unique "type id source/id source/type";
        description
          "List of association base objects.";
        reference
          "RFC4872";
        leaf association-key {
          type string;
          description
            "Association key used to identify a specific
              association in the list";
        }
        leaf type {
          type identityref {
            base te-types:association-type;
          }
          description
            "Association type.";
          reference
            "RFC4872";
        }
        leaf id {
          type uint16;
          description
            "Association identifier.";
          reference
            "RFC4872";
        }
      }
    }
  }
}
```

```
    }
    container source {
      uses te-generic-node-id;
      description
        "Association source.";
      reference
        "RFC4872";
    }
  }
  list association-object-extended {
    key "association-key";
    unique
      "type id source/id source/type global-source extended-id";
    description
      "List of extended association objects.";
    reference
      "RFC6780";
    leaf association-key {
      type string;
      description
        "Association key used to identify a specific
        association in the list";
    }
    leaf type {
      type identityref {
        base te-types:association-type;
      }
      description
        "Association type.";
      reference
        "RFC4872, RFC6780";
    }
    leaf id {
      type uint16;
      description
        "Association identifier.";
      reference
        "RFC4872, RFC6780";
    }
  }
  container source {
    uses te-generic-node-id;
    description
      "Association source.";
    reference
      "RFC4872, RFC6780";
  }
  leaf global-source {
    type uint32;
```

```
        description
            "Association global source.";
        reference
            "RFC6780";
    }
    leaf extended-id {
        type yang:hex-string;
        description
            "Association extended identifier.";
        reference
            "RFC6780";
    }
}
}

/* TE tunnel configuration/state grouping */
/* These grouping will be re-used in path-computation rpc */

grouping encoding-and-switching-type {
    description
        "Common grouping to define the LSP encoding and
        switching types";
    leaf encoding {
        type identityref {
            base te-types:lsp-encoding-types;
        }
        description
            "LSP encoding type.";
        reference
            "RFC3945";
    }
    leaf switching-type {
        type identityref {
            base te-types:switching-capabilities;
        }
        description
            "LSP switching type.";
        reference
            "RFC3945";
    }
}

grouping tunnel-common-attributes {
    description
        "Common grouping to define the TE tunnel parameters";
    leaf source {
        type te-types:te-node-id;
```



```
        description
          "TE tunnel source node ID.";
      }
      leaf destination {
        type te-types:te-node-id;
        description
          "TE tunnel destination node identifier.";
      }
      leaf src-tunnel-tp-id {
        type binary;
        description
          "TE tunnel source termination point identifier.";
      }
      leaf dst-tunnel-tp-id {
        type binary;
        description
          "TE tunnel destination termination point identifier.";
      }
      leaf bidirectional {
        type boolean;
        default "false";
        description
          "Indicates a bidirectional co-routed LSP.";
      }
    }
  }

  grouping tunnel-hierarchy-properties {
    description
      "A grouping for TE tunnel hierarchy information.";
    container hierarchy {
      description
        "Container for TE hierarchy related information.";
      container dependency-tunnels {
        description
          "List of tunnels that this tunnel can be potentially
          dependent on.";
        list dependency-tunnel {
          key "name";
          description
            "A tunnel entry that this tunnel can potentially depend
            on.";
          leaf name {
            type leafref {
              path "/te:te/te:tunnels/te:tunnel/te:name";
              require-instance false;
            }
          }
          description
            "Dependency tunnel name. The tunnel may not have been
```

```
        instantiated yet.";
    }
    uses encoding-and-switching-type;
}
}
container hierarchical-link {
    description
        "Identifies a hierarchical link (in client layer)
        that this tunnel is associated with.";
    reference
        "RFC4206";
    leaf local-te-node-id {
        type te-types:te-node-id;
        default "0.0.0.0";
        description
            "The local TE node identifier.";
    }
    leaf local-te-link-tp-id {
        type te-types:te-tp-id;
        default "0";
        description
            "The local TE link termination point identifier.";
    }
    leaf remote-te-node-id {
        type te-types:te-node-id;
        default "0.0.0.0";
        description
            "Remote TE node identifier.";
    }
    uses te-types:te-topology-identifier {
        description
            "The topology identifier where the hierarchical link
            supported by this TE tunnel is instantiated.";
    }
}
}
}

grouping tunnel-properties {
    description
        "Top level grouping for tunnel properties.";
    leaf name {
        type string;
        description
            "TE tunnel name.";
    }
    leaf alias {
        type string;
    }
}
```

```
    description
      "An alternate name of the TE tunnel that can be modified
       anytime during its lifetime.";
  }
  leaf identifier {
    type uint32;
    description
      "TE tunnel Identifier.";
    reference
      "RFC3209";
  }
  leaf color {
    type uint32;
    description "The color associated with the TE tunnel.";
    reference "RFC9012";
  }
  leaf description {
    type string;
    default "None";
    description
      "Textual description for this TE tunnel.";
  }
  leaf admin-state {
    type identityref {
      base te-types:tunnel-admin-state-type;
    }
    default "te-types:tunnel-admin-state-up";
    description
      "TE tunnel administrative state.";
  }
  leaf operational-state {
    type identityref {
      base te-types:tunnel-state-type;
    }
    config false;
    description
      "TE tunnel operational state.";
  }
  uses encoding-and-switching-type;
  uses tunnel-common-attributes;
  container controller {
    description
      "Contains tunnel data relevant to external controller(s).
       This target node may be augmented by external module(s),
       for example, to add data for PCEP initiated and/or
       delegated tunnels.";
    leaf protocol-origin {
      type identityref {
```

```
        base protocol-origin-type;
    }
    description
        "The protocol origin for instantiating the tunnel.";
}
leaf controller-entity-id {
    type string;
    description
        "An identifier unique within the scope of visibility that
        associated with the entity that controls the tunnel";
    reference "RFC8232";
}
}
leaf reoptimize-timer {
    type uint16;
    units "seconds";
    description
        "Frequency of reoptimization of a traffic engineered LSP.";
}
uses tunnel-associations-properties;
uses protection-restoration-properties;
uses te-types:tunnel-constraints;
uses tunnel-hierarchy-properties;
container primary-paths {
    description
        "The set of primary paths.";
    list primary-path {
        key "name";
        description
            "List of primary paths for this tunnel.";
        uses primary-path;
        container primary-reverse-path {
            description
                "The reverse primary path properties.";
            uses primary-reverse-path;
            container candidate-secondary-reverse-paths {
                description
                    "The set of referenced candidate reverse secondary
                    paths from the full set of secondary reverse paths
                    which may be used for this primary path.";
                list candidate-secondary-reverse-path {
                    key "secondary-path";
                    ordered-by user;
                    description
                        "List of candidate secondary reverse path(s)";
                    leaf secondary-path {
                        type leafref {
                            path "../.../.../.../.../..."
                        }
                    }
                }
            }
        }
    }
}
```

```
        + "te:secondary-reverse-paths/"
        + "te:secondary-reverse-path/te:name";
    }
    description
        "A reference to the secondary reverse path that
        should be utilised when the containing primary
        reverse path option is in use.";
    }
}
}
}
container candidate-secondary-paths {
    description
        "The set of candidate secondary paths which may be used
        for this primary path. When secondary paths are
        specified in the list the path of the secondary LSP in
        use must be restricted to those path options referenced.
        The priority of the secondary paths is specified within
        the list. Higher priority values are less preferred -
        that is to say that a path with priority 0 is the most
        preferred path. In the case that the list is empty, any
        secondary path option may be utilised when the current
        primary path is in use.";
    list candidate-secondary-path {
        key "secondary-path";
        ordered-by user;
        description
            "List of candidate secondary paths for this tunnel.";
        leaf secondary-path {
            type leafref {
                path "../../../te:secondary-paths/"
                + "te:secondary-path/te:name";
            }
            description
                "A reference to the secondary path that should be
                utilised when the containing primary path option is
                in use.";
        }
        leaf active {
            type boolean;
            config false;
            description
                "Indicates the current active path option that has
                been selected of the candidate secondary paths.";
        }
    }
}
}
```

```
    }
    container secondary-paths {
      description
        "The set of secondary paths.";
      list secondary-path {
        key "name";
        description
          "List of secondary paths for this tunnel.";
        uses secondary-path;
      }
    }
    container secondary-reverse-paths {
      description
        "The set of secondary reverse paths.";
      list secondary-reverse-path {
        key "name";
        description
          "List of secondary paths for this tunnel.";
        uses secondary-reverse-path;
      }
    }
  }
}

grouping tunnel-actions {
  description
    "Tunnel actions.";
  action tunnel-action {
    description
      "Tunnel action.";
    input {
      leaf action-type {
        type identityref {
          base tunnel-actions-type;
        }
        description
          "Tunnel action type.";
      }
    }
    output {
      leaf action-result {
        type identityref {
          base te-types:te-action-result;
        }
        description
          "The result of the tunnel action operation.";
      }
    }
  }
}
```

```
}

grouping tunnel-protection-actions {
  description
    "Protection external command actions.";
  action protection-external-commands {
    input {
      leaf protection-external-command {
        type identityref {
          base te-types:protection-external-commands;
        }
        description
          "Protection external command.";
      }
      leaf protection-group-ingress-node-id {
        type te-types:te-node-id;
        description
          "When specified, indicates whether the action is
          applied on ingress node.
          By default, if neither ingress nor egress node-id
          is set, the action applies to ingress node only.";
      }
      leaf protection-group-egress-node-id {
        type te-types:te-node-id;
        description
          "When specified, indicates whether the action is
          applied on egress node.
          By default, if neither ingress nor egress node-id
          is set, the action applies to ingress node only.";
      }
      leaf path-ref {
        type path-ref;
        description
          "Indicates to which path the external command applies
          to.";
      }
      leaf traffic-type {
        type enumeration {
          enum normal-traffic {
            description
              "The manual-switch or forced-switch command applies
              to the normal traffic (this Tunnel).";
          }
          enum null-traffic {
            description
              "The manual-switch or forced-switch command applies
              to the null traffic.";
          }
        }
      }
    }
  }
}
```

```
enum extra-traffic {
  description
    "The manual-switch or forced-switch command applies
    to the extra traffic (the extra-traffic Tunnel
    sharing protection bandwidth with this Tunnel).";
}
}
description
  "Indicates whether the manual-switch or forced-switch
  commands applies to the normal traffic, the null traffic
  or the extra-traffic.";
reference
  "RFC4427";
}
leaf extra-traffic-tunnel-ref {
  type tunnel-ref;
  description
    "In case there are multiple extra-traffic tunnels sharing
    protection bandwidth with this Tunnel (m:n protection),
    represents which extra-traffic Tunnel the manual-switch
    or forced-switch to extra-traffic command applies to.";
}
}
}

/** End of TE tunnel groupings */
/**
 * LSP related generic groupings
 */

grouping lsp-record-route-information-state {
  description
    "LSP Recorded route information grouping.";
  container lsp-record-route-information {
    description
      "RSVP recorded route object information.";
    list lsp-record-route-information {
      when "../origin-type = 'ingress'" {
        description
          "Applicable on ingress LSPs only.";
      }
      key "index";
      description
        "Record route list entry.";
      uses te-types:record-route-state;
    }
  }
}
```



```
    }

    grouping lsp-grouping {
      description
        "LSPs state operational data grouping.";
      container lsp {
        config false;
        description
          "TE LSPs state container.";
        list lsp {
          key "tunnel-name lsp-id node";
          unique "source destination tunnel-id lsp-id "
            + "extended-tunnel-id";
          description
            "List of LSPs associated with the tunnel.";
          leaf tunnel-name {
            type string;
            description "The TE tunnel name.";
          }
          leaf lsp-id {
            type uint16;
            description
              "Identifier used in the SENDER_TEMPLATE and the
              FILTER_SPEC that can be changed to allow a sender to
              share resources with itself.";
            reference
              "RFC3209";
          }
          leaf node {
            type te-types:te-node-id;
            description
              "The node where the TE LSP state resides on.";
          }
          uses lsp-properties-state;
          uses lsp-record-route-information-state;
        }
      }
    }

    /*** End of TE LSP groupings ***/
    /**
     * TE global generic groupings
     */
    /* Global named admin-groups configuration data */

    grouping named-admin-groups-properties {
      description
        "Global named administrative groups configuration
```

```
        grouping.";
    leaf name {
        type string;
        description
            "A string name that uniquely identifies a TE
            interface named admin-group.";
    }
    leaf bit-position {
        type uint32;
        description
            "Bit position representing the administrative group.";
        reference
            "RFC3209 and RFC7308";
    }
}

grouping named-admin-groups {
    description
        "Global named administrative groups configuration
        grouping.";
    container named-admin-groups {
        description
            "TE named admin groups container.";
        list named-admin-group {
            if-feature "te-types:extended-admin-groups";
            if-feature "te-types:named-extended-admin-groups";
            key "name";
            description
                "List of named TE admin-groups.";
            uses named-admin-groups-properties;
        }
    }
}

/* Global named admin-srlgs configuration data */

grouping named-srlgs {
    description
        "Global named SRLGs configuration grouping.";
    container named-srlgs {
        description
            "TE named SRLGs container.";
        list named-srlg {
            if-feature "te-types:named-srlg-groups";
            key "name";
            description
                "A list of named SRLG groups.";
            leaf name {
```

```
        type string;
        description
            "A string name that uniquely identifies a TE
             interface named SRLG.";
    }
    leaf value {
        type te-types:srlg;
        description
            "An SRLG value.";
    }
    leaf cost {
        type uint32;
        description
            "SRLG associated cost. Used during path to append
             the path cost when traversing a link with this SRLG.";
    }
}
}
}

/* Global named paths constraints configuration data */

grouping path-constraints-common {
    description
        "Global named path constraints configuration
         grouping.";
    uses te-types:common-path-constraints-attributes {
        description
            "The constraints applicable to the path. This includes:
            - The path bandwidth constraint
            - The path link protection type constraint
            - The path setup/hold priority constraint
            - path signaling type constraint
            - path metric bounds constraint. The unit of path metric
              bound is interpreted in the context of the metric-type.
              For example for metric-type 'path-metric-loss', the bound
              is multiples of the basic unit 0.000003% as described
              in RFC7471 for OSPF, and RFC8570 for ISIS.
            - path affinity constraints
            - path SRLG constraints";
    }
    uses te-types:generic-path-disjointness;
    uses te-types:path-constraints-route-objects;
    container path-in-segment {
        presence "The end-to-end tunnel starts in a previous domain;
                 this tunnel is a segment in the current domain.";
        description
    }
}
```

```
        "If an end-to-end tunnel crosses multiple domains using
        the same technology, some additional constraints have to be
        taken in consideration in each domain.
        This TE tunnel segment is stitched to the upstream TE tunnel
        segment.";
    uses te-types:label-set-info;
}
container path-out-segment {
    presence
        "The end-to-end tunnel is not terminated in this domain;
        this tunnel is a segment in the current domain.";
    description
        "If an end-to-end tunnel crosses multiple domains using
        the same technology, some additional constraints have to be
        taken in consideration in each domain.
        This TE tunnel segment is stitched to the downstream TE
        tunnel segment.";
    uses te-types:label-set-info;
}
}

grouping named-path-constraints {
    description
        "Global named path constraints configuration
        grouping.";
    container named-path-constraints {
        description
            "TE named path constraints container.";
        list named-path-constraint {
            if-feature "te-types:named-path-constraints";
            key "name";
            leaf name {
                type string;
                description
                    "A string name that uniquely identifies a
                    path constraint set.";
            }
            uses path-constraints-common;
            description
                "A list of named path constraints.";
        }
    }
}

/* TE globals container data */

grouping globals-grouping {
    description
```

```
    "Globals TE system-wide configuration data grouping.";
  container globals {
    description
      "Globals TE system-wide configuration data container.";
    uses named-admin-groups;
    uses named-srlgs;
    uses named-path-constraints;
  }
}

/* TE tunnels container data */

grouping tunnels-grouping {
  description
    "Tunnels TE configuration data grouping.";
  container tunnels {
    description
      "Tunnels TE configuration data container.";
    list tunnel {
      key "name";
      description
        "The list of TE tunnels.";
      uses tunnel-properties;
      uses tunnel-actions;
      uses tunnel-protection-actions;
    }
  }
}

/* TE LSPs ephemeral state container data */

grouping lsp-properties-state {
  description
    "LSPs state operational data grouping.";
  leaf source {
    type te-types:te-node-id;
    description
      "Tunnel sender address extracted from
       SENDER_TEMPLATE object.";
    reference
      "RFC3209";
  }
  leaf destination {
    type te-types:te-node-id;
    description
      "The tunnel endpoint address extracted from SESSION object.";
    reference
      "RFC3209";
  }
}
```

```
    }
    leaf tunnel-id {
      type uint16;
      description
        "The tunnel identifier used in the SESSION that remains
        constant over the life of the tunnel.";
      reference
        "RFC3209";
    }
    leaf extended-tunnel-id {
      type yang:dotted-quad;
      description
        "The LSP Extended Tunnel ID.";
      reference
        "RFC3209";
    }
    leaf operational-state {
      type identityref {
        base te-types:lsp-state-type;
      }
      description
        "The LSP operational state.";
    }
    leaf signaling-type {
      type identityref {
        base te-types:path-signaling-type;
      }
      description
        "The signaling protocol used to set up this LSP.";
    }
    leaf origin-type {
      type enumeration {
        enum ingress {
          description
            "Origin ingress.";
        }
        enum egress {
          description
            "Origin egress.";
        }
        enum transit {
          description
            "Origin transit.";
        }
      }
      default "ingress";
      description
        "The origin of the LSP relative to the location of the local
```

```
        switch in the path.";
    }
    leaf lsp-resource-status {
        type enumeration {
            enum primary {
                description
                "A primary LSP is a fully established LSP for which the
                resource allocation has been committed at the data
                plane.";
            }
            enum secondary {
                description
                "A secondary LSP is an LSP that has been provisioned
                in the control plane only; e.g. resource allocation
                has not been committed at the data plane.";
            }
        }
        default "primary";
        description
        "LSP resource allocation state.";
        reference
        "RFC4872, section 4.2.1";
    }
    uses protection-restoration-properties-state;
}

/** End of TE global groupings */
/**
 * TE container
 */

container te {
    presence "Enable TE feature.";
    description
    "TE global container.";
    /* TE Global Data */
    uses globals-grouping;

    /* TE Tunnel Data */
    uses tunnels-grouping;

    /* TE LSPs Data */
    uses lsps-grouping;
}

/* TE Tunnel RPCs/execution Data */

rpc tunnels-path-compute {
```

```
description
  "TE tunnels RPC nodes.";
input {
  container path-compute-info {
    /*
     * An external path compute module may augment this
     * target.
     */
    description
      "RPC input information.";
  }
}
output {
  container path-compute-result {
    /*
     * An external path compute module may augment this
     * target.
     */
    description
      "RPC output information.";
  }
}

rpc tunnels-actions {
  description
    "TE tunnels actions RPC";
  input {
    container tunnel-info {
      description
        "TE tunnel information.";
      choice filter-type {
        mandatory true;
        description
          "Filter choice.";
        case all-tunnels {
          leaf all {
            type empty;
            mandatory true;
            description
              "Apply action on all TE tunnels.";
          }
        }
        case one-tunnel {
          leaf tunnel {
            type tunnel-ref;
            description
              "Apply action on the specific TE tunnel.";
          }
        }
      }
    }
  }
}
```



```

    }
  }
}
container action-info {
  description
    "TE tunnel action information.";
  leaf action {
    type identityref {
      base tunnel-actions-type;
    }
    description
      "The action type.";
  }
  leaf disruptive {
    when "derived-from-or-self(..../action, "
      + "'te:tunnel-action-reoptimize')";
    type empty;
    description
      "Specifies whether or not the reoptimization action
        is allowed to be disruptive.";
  }
}
}
output {
  leaf action-result {
    type identityref {
      base te-types:te-action-result;
    }
    description
      "The result of the tunnel action operation.";
  }
}
}
}
<CODE ENDS>

```

Figure 9: TE Tunnel data model YANG module

## 6. TE Device YANG Model

The device TE YANG module ('ietf-te-device') models data that is specific to managing a TE device. This module augments the generic TE YANG module.

## 6.1. Module Structure

### 6.1.1. TE Interfaces

This branch of the model manages TE interfaces that are present on a device. Examples of TE interface properties are:

- \* Maximum reservable bandwidth, bandwidth constraints (BC)
- \* Flooding parameters
  - Flooding intervals and threshold values
- \* interface attributes
  - (Extended) administrative groups
  - SRLG values
  - TE metric value
- \* Fast reroute backup tunnel properties (such as static, auto-tunnel)

The derived state associated with interfaces is grouped under the interface "state" sub-container as shown in Figure 10. This covers state data such as:

- \* Bandwidth information: maximum bandwidth, available bandwidth at different priorities and for each class-type (CT)
- \* List of admitted LSPs
  - Name, bandwidth value and pool, time, priority
- \* Statistics: state counters, flooding counters, admission counters (accepted/rejected), preemption counters
- \* Adjacency information
  - Neighbor address
  - Metric value

```

module: ietf-te-device
  augment /te:te:
    +--rw interfaces
    .
    +-- rw te-dev:te-attributes
       <<intended configuration>>
    .
    +-- ro state
       <<derived state associated with the TE interface>>

```

Figure 10: TE interface state YANG subtree

## 6.2. Tree Diagram

Figure 11 shows the tree diagram of the device TE YANG model defined in modules 'ietf-te.yang'.

```

module: ietf-te-device
  augment /te:te:
    +--rw interfaces
    |
    |   +--rw threshold-type?          enumeration
    |   +--rw delta-percentage?       rt-types:percentage
    |   +--rw threshold-specification? enumeration
    |   +--rw up-thresholds*          rt-types:percentage
    |   +--rw down-thresholds*        rt-types:percentage
    |   +--rw up-down-thresholds*     rt-types:percentage
    |   +--rw interface* [interface]
    |       +--rw interface                if:interface-ref
    |       +--rw te-metric?
    |           |
    |           |   te-types:te-metric
    |       +--rw (admin-group-type)?
    |           |
    |           |   +--:(value-admin-groups)
    |           |   |
    |           |   |   +--rw (value-admin-group-type)?
    |           |   |   |
    |           |   |   |   +--:(admin-groups)
    |           |   |   |   |
    |           |   |   |   |   +--rw admin-group?
    |           |   |   |   |   |
    |           |   |   |   |   |   te-types:admin-group
    |           |   |   |   |   +--:(extended-admin-groups)
    |           |   |   |   |   |   {te-types:extended-admin-groups}?
    |           |   |   |   |   |   +--rw extended-admin-group?
    |           |   |   |   |   |   |
    |           |   |   |   |   |   |   te-types:extended-admin-group
    |           |   |   |   |   +--:(named-admin-groups)
    |           |   |   |   |   |   +--rw named-admin-groups* [named-admin-group]
    |           |   |   |   |   |   |
    |           |   |   |   |   |   |   {te-types:extended-admin-groups, te-types:named-
    |           |   |   |   |   |   |   extended-admin-groups}?
    |           |   |   |   |   |   |   +--rw named-admin-group    leafref
    |           |   |   |   |   +--rw (srlg-type)?
    |           |   |   |   |   |
    |           |   |   |   |   |   +--:(value-srlgs)
    |           |   |   |   |   |   |
    |           |   |   |   |   |   |   +--rw values* [value]

```

```

    |         +---rw value      uint32
    +---:(named-srlgs)
    |         +---rw named-srlgs* [named-srlg]
    |         |         {te-types:named-srlg-groups}?
    |         +---rw named-srlg      leafref
+---rw threshold-type?          enumeration
+---rw delta-percentage?
|         rt-types:percentage
+---rw threshold-specification?  enumeration
+---rw up-thresholds*
|         rt-types:percentage
+---rw down-thresholds*
|         rt-types:percentage
+---rw up-down-thresholds*
|         rt-types:percentage
+---rw switching-capabilities* [switching-capability]
|         +---rw switching-capability      identityref
|         +---rw encoding?                  identityref
+---ro state
    +---ro te-advertisements-state
    |         +---ro flood-interval?          uint32
    |         +---ro last-flooded-time?       uint32
    |         +---ro next-flooded-time?       uint32
    |         +---ro last-flooded-trigger?    enumeration
    |         +---ro advertised-level-areas* [level-area]
    |         +---ro level-area      uint32
+---rw performance-thresholds
augment /te:te/te:globals:
+---rw lsp-install-interval?      uint32
+---rw lsp-cleanup-interval?      uint32
+---rw lsp-invalidation-interval? uint32
augment /te:te/te:tunnels/te:tunnel:
+---rw path-invalidation-action?  identityref
+---rw lsp-install-interval?      uint32
+---rw lsp-cleanup-interval?      uint32
+---rw lsp-invalidation-interval? uint32
augment /te:te/te:lsps/te:lsp:
+---ro lsp-timers
|         +---ro life-time?      uint32
|         +---ro time-to-install? uint32
|         +---ro time-to-destroy? uint32
+---ro downstream-info
|         +---ro nhop?            te-types:te-tp-id
|         +---ro outgoing-interface? if:interface-ref
|         +---ro neighbor
|         |         +---ro id?      te-gen-node-id
|         |         +---ro type?    enumeration
+---ro label?          rt-types:generalized-label

```

```

+--ro upstream-info
  +--ro phop?      te-types:te-tp-id
  +--ro neighbor
    | +--ro id?    te-gen-node-id
    | +--ro type?  enumeration
  +--ro label?     rt-types:generalized-label

rpcs:
  +---x link-state-update
    +---w input
      +---w (filter-type)
        +---:(match-all)
          | +---w all          empty
        +---:(match-one-interface)
          +---w interface?    if:interface-ref

```

Figure 11: TE Tunnel device model YANG tree diagram

### 6.3. YANG Module

The device TE YANG module 'ietf-te-device' imports the following module(s):

- \* ietf-yang-types and ietf-inet-types defined in [RFC6991]
- \* ietf-interfaces defined in [RFC8343]
- \* ietf-routing-types defined in [RFC8294]
- \* ietf-te-types defined in [RFC8776]
- \* ietf-te defined in this document

```

<CODE BEGINS> file "ietf-te-device@2021-10-22.yang"
module ietf-te-device {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-te-device";

  /* Replace with IANA when assigned */

  prefix te-dev;

  /* Import TE module */

  import ietf-te {
    prefix te;
    reference
      "draft-ietf-teas-yang-te: A YANG Data Model for Traffic

```

```
    Engineering Tunnels and Interfaces";
}

/* Import TE types */

import ietf-te-types {
    prefix te-types;
    reference
        "RFC8776: Common YANG Data Types for Traffic Engineering.";
}
import ietf-interfaces {
    prefix if;
    reference
        "RFC8343: A YANG Data Model for Interface Management";
}
import ietf-routing-types {
    prefix rt-types;
    reference
        "RFC8294: Common YANG Data Types for the Routing Area";
}

organization
    "IETF Traffic Engineering Architecture and Signaling (TEAS)
    Working Group";
contact
    "WG Web:    <http://tools.ietf.org/wg/teas/>
    WG List:    <mailto:teas@ietf.org>

    Editor:     Tarek Saad
                <mailto:tsaad@juniper.net>

    Editor:     Rakesh Gandhi
                <mailto:rgandhi@cisco.com>

    Editor:     Vishnu Pavan Beeram
                <mailto:vbeeram@juniper.net>

    Editor:     Himanshu Shah
                <mailto:hshah@ciena.com>

    Editor:     Xufeng Liu
                <mailto:xufeng.liu.ietf@gmail.com>

    Editor:     Igor Bryskin
                <mailto:i_bryskin@yahoo.com>";
description
    "YANG data module for TE device configurations,
    state, and RPCs. The model fully conforms to the
```

Network Management Datastore Architecture (NMDA).

Copyright (c) 2019 IETF Trust and the persons  
identified as authors of the code. All rights reserved.

Redistribution and use in source and binary forms, with or  
without modification, is permitted pursuant to, and subject  
to the license terms contained in, the Simplified BSD License  
set forth in Section 4.c of the IETF Trust's Legal Provisions  
Relating to IETF Documents  
(<https://trustee.ietf.org/license-info>).  
This version of this YANG module is part of RFC XXXX; see  
the RFC itself for full legal notices.";

```
// RFC Ed.: replace XXXX with actual RFC number and remove this
// note.
// RFC Ed.: update the date below with the date of RFC publication
// and remove this note.
```

```
revision 2021-10-22 {
  description
    "Latest update to TE device YANG module.";
  reference
    "RFCXXXX: A YANG Data Model for Traffic Engineering Tunnels
    and Interfaces";
}
```

```
/**
 * TE LSP device state grouping
 */
```

```
grouping lsps-device-info {
  description
    "TE LSP device state grouping.";
  container lsp-timers {
    when "../te:origin-type = 'ingress'" {
      description
        "Applicable to ingress LSPs only.";
    }
    description
      "Ingress LSP timers.";
    leaf life-time {
      type uint32;
      units "seconds";
      description
        "TE LSP lifetime.";
    }
    leaf time-to-install {
```

```
        type uint32;
        units "seconds";
        description
            "TE LSP installation delay time.";
    }
    leaf time-to-destroy {
        type uint32;
        units "seconds";
        description
            "TE LSP expiration delay time.";
    }
}
container downstream-info {
    when "../te:origin-type != 'egress'" {
        description
            "Downstream information of the LSP.";
    }
    description
        "downstream information.";
    leaf nhop {
        type te-types:te-tp-id;
        description
            "downstream next-hop address.";
    }
    leaf outgoing-interface {
        type if:interface-ref;
        description
            "downstream interface.";
    }
    container neighbor {
        uses te:te-generic-node-id;
        description
            "downstream neighbor address.";
    }
    leaf label {
        type rt-types:generalized-label;
        description
            "downstream label.";
    }
}
container upstream-info {
    when "../te:origin-type != 'ingress'" {
        description
            "Upstream information of the LSP.";
    }
    description
        "upstream information.";
    leaf phop {
```



```
        type te-types:te-tp-id;
        description
            "upstream next-hop or previous-hop address.";
    }
    container neighbor {
        uses te:te-generic-node-id;
        description
            "upstream neighbor address.";
    }
    leaf label {
        type rt-types:generalized-label;
        description
            "upstream label.";
    }
}

/**
 * Device general groupings.
 */

grouping lsp-device-timers {
    description
        "Device TE LSP timers configs.";
    leaf lsp-install-interval {
        type uint32;
        units "seconds";
        description
            "TE LSP installation delay time.";
    }
    leaf lsp-cleanup-interval {
        type uint32;
        units "seconds";
        description
            "TE LSP cleanup delay time.";
    }
    leaf lsp-invalidation-interval {
        type uint32;
        units "seconds";
        description
            "TE LSP path invalidation before taking action delay time.";
    }
}

/**
 * TE global device groupings
 */
/* TE interface container data */
```

```
grouping interfaces-grouping {
  description
    "TE interface configuration data grouping.";
  container interfaces {
    description
      "Configuration data model for TE interfaces.";
    uses te-all-attributes;
    list interface {
      key "interface";
      description
        "TE interfaces.";
      leaf interface {
        type if:interface-ref;
        description
          "TE interface name.";
      }
      /* TE interface parameters */
      uses te-attributes;
    }
  }
}

/**
 * TE interface device groupings
 */

grouping te-admin-groups-config {
  description
    "TE interface affinities grouping.";
  choice admin-group-type {
    description
      "TE interface administrative groups
      representation type.";
    case value-admin-groups {
      choice value-admin-group-type {
        description
          "choice of admin-groups.";
        case admin-groups {
          description
            "Administrative group/Resource
            class/Color.";
          leaf admin-group {
            type te-types:admin-group;
            description
              "TE interface administrative group.";
          }
        }
      }
    case extended-admin-groups {
```

```
        if-feature "te-types:extended-admin-groups";
        description
            "Extended administrative group/Resource
             class/Color.";
        leaf extended-admin-group {
            type te-types:extended-admin-group;
            description
                "TE interface extended administrative group.";
        }
    }
}
}
case named-admin-groups {
    list named-admin-groups {
        if-feature "te-types:extended-admin-groups";
        if-feature "te-types:named-extended-admin-groups";
        key "named-admin-group";
        description
            "A list of named admin-group entries.";
        leaf named-admin-group {
            type leafref {
                path "../..../te:globals/"
                    + "te:named-admin-groups/te:named-admin-group/"
                    + "te:name";
            }
            description
                "A named admin-group entry.";
        }
    }
}
}
}

/* TE interface SRLGs */

grouping te-srlgs-config {
    description
        "TE interface SRLG grouping.";
    choice srlg-type {
        description
            "Choice of SRLG configuration.";
        case value-srlgs {
            list values {
                key "value";
                description
                    "List of SRLG values that
                     this link is part of.";
                leaf value {
```

```
        type uint32 {
            range "0..4294967295";
        }
        description
            "Value of the SRLG";
    }
}
}
case named-srlgs {
    list named-srlgs {
        if-feature "te-types:named-srlg-groups";
        key "named-srlg";
        description
            "A list of named SRLG entries.";
        leaf named-srlg {
            type leafref {
                path "../..../te:globals/"
                    + "te:named-srlgs/te:named-srlg/te:name";
            }
            description
                "A named SRLG entry.";
        }
    }
}
}
}
}

grouping te-igp-flooding-bandwidth-config {
    description
        "Configurable items for igp flooding bandwidth
        threshold configuration.";
    leaf threshold-type {
        type enumeration {
            enum delta {
                description
                    "'delta' indicates that the local
                    system should flood IGP updates when a
                    change in reserved bandwidth >= the specified
                    delta occurs on the interface.";
            }
            enum threshold-crossed {
                description
                    "THRESHOLD-CROSSED indicates that
                    the local system should trigger an update (and
                    hence flood) the reserved bandwidth when the
                    reserved bandwidth changes such that it crosses,
                    or becomes equal to one of the threshold values.";
            }
        }
    }
}
```

```
}
description
  "The type of threshold that should be used to specify the
  values at which bandwidth is flooded. 'delta' indicates that
  the local system should flood IGP updates when a change in
  reserved bandwidth >= the specified delta occurs on the
  interface. Where 'threshold-crossed' is specified, the local
  system should trigger an update (and hence flood) the
  reserved bandwidth when the reserved bandwidth changes such
  that it crosses, or becomes equal to one of the threshold
  values."
}
leaf delta-percentage {
  when "../threshold-type = 'delta'" {
    description
      "The percentage delta can only be specified when the
      threshold type is specified to be a percentage delta of
      the reserved bandwidth."
  }
  type rt-types:percentage;
  description
    "The percentage of the maximum-reservable-bandwidth
    considered as the delta that results in an IGP update
    being flooded."
}
leaf threshold-specification {
  when "../threshold-type = 'threshold-crossed'" {
    description
      "The selection of whether mirrored or separate threshold
      values are to be used requires user specified thresholds
      to be set."
  }
  type enumeration {
    enum mirrored-up-down {
      description
        "mirrored-up-down indicates that a single set of
        threshold values should be used for both increasing
        and decreasing bandwidth when determining whether
        to trigger updated bandwidth values to be flooded
        in the IGP TE extensions."
    }
    enum separate-up-down {
      description
        "separate-up-down indicates that a separate
        threshold values should be used for the increasing
        and decreasing bandwidth when determining whether
        to trigger updated bandwidth values to be flooded
        in the IGP TE extensions."
    }
  }
}
```

```
    }
  }
  description
    "This value specifies whether a single set of threshold
    values should be used for both increasing and decreasing
    bandwidth when determining whether to trigger updated
    bandwidth values to be flooded in the IGP TE extensions.
    'mirrored-up-down' indicates that a single value (or set of
    values) should be used for both increasing and decreasing
    values, where 'separate-up-down' specifies that the
    increasing and decreasing values will be separately
    specified.";
}
leaf-list up-thresholds {
  when "../threshold-type = 'threshold-crossed'"
    + "and ../threshold-specification = 'separate-up-down'" {
    description
      "A list of up-thresholds can only be specified when the
      bandwidth update is triggered based on crossing a
      threshold and separate up and down thresholds are
      required.";
  }
  type rt-types:percentage;
  description
    "The thresholds (expressed as a percentage of the maximum
    reservable bandwidth) at which bandwidth updates are to be
    triggered when the bandwidth is increasing.";
}
leaf-list down-thresholds {
  when "../threshold-type = 'threshold-crossed'"
    + "and ../threshold-specification = 'separate-up-down'" {
    description
      "A list of down-thresholds can only be specified when the
      bandwidth update is triggered based on crossing a
      threshold and separate up and down thresholds are
      required.";
  }
  type rt-types:percentage;
  description
    "The thresholds (expressed as a percentage of the maximum
    reservable bandwidth) at which bandwidth updates are to be
    triggered when the bandwidth is decreasing.";
}
leaf-list up-down-thresholds {
  when "../threshold-type = 'threshold-crossed'"
    + "and ../threshold-specification = 'mirrored-up-down'" {
    description
      "A list of thresholds corresponding to both increasing
```

```
        and decreasing bandwidths can be specified only when an
        update is triggered based on crossing a threshold, and
        the same up and down thresholds are required.";
    }
    type rt-types:percentage;
    description
        "The thresholds (expressed as a percentage of the maximum
        reservable bandwidth of the interface) at which bandwidth
        updates are flooded - used both when the bandwidth is
        increasing and decreasing.";
}
}

/* TE interface metric */

grouping te-metric-config {
    description
        "TE interface metric grouping.";
    leaf te-metric {
        type te-types:te-metric;
        description
            "TE interface metric.";
    }
}

/* TE interface switching capabilities */

grouping te-switching-cap-config {
    description
        "TE interface switching capabilities.";
    list switching-capabilities {
        key "switching-capability";
        description
            "List of interface capabilities for this interface.";
        leaf switching-capability {
            type identityref {
                base te-types:switching-capabilities;
            }
            description
                "Switching Capability for this interface.";
        }
        leaf encoding {
            type identityref {
                base te-types:lsp-encoding-types;
            }
            description
                "Encoding supported by this interface.";
        }
    }
}
```

```
    }  
  }  
  
  grouping te-advertisements-state {  
    description  
      "TE interface advertisements state grouping.";  
    container te-advertisements-state {  
      description  
        "TE interface advertisements state container.";  
      leaf flood-interval {  
        type uint32;  
        description  
          "The periodic flooding interval.";  
      }  
      leaf last-flooded-time {  
        type uint32;  
        units "seconds";  
        description  
          "Time elapsed since last flooding in seconds.";  
      }  
      leaf next-flooded-time {  
        type uint32;  
        units "seconds";  
        description  
          "Time remained for next flooding in seconds.";  
      }  
      leaf last-flooded-trigger {  
        type enumeration {  
          enum link-up {  
            description  
              "Link-up flooding trigger.";  
          }  
          enum link-down {  
            description  
              "Link-down flooding trigger.";  
          }  
          enum threshold-up {  
            description  
              "Bandwidth reservation up threshold.";  
          }  
          enum threshold-down {  
            description  
              "Bandwidth reservation down threshold.";  
          }  
          enum bandwidth-change {  
            description  
              "Bandwidth capacity change.";  
          }  
        }  
      }  
    }  
  }
```



```
        enum user-initiated {
            description
                "Initiated by user.";
        }
        enum srlg-change {
            description
                "SRLG property change.";
        }
        enum periodic-timer {
            description
                "Periodic timer expired.";
        }
    }
    default "periodic-timer";
    description
        "Trigger for the last flood.";
}
list advertised-level-areas {
    key "level-area";
    description
        "List of level-areas that the TE interface is advertised
        in.";
    leaf level-area {
        type uint32;
        description
            "The IGP area or level where the TE interface link state
            is advertised in.";
    }
}
}
}

/* TE interface attributes grouping */

grouping te-attributes {
    description
        "TE attributes configuration grouping.";
    uses te-metric-config;
    uses te-admin-groups-config;
    uses te-srlgs-config;
    uses te-igp-flooding-bandwidth-config;
    uses te-switching-cap-config;
    container state {
        config false;
        description
            "State parameters for interface TE metric.";
        uses te-advertisements-state;
    }
}
```

```
}

grouping te-all-attributes {
  description
    "TE attributes configuration grouping for all
    interfaces.";
  uses te-igp-flooding-bandwidth-config;
}

/** End of TE interfaces device groupings */
/**
 * TE device augmentations
 */

augment "/te:te" {
  description
    "TE global container.";
  /* TE Interface Configuration Data */
  uses interfaces-grouping;
  container performance-thresholds {
    description
      "Performance parameters configurable thresholds.";
  }
}

/* TE globals device augmentation */

augment "/te:te/te:globals" {
  description
    "Global TE device specific configuration parameters.";
  uses lsp-device-timers;
}

/* TE tunnels device configuration augmentation */

augment "/te:te/te:tunnels/te:tunnel" {
  description
    "Tunnel device dependent augmentation.";
  leaf path-invalidation-action {
    type identityref {
      base te-types:path-invalidation-action-type;
    }
    description
      "Tunnel path invalidation action.";
  }
  uses lsp-device-timers;
}
```

```

/* TE LSPs device state augmentation */

augment "/te:te/te:lsps/te:lsp" {
  description
    "TE LSP device dependent augmentation.";
  uses lsp-device-info;
}

/* TE interfaces RPCs/execution Data */

rpc link-state-update {
  description
    "Triggers a link state update for the specific interface.";
  input {
    choice filter-type {
      mandatory true;
      description
        "Filter choice.";
      case match-all {
        leaf all {
          type empty;
          mandatory true;
          description
            "Match all TE interfaces.";
        }
      }
      case match-one-interface {
        leaf interface {
          type if:interface-ref;
          description
            "Match a specific TE interface.";
        }
      }
    }
  }
}
}

<CODE ENDS>

```

Figure 12: TE device data model YANG module

## 7. Notifications

Notifications are a key component of any topology data model.

[RFC8639] and [RFC8641] define a subscription mechanism and a push mechanism for YANG datastores. These mechanisms currently allow the user to:

- \* Subscribe to notifications on a per-client basis.
- \* Specify subtree filters or XML Path Language (XPath) filters so that only contents of interest will be sent.
- \* Specify either periodic or on-demand notifications.

## 8. TE Generic and Helper YANG Modules

## 9. IANA Considerations

This document registers the following URIs in the IETF XML registry [RFC3688]. Following the format in [RFC3688], the following registrations are requested to be made.

URI: urn:ietf:params:xml:ns:yang:ietf-te  
Registrant Contact: The IESG.  
XML: N/A, the requested URI is an XML namespace.

URI: urn:ietf:params:xml:ns:yang:ietf-te-device  
Registrant Contact: The IESG.  
XML: N/A, the requested URI is an XML namespace.

This document registers two YANG modules in the YANG Module Names registry [RFC6020].

Name: ietf-te  
Namespace: urn:ietf:params:xml:ns:yang:ietf-te  
Prefix: te  
Reference: RFCXXXX

Name: ietf-te-device  
Namespace: urn:ietf:params:xml:ns:yang:ietf-te-device  
Prefix: te-device  
Reference: RFCXXXX

## 10. Security Considerations

The YANG module specified in this document defines a schema for data that is designed to be accessed via network management protocols such as NETCONF [RFC6241] or RESTCONF [RFC8040]. The lowest NETCONF layer is the secure transport layer, and the mandatory-to-implement secure transport is Secure Shell (SSH) [RFC6242]. The lowest RESTCONF layer is HTTPS, and the mandatory-to-implement secure transport is TLS [RFC8446].

The Network Configuration Access Control Model (NACM) [RFC8341] provides the means to restrict access for particular NETCONF or RESTCONF users to a preconfigured subset of all available NETCONF or RESTCONF protocol operations and content.

There are a number of data nodes defined in this YANG module that are writable/creatable/deletable (i.e., config true, which is the default). These data nodes may be considered sensitive or vulnerable in some network environments. Write operations (e.g., edit-config) to these data nodes without proper protection can have a negative effect on network operations. These are the subtrees and data nodes and their sensitivity/vulnerability:

`"/te/globals"`: This module specifies the global TE configurations on a device. Unauthorized access to this container could cause the device to ignore packets it should receive and process.

`"/te/tunnels"`: This list specifies the configuration and state of TE Tunnels present on the device or controller. Unauthorized access to this list could cause the device to ignore packets it should receive and process. An attacker may also use state to derive information about the network topology, and subsequently orchestrate further attacks.

`"/te/interfaces"`: This list specifies the configuration and state TE interfaces on a device. Unauthorized access to this list could cause the device to ignore packets it should receive and process.

Some of the readable data nodes in this YANG module may be considered sensitive or vulnerable in some network environments. It is thus important to control read access (e.g., via get, get-config, or notification) to these data nodes. These are the subtrees and data nodes and their sensitivity/vulnerability:

`"/te/lspss"`: this list contains information state about established LSPs in the network. An attacker can use this information to derive information about the network topology, and subsequently orchestrate further attacks.

Some of the RPC operations in this YANG module may be considered sensitive or vulnerable in some network environments. It is thus important to control access to these operations. These are the operations and their sensitivity/vulnerability:

`"/te/tunnels-actions"`: using this RPC, an attacker can modify existing paths that may be carrying live traffic, and hence result to interruption to services carried over the network.

`"/te/tunnels-path-compute"`: using this RPC, an attacker can retrieve secured information about the network provider which can be used to orchestrate further attacks.

The security considerations spelled out in the YANG 1.1 specification [RFC7950] apply for this document as well.

## 11. Acknowledgement

The authors would like to thank the members of the multi-vendor YANG design team who are involved in the definition of this model.

The authors would like to thank Tom Petch for reviewing and providing useful feedback about the document. The authors would also like to thank Loa Andersson, Lou Berger, Sergio Belotti, Italo Busi, Carlo Perocchio, Francesco Lazzeri, Aihua Guo, Dhruv Dhody, and Raqib Jones for providing useful feedback on this document.

## 12. Contributors

Himanshu Shah  
Ciena

Email: hshah@ciena.com

Xia Chen  
Huawei Technologies

Email: jescia.chenxia@huawei.com

Bin Wen  
Comcast

Email: Bin\_Wen@cable.comcast.com

## 13. Appendix A: Data Tree Examples

This section contains examples of use of the model with RESTCONF [RFC8040] and JSON encoding.

For the example we will use a 4 node MPLS network where RSVP-TE MPLS Tunnels can be setup. The loopbacks of each router are shown. The network in Figure 13 will be used in the examples described in the following sections.

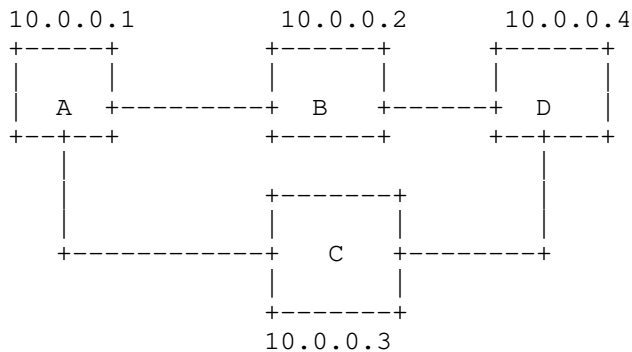


Figure 13: TE network used in data tree examples

### 13.1. Basic Tunnel Setup

This example uses the TE Tunnel YANG data model defined in this document to create an RSVP-TE signaled Tunnel of packet LSP encoding type. First, the TE Tunnel is created with no specific restrictions or constraints (e.g., protection or restoration). The TE Tunnel ingresses on router A and egresses on router D.

In this case, the TE Tunnel is created without specifying additional information about the primary paths.

```

POST /restconf/data/ietf-te:te/tunnels HTTP/1.1
Host: example.com
Accept: application/yang-data+json
Content-Type: application/yang-data+json

{
  "ietf-te:tunnel": [
    {
      "name": "Example_LSP_Tunnel_A_2",
      "encoding": "te-types:lsp-encoding-packet",
      "admin-state": "te-types:tunnel-state-up",
      "source": "10.0.0.1",
      "destination": "10.0.0.4",
      "bidirectional": "false",
      "signaling-type": "te-types:path-setup-rsvp"
    }
  ]
}

```

### 13.2. Global Named Path Constraints

This example uses the YANG data model to create a 'named path constraint' that can be reference by TE Tunnels. The path constraint, in this case, limits the TE Tunnel hops for the computed path.

```
POST /restconf/data/ietf-te:te/globals/named-path-constraints HTTP/1.1
```

```
Host: example.com
```

```
Accept: application/yang-data+json
```

```
Content-Type: application/yang-data+json
```

```
{
  "ietf-te:named-path-constraint": {
    "name": "max-hop-3",
    "path-metric-bounds": {
      "path-metric-bound": {
        "metric-type": "te-types:path-metric-hop",
        "upper-bound": "3"
      }
    }
  }
}
```

### 13.3. Tunnel with Global Path Constraint

In this example, the previously created 'named path constraint' is applied to the TE Tunnel created in Section 13.1.



```
POST /restconf/data/ietf-te:te/tunnels HTTP/1.1
Host: example.com
Accept: application/yang-data+json
Content-Type: application/yang-data+json

{
  "ietf-te:ietf-tunnel": [
    {
      "name": "Example_LSP_Tunnel_A_4_1",
      "encoding": "te-types:lsp-encoding-packet",
      "description": "Simple_LSP_with_named_path",
      "admin-state": "te-types:tunnel-state-up",
      "source": "10.0.0.1",
      "destination": "10.0.0.4",
      "signaling-type": "path-setup-rsvp",
      "bidirectional": "false",
      "primary-paths": [
        {
          "primary-path": {
            "name": "Simple_LSP_1",
            "use-path-computation": "true",
            "named-path-constraint": "max-hop-3"
          }
        }
      ]
    }
  ]
}
```

#### 13.4. Tunnel with Per-tunnel Path Constraint

In this example, the a per tunnel path constraint is explicitly indicated under the TE Tunnel created in Section 13.1 to constrain the computed path for the tunnel.

```
POST /restconf/data/ietf-te:te/tunnels HTTP/1.1
Host: example.com
Accept: application/yang-data+json
Content-Type: application/yang-data+json

{
  "ietf-te:tunnel": [
    {
      "name": "Example_LSP_Tunnel_A_4_2",
      "encoding": "te-types:lsp-encoding-packet",
      "admin-state": "te-types:tunnel-state-up",
      "source": "10.0.0.1",
      "destination": "10.0.0.4",
      "bidirectional": "false",
      "signaling-type": "te-types:path-setup-rsvp",
      "primary-paths": {
        "primary-path": [
          {
            "name": "path1",
            "path-metric-bounds": {
              "path-metric-bound": [
                {
                  "metric-type": "te-types:path-metric-hop",
                  "upper-bound": "3"
                }
              ]
            }
          }
        ]
      }
    }
  ]
}
```

### 13.5. Tunnel State

In this example, the 'GET' query is sent to return the state stored about the tunnel.

```
GET /restconf/data/ietf-te:te/tunnels/tunnel="Example_LSP_Tunnel_A_4_1"
/p2p-primary-paths/ HTTP/1.1
Host: example.com
Accept: application/yang-data+json
```

The request, with status code 200 would include, for example, the following json:

```

{
  "ietf-te:primary-paths": {
    "primary-path": [
      {
        "name": "path1",
        "path-computation-method": "te-types:path-locally-computed",
        "computed-paths-properties": {
          "computed-path-properties": [
            {
              "k-index": "1",
              "path-properties": {
                "path-route-objects": {
                  "path-route-object": [
                    {
                      "index": "1",
                      "numbered-node-hop": {
                        "node-id": "10.0.0.2"
                      }
                    },
                    {
                      "index": "2",
                      "numbered-node-hop": {
                        "node-id": "10.0.0.4"
                      }
                    }
                  ]
                }
              }
            }
          ]
        }
      }
    ]
  },
  "lsp": {
    "lsp": [
      {
        "tunnel-name": "Example_LSP_Tunnel_A_4_1",
        "node": "10.0.0.1 ",
        "lsp-id": "25356"
      }
    ]
  }
}

```

## 14. References

### 14.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC3209] Awduche, D., Berger, L., Gan, D., Li, T., Srinivasan, V., and G. Swallow, "RSVP-TE: Extensions to RSVP for LSP Tunnels", RFC 3209, DOI 10.17487/RFC3209, December 2001, <<https://www.rfc-editor.org/info/rfc3209>>.
- [RFC3473] Berger, L., Ed., "Generalized Multi-Protocol Label Switching (GMPLS) Signaling Resource ReserVation Protocol-Traffic Engineering (RSVP-TE) Extensions", RFC 3473, DOI 10.17487/RFC3473, January 2003, <<https://www.rfc-editor.org/info/rfc3473>>.
- [RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.17487/RFC3688, January 2004, <<https://www.rfc-editor.org/info/rfc3688>>.
- [RFC3945] Mannie, E., Ed., "Generalized Multi-Protocol Label Switching (GMPLS) Architecture", RFC 3945, DOI 10.17487/RFC3945, October 2004, <<https://www.rfc-editor.org/info/rfc3945>>.
- [RFC4206] Kompella, K. and Y. Rekhter, "Label Switched Paths (LSP) Hierarchy with Generalized Multi-Protocol Label Switching (GMPLS) Traffic Engineering (TE)", RFC 4206, DOI 10.17487/RFC4206, October 2005, <<https://www.rfc-editor.org/info/rfc4206>>.
- [RFC4872] Lang, J.P., Ed., Rekhter, Y., Ed., and D. Papadimitriou, Ed., "RSVP-TE Extensions in Support of End-to-End Generalized Multi-Protocol Label Switching (GMPLS) Recovery", RFC 4872, DOI 10.17487/RFC4872, May 2007, <<https://www.rfc-editor.org/info/rfc4872>>.
- [RFC4875] Aggarwal, R., Ed., Papadimitriou, D., Ed., and S. Yasukawa, Ed., "Extensions to Resource Reservation Protocol - Traffic Engineering (RSVP-TE) for Point-to-Multipoint TE Label Switched Paths (LSPs)", RFC 4875, DOI 10.17487/RFC4875, May 2007, <<https://www.rfc-editor.org/info/rfc4875>>.
- [RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, DOI 10.17487/RFC6020, October 2010, <<https://www.rfc-editor.org/info/rfc6020>>.

- [RFC6107] Shiomoto, K., Ed. and A. Farrel, Ed., "Procedures for Dynamically Signaled Hierarchical Label Switched Paths", RFC 6107, DOI 10.17487/RFC6107, February 2011, <<https://www.rfc-editor.org/info/rfc6107>>.
- [RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <<https://www.rfc-editor.org/info/rfc6241>>.
- [RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure Shell (SSH)", RFC 6242, DOI 10.17487/RFC6242, June 2011, <<https://www.rfc-editor.org/info/rfc6242>>.
- [RFC6780] Berger, L., Le Faucheur, F., and A. Narayanan, "RSVP ASSOCIATION Object Extensions", RFC 6780, DOI 10.17487/RFC6780, October 2012, <<https://www.rfc-editor.org/info/rfc6780>>.
- [RFC6991] Schoenwaelder, J., Ed., "Common YANG Data Types", RFC 6991, DOI 10.17487/RFC6991, July 2013, <<https://www.rfc-editor.org/info/rfc6991>>.
- [RFC7308] Osborne, E., "Extended Administrative Groups in MPLS Traffic Engineering (MPLS-TE)", RFC 7308, DOI 10.17487/RFC7308, July 2014, <<https://www.rfc-editor.org/info/rfc7308>>.
- [RFC7551] Zhang, F., Ed., Jing, R., and R. Gandhi, Ed., "RSVP-TE Extensions for Associated Bidirectional Label Switched Paths (LSPs)", RFC 7551, DOI 10.17487/RFC7551, May 2015, <<https://www.rfc-editor.org/info/rfc7551>>.
- [RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", RFC 7950, DOI 10.17487/RFC7950, August 2016, <<https://www.rfc-editor.org/info/rfc7950>>.
- [RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017, <<https://www.rfc-editor.org/info/rfc8040>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.

- [RFC8294] Liu, X., Qu, Y., Lindem, A., Hopps, C., and L. Berger, "Common YANG Data Types for the Routing Area", RFC 8294, DOI 10.17487/RFC8294, December 2017, <<https://www.rfc-editor.org/info/rfc8294>>.
- [RFC8340] Bjorklund, M. and L. Berger, Ed., "YANG Tree Diagrams", BCP 215, RFC 8340, DOI 10.17487/RFC8340, March 2018, <<https://www.rfc-editor.org/info/rfc8340>>.
- [RFC8341] Bierman, A. and M. Bjorklund, "Network Configuration Access Control Model", STD 91, RFC 8341, DOI 10.17487/RFC8341, March 2018, <<https://www.rfc-editor.org/info/rfc8341>>.
- [RFC8342] Bjorklund, M., Schoenwaelder, J., Shafer, P., Watsen, K., and R. Wilton, "Network Management Datastore Architecture (NMDA)", RFC 8342, DOI 10.17487/RFC8342, March 2018, <<https://www.rfc-editor.org/info/rfc8342>>.
- [RFC8343] Bjorklund, M., "A YANG Data Model for Interface Management", RFC 8343, DOI 10.17487/RFC8343, March 2018, <<https://www.rfc-editor.org/info/rfc8343>>.
- [RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <<https://www.rfc-editor.org/info/rfc8446>>.
- [RFC8639] Voit, E., Clemm, A., Gonzalez Prieto, A., Nilsen-Nygaard, E., and A. Tripathy, "Subscription to YANG Notifications", RFC 8639, DOI 10.17487/RFC8639, September 2019, <<https://www.rfc-editor.org/info/rfc8639>>.
- [RFC8641] Clemm, A. and E. Voit, "Subscription to YANG Notifications for Datastore Updates", RFC 8641, DOI 10.17487/RFC8641, September 2019, <<https://www.rfc-editor.org/info/rfc8641>>.
- [RFC8776] Saad, T., Gandhi, R., Liu, X., Beeram, V., and I. Bryskin, "Common YANG Data Types for Traffic Engineering", RFC 8776, DOI 10.17487/RFC8776, June 2020, <<https://www.rfc-editor.org/info/rfc8776>>.
- [RFC8795] Liu, X., Bryskin, I., Beeram, V., Saad, T., Shah, H., and O. Gonzalez de Dios, "YANG Data Model for Traffic Engineering (TE) Topologies", RFC 8795, DOI 10.17487/RFC8795, August 2020, <<https://www.rfc-editor.org/info/rfc8795>>.

## 14.2. Informative References

- [I-D.ietf-spring-segment-routing-policy]  
Filsfils, C., Talaulikar, K., Voyer, D., Bogdanov, A., and P. Mattes, "Segment Routing Policy Architecture", Work in Progress, Internet-Draft, draft-ietf-spring-segment-routing-policy-16, 28 January 2022, <<https://www.ietf.org/archive/id/draft-ietf-spring-segment-routing-policy-16.txt>>.
- [I-D.ietf-teas-yang-rsvp]  
Beeram, V. P., Saad, T., Gandhi, R., Liu, X., and I. Bryskin, "A YANG Data Model for Resource Reservation Protocol (RSVP)", Work in Progress, Internet-Draft, draft-ietf-teas-yang-rsvp-17, 9 January 2022, <<https://www.ietf.org/archive/id/draft-ietf-teas-yang-rsvp-17.txt>>.
- [RFC4427] Mannie, E., Ed. and D. Papadimitriou, Ed., "Recovery (Protection and Restoration) Terminology for Generalized Multi-Protocol Label Switching (GMPLS)", RFC 4427, DOI 10.17487/RFC4427, March 2006, <<https://www.rfc-editor.org/info/rfc4427>>.
- [RFC8800] Litkowski, S., Sivabalan, S., Barth, C., and M. Negi, "Path Computation Element Communication Protocol (PCEP) Extension for Label Switched Path (LSP) Diversity Constraint Signaling", RFC 8800, DOI 10.17487/RFC8800, July 2020, <<https://www.rfc-editor.org/info/rfc8800>>.
- [RFC9012] Patel, K., Van de Velde, G., Sangli, S., and J. Scudder, "The BGP Tunnel Encapsulation Attribute", RFC 9012, DOI 10.17487/RFC9012, April 2021, <<https://www.rfc-editor.org/info/rfc9012>>.

#### Authors' Addresses

Tarek Saad  
Juniper Networks  
  
Email: [tsaad@juniper.net](mailto:tsaad@juniper.net)

Rakesh Gandhi  
Cisco Systems Inc  
  
Email: [rgandhi@cisco.com](mailto:rgandhi@cisco.com)

Xufeng Liu  
Volta Networks

Email: xufeng.liu.ietf@gmail.com

Vishnu Pavan Beeram  
Juniper Networks

Email: vbeeram@juniper.net

Igor Bryskin  
Individual

Email: i\_bryskin@yahoo.com

Oscar Gonzalez de Dios  
Telefonica

Email: oscar.gonzalezdedios@telefonica.com



Link State Routing  
Internet-Draft  
Intended status: Standards Track  
Expires: May 4, 2020

K. Talaulikar  
P. Psenak  
Cisco Systems, Inc.  
A. Fu  
Bloomberg  
M. Rajesh  
Juniper Networks  
November 1, 2019

OSPF Strict-Mode for BFD  
draft-ketant-lsr-ospf-bfd-strict-mode-03

Abstract

This document specifies the extensions to OSPF that enables a router and its neighbor to signal their intention to use Bidirectional Forwarding Detection (BFD) for their adjacency using link-local advertisement between them. The signaling of this BFD enablement, allows the router to block and not allow the establishment of adjacency with its neighbor router until a BFD session is successfully established between them. The document describes this OSPF "strict-mode" of BFD establishment as a prerequisite to adjacency formation.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

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 <https://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 May 4, 2020.

#### Copyright Notice

Copyright (c) 2019 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 (<https://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. LLS B-bit Flag . . . . .                         | 3 |
| 3. Local Interface IPv4 Address TLV . . . . .       | 4 |
| 4. Procedures . . . . .                             | 4 |
| 4.1. OSPFv3 IPv4 Address-Family Specifics . . . . . | 6 |
| 4.2. Graceful Restart Considerations . . . . .      | 6 |
| 5. Operations & Management Considerations . . . . . | 6 |
| 6. Backward Compatibility . . . . .                 | 7 |
| 7. IANA Considerations . . . . .                    | 7 |
| 8. Security Considerations . . . . .                | 8 |
| 9. Acknowledgements . . . . .                       | 8 |
| 10. References . . . . .                            | 8 |
| 10.1. Normative References . . . . .                | 8 |
| 10.2. Informative References . . . . .              | 9 |
| Authors' Addresses . . . . .                        | 9 |

#### 1. Introduction

Bidirectional Forwarding Detection (BFD) [RFC5880] enables routers to monitor dataplane connectivity over links between them and to detect faults in the bidirectional path between them. This capability is leveraged by routing protocols like Open Shortest Path First (OSPFv2) [RFC2328] and OSPFv3 [RFC5340] to detect connectivity failures for their adjacencies and trigger the rerouting of traffic around this failure more quickly than their periodic hello messaging based detection mechanism.

The use of BFD for monitoring routing protocols adjacencies is described in [RFC5882]. When BFD monitoring is enabled for OSPF

adjacencies, the BFD session is bootstrapped based on the neighbor address information discovered by the exchange of OSPF hello messages. Faults in the bidirectional forwarding detected via BFD then result in the bringing down of the OSPF adjacency. Note that it is possible in some failure scenarios for the network to be in a state such that the OSPF adjacency is capable of coming up, but the BFD session cannot be established, and, more particularly, data cannot be forwarded. In certain other scenarios, a degraded or poor quality link may result in OSPF adjacency formation to succeed only to result in BFD session establishment not being successful or the BFD session going down frequently due to its faster detection mechanism.

To avoid such situations which result in routing churn in the network, it would be beneficial not to allow OSPF to establish a neighbor adjacency until the BFD session is successfully established and stabilized. However, this would preclude the OSPF operation in an environment in which not all OSPF routers support BFD and are enabled for BFD monitoring. A solution would be to block the establishment of OSPF adjacencies if both systems are willing to establish a BFD session but a BFD session cannot be established. Such a mode of BFD use by OSPF is referred to as "strict-mode" wherein BFD session establishment becomes a prerequisite for OSPF adjacency coming up.

This document specifies the OSPF protocol extensions using link-local signaling (LLS) [RFC5613] for a router to indicate to its neighbor the willingness to establish a BFD session in the "strict-mode". It also introduces an extension for OSPFv3 link-local signaling of interface IPv4 address when used for IPv4 address-family (AF) instance to enable discovery of the IPv4 addresses for BFD session setup.

A similar functionality for IS-IS is specified [RFC6213].

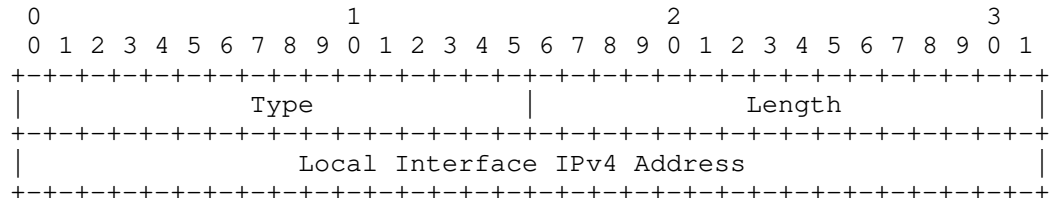
## 2. LLS B-bit Flag

A new B-bit is defined in the LLS Type 1 Extended Options and Flags field. This bit is defined for the LLS block included in Hello packets and indicates that BFD is enabled on the link and that the router supports BFD strict-mode. Section 7 describes the position of this new B-bit.

A router MUST include the LLS block with the LLS Type 1 Extended Options and Flags TLV with the B-bit set its Hello messages when BFD is enabled on the link.

### 3. Local Interface IPv4 Address TLV

The Local Interface IPv4 Address TLV is a new LLS TLV meant for OSPFv3 protocol operations for IPv4 AF instances [RFC5838]. It has following format:



where:

Type: TBD, suggested value 21

Length: 4 octet

Local Interface IPv4 Address: The primary IPv4 address of the local interface.

### 4. Procedures

A router supporting BFD strict-mode advertises this capability through its hello messages as described in Section 2 above. When a router supporting BFD strict-mode, detects a new neighbor router that also supports BFD strict-mode, then it proceeds to establish adjacency with that neighbor as described further in this section.

This document updates the OSPF neighbor state machine as described in [RFC2328] specifically the operations related to the Init state as below when BFD strict-mode is used:

Init (without BFD strict-mode)

In this state, an Hello packet has recently been seen from the neighbor. However, bidirectional communication has not yet been established with the neighbor (i.e., the router itself did not appear in the neighbor's Hello packet). All neighbors in this state (or higher) are listed in the Hello packets sent from the associated interface.

Init (with BFD strict-mode)

In this state, an Hello packet has recently been seen from the neighbor. However, bidirectional communication has not yet been

established with the neighbor (i.e., the router itself did not appear in the neighbor's Hello packet). A BFD session establishment to the neighbor is requested, if not already done (e.g. in the event of transition from 2-way state). All neighbors in higher than Init state and those in Init state with BFD session up are listed in the Hello packets sent from the associated interface.

Whenever the neighbor state transitions to Down state, the removal of the BFD session associated with that neighbor SHOULD be requested by OSPF and the session re-setup SHOULD similarly be requested by OSPF after transitioning into Init state. This may result in the deletion and creation of BFD session respectively when OSPF is the only client interested in BFD session to the neighbor address.

An implementation MUST NOT wait for BFD session establishment in Init state unless BFD strict-mode is enabled on the router and the specific neighbor indicates BFD strict-mode capability via its Hello messages. When BFD is enabled, but the strict-mode of operation cannot be used, then an implementation SHOULD start the BFD session establishment only in 2-Way or higher state. This makes it possible for router to operate a mix of BFD operation in strict-mode or normal mode across different interfaces or even different neighbors on the same multi-access LAN interface.

Once the OSPF state machine has moved beyond the Init state, any change in the B-bit advertised in subsequent Hello messages MUST NOT result in any trigger in either the OSPF adjacency or the BFD session management (i.e. the B-bit is considered only when in the Init state). The disabling of BFD (or BFD strict-mode) on a router would result in its not setting the B-bit in its subsequent Hello messages. The disabling of BFD strict-mode has no change on the BFD operations and would not result in bringing down of any established BFD session. The disabling of BFD would result in the BFD session brought down due to Admin reason and hence would not bring down the OSPF adjacency.

When BFD is enabled on an interface over which we already have an existing OSPF adjacency, it would result in the router setting the B-bit in its subsequent Hello messages. If the adjacency is already up (i.e. in its terminal state of Full or 2-way with non-DR routers on a LAN) with a neighbor that also support BFD strict-mode, then an implementation SHOULD NOT bring this adjacency down and instead use the BFD strict-mode of operations after the next transition into Init state. However, if the adjacency is not up, then an implementation MAY bring such an adjacency down so it can use the BFD strict-mode for its bring up.

#### 4.1. OSPFv3 IPv4 Address-Family Specifics

The multiple AF support in OSPFv3 [RFC5838] requires the use of IPv6 link-local address as source address for hello packets even when forming adjacencies for IPv4 AF instances. In most deployments of OSPFv3 IPv4 AF, it is required that BFD be used to monitor and verify the IPv4 data plane connectivity between the routers on the link and hence the BFD session is setup using IPv4 neighbor addresses. The IPv4 neighbor address on the interface is learnt only later in the adjacency formation phase when the neighbor's Link-LSA is received. This results in the setup of the BFD session either after the adjacency is established or much later in the adjacency formation sequence.

To enable the BFD operations in strict-mode, it is necessary for a router to learn its neighbor's IPv4 link address during the Init state of adjacency formation (ideally when it receives the first hello). The use of the Local Interface IPv4 Address TLV (as defined in Section 3) in the LLS block of the OSPFv3 Hello messages for IPv4 AF instances makes this possible. Implementations that support strict-mode of BFD operations for OSPFv3 IPv4 AF instances MUST include the Local Interface IPv4 Address TLV in the LLS block of their hello messages whenever the B-bit is set. A receiver MUST ignore the B-bit (i.e. not operate in BFD strict mode) unless the Local Interface IPv4 Address TLV is present in OSPFv3 Hello message for IPv4 AF instances.

#### 4.2. Graceful Restart Considerations

An implementation needs to handle scenarios where both graceful restart (GR) and the strict-mode of BFD operations are deployed together. The GR aspects discussed in [RFC5882] also apply with strict-mode of operations. In addition to that, since the OSPF adjacency formation is held up until the BFD session establishment in the strict-mode of operation, the resultant delay in adjacency formation may affect or break the GR based recovery. In such cases, it is RECOMMENDED that the GR timers are setup such that they provide sufficient time to cover for normal BFD session establishment delays.

#### 5. Operations & Management Considerations

An implementation SHOULD report the BFD session status along with the OSPF Init adjacency state when operating in BFD strict-mode and perform logging operations on state transitions to include the BFD events. This allows an operator to detect scenarios where an OSPF adjacency may be stuck waiting for BFD session establishment.

In network deployments with noisy links or those with packet loss, BFD sessions may flap frequently. In such scenarios, OSPF strict-mode for BFD may be deployed in conjunction with an BFD dampening or hold-down mechanism to help avoid frequent adjacency flaps due BFD causing routing churn.

## 6. Backward Compatibility

An implementation MUST support OSPF adjacency formation and operations with a neighbor router that does not advertise the BFD strict-mode capability - both when that neighbor router does not support BFD and when it does support BFD but not in the strict-mode of operation as described in this document. Implementations MAY provide an option to specifically enable BFD operations only in the strict-mode in which case, OSPF adjacency with a neighbor that does not support BFD strict-mode would not be established successfully. Implementations MAY provide an option to disable BFD strict-mode which results in the router not advertising the B-bit and BFD operations being performed in the same way as before this specification.

The signaling specified in this document happens at a link-local level between routers on that link. A router which does not support this specification would ignore the B-bit in the LLS block of hello messages from its neighbors and continue to bootstrap BFD sessions, if enabled, without holding back the OSPF adjacency formation. Since the router which does not support this specification would not have set the B-bit in the LLS block of its own hello messages, its neighbor routers that support this specification would not use BFD strict-mode with it. As a result, the behavior would be the same as before this specification. Therefore, there are no backward compatibility related issues or considerations that need to be taken care of when implementing this specification.

## 7. IANA Considerations

This specification updates Link Local Signaling TLV Identifiers registry.

Following values are requested for allocation:

- o B-bit from "LLS Type 1 Extended Options and Flags" registry at bit position 0x00000010.
- o TBD (Suggested value 21) - Local Interface IPv4 Address TLV

## 8. Security Considerations

The security considerations for "OSPF Link-Local Signaling" [RFC5613] also apply to the extension described in this document. Inappropriate use of the B-bit in the LLS block of an OSPF hello message could prevent an OSPF adjacency from forming or lead to failure to detect bidirectional forwarding failures. If authentication is being used in the OSPF routing domain [RFC5709][RFC7474], then the Cryptographic Authentication TLV [RFC5613] SHOULD also be used to protect the contents of the LLS block.

## 9. Acknowledgements

The authors would like to acknowledge the review and inputs from Acee Lindem, Manish Gupta, Balaji Ganesh and Rajesh M.

The authors would like to acknowledge Dylan van Oudheusden for highlighting the problems in using strict-mode for BFD session for IPv4 AF instance with OSPFv3 and Baalajee S for his suggestions on the approach to address it.

## 10. References

### 10.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC2328] Moy, J., "OSPF Version 2", STD 54, RFC 2328, DOI 10.17487/RFC2328, April 1998, <<https://www.rfc-editor.org/info/rfc2328>>.
- [RFC5340] Coltun, R., Ferguson, D., Moy, J., and A. Lindem, "OSPF for IPv6", RFC 5340, DOI 10.17487/RFC5340, July 2008, <<https://www.rfc-editor.org/info/rfc5340>>.
- [RFC5613] Zinin, A., Roy, A., Nguyen, L., Friedman, B., and D. Yeung, "OSPF Link-Local Signaling", RFC 5613, DOI 10.17487/RFC5613, August 2009, <<https://www.rfc-editor.org/info/rfc5613>>.
- [RFC5838] Lindem, A., Ed., Mirtorabi, S., Roy, A., Barnes, M., and R. Aggarwal, "Support of Address Families in OSPFv3", RFC 5838, DOI 10.17487/RFC5838, April 2010, <<https://www.rfc-editor.org/info/rfc5838>>.



- [RFC5882] Katz, D. and D. Ward, "Generic Application of Bidirectional Forwarding Detection (BFD)", RFC 5882, DOI 10.17487/RFC5882, June 2010, <<https://www.rfc-editor.org/info/rfc5882>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.

## 10.2. Informative References

- [RFC5709] Bhatia, M., Manral, V., Fanto, M., White, R., Barnes, M., Li, T., and R. Atkinson, "OSPFv2 HMAC-SHA Cryptographic Authentication", RFC 5709, DOI 10.17487/RFC5709, October 2009, <<https://www.rfc-editor.org/info/rfc5709>>.
- [RFC5880] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD)", RFC 5880, DOI 10.17487/RFC5880, June 2010, <<https://www.rfc-editor.org/info/rfc5880>>.
- [RFC6213] Hopps, C. and L. Ginsberg, "IS-IS BFD-Enabled TLV", RFC 6213, DOI 10.17487/RFC6213, April 2011, <<https://www.rfc-editor.org/info/rfc6213>>.
- [RFC7474] Bhatia, M., Hartman, S., Zhang, D., and A. Lindem, Ed., "Security Extension for OSPFv2 When Using Manual Key Management", RFC 7474, DOI 10.17487/RFC7474, April 2015, <<https://www.rfc-editor.org/info/rfc7474>>.

## Authors' Addresses

Ketan Talaulikar  
Cisco Systems, Inc.  
India

Email: [ketant@cisco.com](mailto:ketant@cisco.com)

Peter Psenak  
Cisco Systems, Inc.  
Apollo Business Center  
Mlynske nivy 43  
Bratislava 821 09  
Slovakia

Email: [ppsenak@cisco.com](mailto:ppsenak@cisco.com)

Albert Fu  
Bloomberg  
USA

Email: [afu14@bloomberg.net](mailto:afu14@bloomberg.net)

Rajesh M  
Juniper Networks  
India

Email: [mrjesh@juniper.net](mailto:mrjesh@juniper.net)

IDR WorkGroup  
Internet-Draft  
Intended status: Standards Track  
Expires: January 8, 2020

M. Zheng  
A. Lindem  
Cisco Systems  
J. Haas  
Juniper Networks, Inc.  
July 7, 2019

BGP BFD Strict-Mode  
draft-merciaz-idr-bgp-bfd-strict-mode-02

Abstract

This document specifies extensions to RFC4271 BGP-4 that enable a BGP speaker to negotiate additional Bidirectional Forwarding Detection (BFD) extensions using a BGP capability. This BFD capability enables a BGP speaker to prevent a BGP session from being established until a BFD session is established. It is referred to as BGP BFD "strict-mode". BGP BFD strict-mode will be supported when both the local speaker and its remote peer are BFD strict-mode capable.

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 <https://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 January 8, 2020.

Copyright Notice

Copyright (c) 2019 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 (<https://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. Requirements Language . . . . .        | 3 |
| 3. BFD Capability . . . . .               | 3 |
| 4. Operation . . . . .                    | 4 |
| 5. Manageability Considerations . . . . . | 5 |
| 6. Security Considerations . . . . .      | 5 |
| 7. IANA Considerations . . . . .          | 5 |
| 8. Acknowledgement . . . . .              | 5 |
| 9. Normative References . . . . .         | 6 |
| Authors' Addresses . . . . .              | 6 |

## 1. Introduction

Bidirectional Forwarding Detection BFD [RFC5882] enables routers to monitor data plane connectivity and to detect faults in the bidirectional forwarding path between them. This capability is leveraged by routing protocols such as BGP [RFC4271] to rapidly react to topology changes in the face of path failures.

The BFD interaction with BGP is specified in Section 10.2 of [RFC5882]. When BFD is enabled for a BGP neighbor, faults in the bidirectional forwarding detected by BFD result in session termination. It is possible in some failure scenarios for the network to be in a state such that a BGP session may be established but a BFD session cannot be established. In some other scenarios, it may be possible to establish a BGP session, but a degraded or poor-quality link may result in the corresponding BFD session going up and down frequently.

To avoid situations which result in routing churn and to minimize the impact of network interruptions, it will be beneficial to disallow BGP to establish a session until BFD session is successfully established and has stabilized. We refer to this mode of operation as BGP BFD "strict-mode". However, always using "strict-mode" would preclude BGP operation in an environment where not all routers support BFD strict-mode or have BFD enabled. This document defines BGP "strict-mode" operation as preventing BGP session establishment until both the local and remote speakers have a stable BFD session. The document also specifies the BGP protocol extensions for BGP capability [RFC5492] for announcing BFD parameters including a BGP

speaker's support for "strict-mode", i.e., requiring a BFD session for BGP session establishment.

## 2. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

## 3. BFD Capability

The BGP Capability [RFC5492] for BFD parameters will allow a BGP speaker's BFD capabilities including its support for BFD strict-mode. This capability is defined as follows:

Capability code: TBD

Capability length: 1 octet

Capability value: Consists of 1 octet BFD flags as follows:

```
+-----+
| BFD Flags (8 bits) |
+-----+
```

The use and meaning of the fields are as follows:

BFD Flags: This field contains bit flags relating to BFD.

```
  0 1 2 3 4 5 6 7
+---+---+---+---+
| S | Reserved |
+---+---+---+---+
```

The most significant bit is defined as state of Strict-Mode ("Strict-Mode", or "S") bit, which can be used by a BGP speaker to signal its support for BFD Strict-mode. When set (value 1), this bit indicates that the BGP speaker has the BFD "Strict-mode" enabled. If both local BGP speaker and its peer have BFD strict-mode enabled, then BGP session establishment will be prevented until a BFD session is established between the peering addresses. A BGP speaker with BFD

strict-mode enabled MUST advertise the BFD capability with "S" bit set.

The remaining bits are reserved and SHOULD be set to zero by the sender and MUST be ignored by the receiver.

#### 4. Operation

A BGP speaker which supports capabilities advertisement and has BFD strict-mode enabled MUST include the BGP BFD capability with the "S" Bit set in the BGP capabilities it advertises.

A BGP speaker which supports BFD capability, examines the list of capabilities present in the Capabilities BFD Parameter that the speaker receives from its peer. If both the local and remote BGP speakers have BFD strict-mode enabled, the BGP finite state machine does not transition to the Established state from OpenSent or OpenConfirm state [RFC4271] until the BFD session is in the Up state (see below for AdminDown state). This means that a KEEPALIVE message is not sent nor is the KeepaliveTimer set.

If the BFD session does not transition to the Up state, and the HoldTimer has been negotiated to a non-zero value, the BGP FSM will close the session appropriately. If the HoldTimer has been negotiated to a zero value, the session should be closed after a time of X. This time X is referred as "BGP BFD Hold time". The proposed default BGP BFD Hold time value is 30 seconds. The BGP BFD Hold time value is configurable.

If BFD session is in the AdminDown state, then the BGP finite state machine will proceed normally without input from BFD. This means that BFD session "AdminDown" state WILL NOT prevent the BGP state transition to Established state from OpenConfirm.

Once the BFD session has transitioned to the Up state, the BGP FSM may proceed to transition to the Established state from the OpenSent or OpenConfirm state appropriately. I.e. a KEEPALIVE message is sent, and the KeepaliveTimer is started.

If either BGP peer has not advertised the BFD Capability with strict-mode enabled, then a BFD session WILL NOT be required for the BGP session to reach Established state. This does not preclude usage of BFD after BGP session establishment [RFC5882].

## 5. Manageability Considerations

Auto-configuration is possible for the enabling BGP BFD restrict-mode. However, the configuration automation is out of the scope of this document.

A BGP NOTIFICATION message subcode indicating BFD Hold timer expiration may be required for network management. (To be discussed in the next revision of this document.)

## 6. Security Considerations

The mechanism defined in this document interacts with the BGP finite state machine when so configured. The security considerations of BFD thus become considerations for BGP-4 [RFC4271] so used. The use of the BFD Authentication mechanism defined in [RFC5880] is thus RECOMMENDED when used to protect BGP-4 [RFC4271].

## 7. IANA Considerations

This document defines a new BGP capability - BFD Capability. The Capability Code for BFD Capability is TBD.

IANA is requested to establish a "BGP BFD Capability Flags" registry within the "Border Gateway Protocol (BGP) Parameters" grouping. The Registration Procedure should be Standards Action, the initial values as follows:

| Bit Position | Name        | Short Name | Reference     |
|--------------|-------------|------------|---------------|
| 0            | Strict-Mode | S          | this document |
| 1-7          | Unassigned  |            | this document |

## 8. Acknowledgement

The authors would like to acknowledge the review and inputs from Shyam Sethuram, Mohammed Mirza, Bruno Decraene, and Carlos Pignataro.

## 9. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC4271] Rekhter, Y., Ed., Li, T., Ed., and S. Hares, Ed., "A Border Gateway Protocol 4 (BGP-4)", RFC 4271, DOI 10.17487/RFC4271, January 2006, <<https://www.rfc-editor.org/info/rfc4271>>.
- [RFC5492] Scudder, J. and R. Chandra, "Capabilities Advertisement with BGP-4", RFC 5492, DOI 10.17487/RFC5492, February 2009, <<https://www.rfc-editor.org/info/rfc5492>>.
- [RFC5880] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD)", RFC 5880, DOI 10.17487/RFC5880, June 2010, <<https://www.rfc-editor.org/info/rfc5880>>.
- [RFC5882] Katz, D. and D. Ward, "Generic Application of Bidirectional Forwarding Detection (BFD)", RFC 5882, DOI 10.17487/RFC5882, June 2010, <<https://www.rfc-editor.org/info/rfc5882>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.

## Authors' Addresses

Mercia Zheng  
Cisco Systems  
821 Alder Drive  
MILPITAS, CALIFORNIA 95035  
UNITED STATES

Email: [merciarz@cisco.com](mailto:merciarz@cisco.com)

Acee Lindem  
Cisco Systems  
301 Midenhall Way  
GARY, NC 27513  
UNITED STATES

Email: [acee@cisco.com](mailto:acee@cisco.com)



Jeffrey Haas  
Juniper Networks, Inc.  
1133 Innovation Way  
SUNNYVALE, CALIFORNIA 94089  
UNITED STATES

Email: [jhaas@juniper.net](mailto:jhaas@juniper.net)

BFD Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: October 16, 2020

G. Mirsky  
X. Min  
ZTE Corp.  
April 14, 2020

Extended Bidirectional Forwarding Detection  
draft-mirmin-bfd-extended-03

Abstract

This document describes a mechanism to extend the capabilities of Bidirectional Forwarding Detection (BFD). These extensions enable BFD to measure performance metrics like packet loss and packet delay. Also, a method to perform lightweight on-demand authentication is defined in this specification.

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 <https://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 October 16, 2020.

Copyright Notice

Copyright (c) 2020 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 (<https://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. Conventions used in this document . . . . .                           | 3  |
| 2.1. Terminology . . . . .                                               | 3  |
| 2.2. Requirements Language . . . . .                                     | 3  |
| 3. Extended BFD Control Message . . . . .                                | 3  |
| 3.1. Extended BFD Capability Negotiation . . . . .                       | 5  |
| 3.2. Padding TLV . . . . .                                               | 6  |
| 3.3. Diagnostic TLV . . . . .                                            | 7  |
| 3.4. Performance Measurement with Extended BFD Control Message . . . . . | 8  |
| 3.5. Lightweight Authentication . . . . .                                | 9  |
| 3.5.1. Lightweight Authentication Mode Negotiation . . . . .             | 10 |
| 3.5.2. Using Lightweight Authentication . . . . .                        | 11 |
| 4. IANA Considerations . . . . .                                         | 12 |
| 4.1. Extended BFD Message Types . . . . .                                | 12 |
| 4.2. Lightweight Authentication Modes . . . . .                          | 13 |
| 4.3. Return Codes . . . . .                                              | 14 |
| 5. Security Considerations . . . . .                                     | 14 |
| 6. References . . . . .                                                  | 15 |
| 6.1. Normative References . . . . .                                      | 15 |
| 6.2. Informative References . . . . .                                    | 15 |
| Appendix A. Acknowledgements . . . . .                                   | 15 |
| Authors' Addresses . . . . .                                             | 16 |

## 1. Introduction

[RFC5880] has provided the base specification of Bidirectional Detection (BFD) as the light-weight mechanism to monitor a path continuity between two systems and detect a failure in the data-plane. Since its introduction, BFD has been broadly deployed. There were several attempts to introduce new capabilities in the protocol, some more successful than others. One of the significant obstacles to extending BFD capabilities may be seen in the compact format of the BFD control message. This document introduces an Extended BFD control message and describes the use of the new format for new BFD capabilities.

The Extended BFD protocol may be seen as the Operations, Administration, and Maintenance (OAM) protocol that provides both Fault Management (FM) Performance Monitoring (PM) OAM functions. In some networks, for example in a Deterministic Networking (DetNet) domain [RFC8655], it is easier to ensure that a test packet of a single OAM protocol is fate-sharing with data packets rather than map several FM and PM OAM protocols to that DetNet data flow.

## 2. Conventions used in this document

### 2.1. Terminology

BFD: Bidirectional Forwarding Detection

G-ACh Generic Associated Channel

HMAC Hashed Message Authentication Code

MTU Maximum Transmission Unit

PMTUD Path MTU Discovery

PMTUM Path MTU Monitoring

p2p: Point-to-Point

TLV Type, Length, Value

OAM Operations, Administration, and Maintenance

FM Fault Management

PM Performance Monitoring

DetNet Deterministic Networking

### 2.2. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

## 3. Extended BFD Control Message

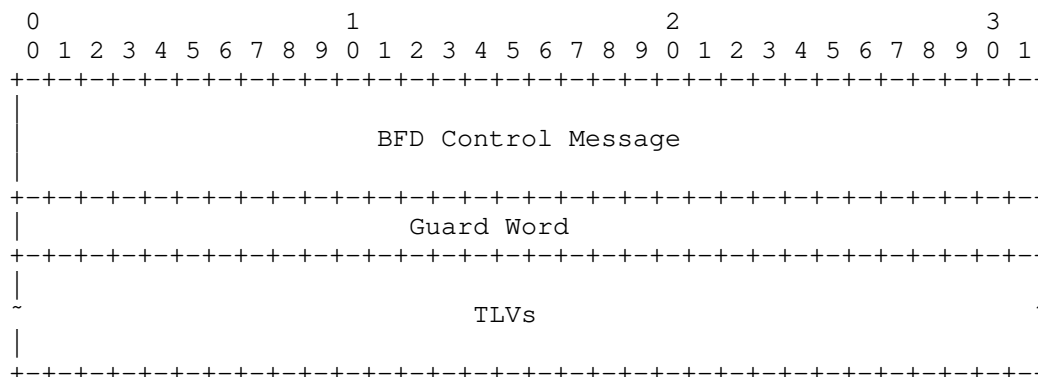


Figure 1: Extended BFD Control Message Format

where fields are defined as the following:

- o BFD control message as defined [RFC5880].
- o Guard word - four octets long field to identify the role of the BFD system - sender or responder.
- o TLVs - variable-length field that contains commands and/or data encoded as type-length-value (TLV).

If an Extended BFD control message is encapsulated in IP/UDP, the value of the Total Length in the IP header includes the length of the Extended BFD control message while the value of the Length field of the BFD control message equals the value as defined in [RFC5880]. If an Extended BFD control message is to be used over Generic Associated Channel (G-ACh), e.g., [RFC6428] new code point for G-ACh may be allocated.

Figure 2 displays the generic TLV format.

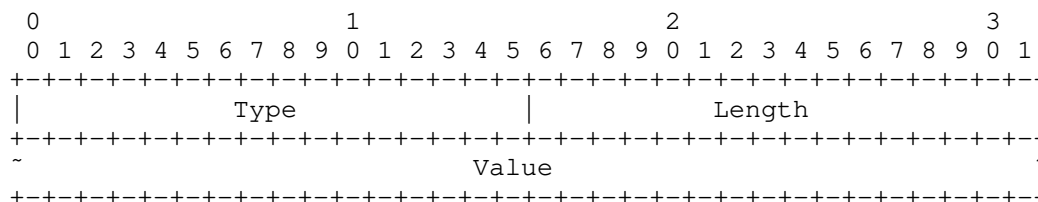


Figure 2: General Type-Length-Value Encoding

where fields are defined as the following:

- o Type - two octets long field that defines the encoding of the Value field
- o Length - two octets long field equals length on the Value field in octets.
- o Value - depends on the Type.

TLVs may be included within other TLVs, in which case the former TLVs are referred to as sub-TLVs. Sub-TLVs have independent types.

### 3.1. Extended BFD Capability Negotiation

A BFD system also referred to as a node in this document, that supports Extended BFD first MUST discover whether other nodes in the given BFD session support the Extended BFD. The node MUST send Extended BFD control message initiating the Poll Sequence as defined in [RFC5880]. If the remote system fails to respond with the Extended BFD control message and the Final flag set, then the initiator node MUST conclude that the BFD peer does not support the use of the Extended BFD control messages.

The first Extended BFD control message initiating the Poll Sequence SHOULD include the Capability TLV that lists capabilities that may be used at some time during the lifetime of the BFD session. The format of the Capability TLV and the capabilities that use the Extended BFD control message are presented in Figure 3.

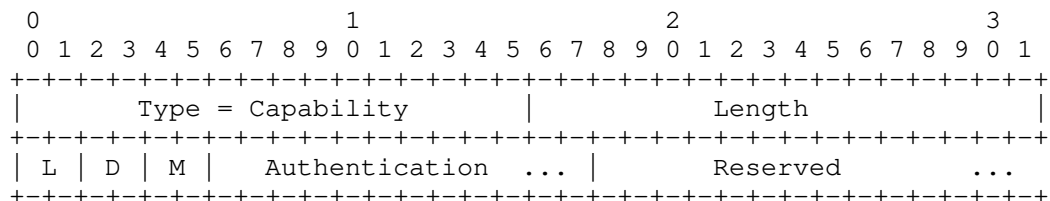


Figure 3: Capability TLV Format

where fields are defined as the following:

- o Type - TBA1 allocated by IANA in Section 4
- o Length - two octets long field equals length on the Capability field in octets. The value of the Length field MUST be a multiple of 4.
- o Loss - two bits size field. The least significant of two bits is set if the node is capable of measuring packet loss using

periodically transmitted Extended BFD control message. The most significant of two bits is set if the node is capable of measuring packet loss using the Poll Sequence with Extended BFD control message.

- o Delay - two bits size field. The least significant of two bits is set if the node is capable of measuring packet delay using periodically transmitted Extended BFD control message. The most significant of two bits is set if the node is capable of measuring packet delay using the Poll Sequence with Extended BFD control message.
- o MTU - two bits size field. Set if the node is capable of using the Extended BFD control message in Path MTU Discovery (PMTUD). or PMTU Monitoring (PMTUM). The least significant of two bits is set if the node is capable of PMTUD/PMTUM using periodically transmitted Extended BFD control message. The most significant of two bits is set if the node is capable of PMTUD/PMTUM using the Poll Sequence with Extended BFD control message.
- o (Lightweight) Authentication - variable-length field. The Authentication field is used by a BFD system to advertise its lightweight authentication capabilities. The format and the use of the Authentication field are defined in Section 3.5.1.
- o Reserved - MUST be zeroed on transmission and ignored on receipt. The Reserved field is zero-padded to align the length of the Capability TLV to a 4-octet boundary.

The remote BFD node that supports this specification MUST respond to the Capability TLV with the Extended BFD control message that includes the Capability TLV listing capabilities the responder supports. The responder MUST set the Final flag in the Extended BFD control message.

### 3.2. Padding TLV

Padding TLV MAY be used to generate Extended BFD control packets of the desired length.

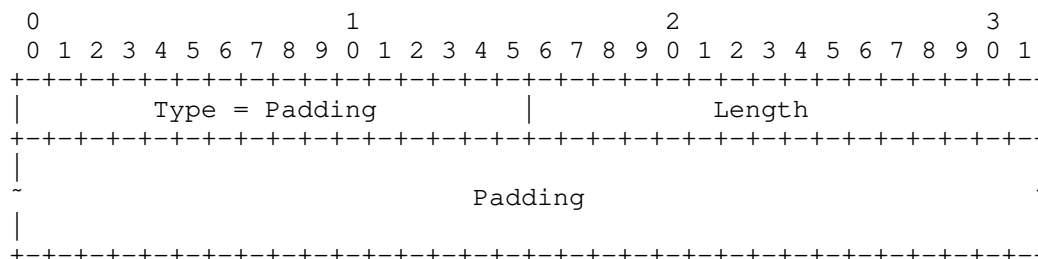


Figure 4: Padding TLV Format

where fields are defined as the following:

- o Type - TBA1 allocated by IANA in Section 4
- o Length - two octets long field equals length on the Padding field in octets.
- o Padding - variable-length field. MUST be zeroed on transmit and ignored on receipt.

Padding TLV MAY be used to generate Extended BFD Control packets of different lengths. That capability is necessary to perform PMTUD, PMTUM, and measure synthetic packet loss and/or packet delay. When Padding TLV is used in combination with one of performance measurement messages carried in Performance Metric TLVs as defined in Section 3.4, Padding TLV MUST follow the Performance Metric TLV.

Padding TLV MAY be used in PMTUM as part of periodically sent Extended BFD Control messages. In this case, the number of consecutive messages that include Padding TLV MUST be not lesser than Detect Multiplier to ensure that the remote BFD peer will detect loss of messages with the Padding TLV. Also, Padding TLV MAY be present in an Extended BFD Control message with the Poll flag set. If the remote BFD peer that supports this specification receives an Extended BFD Control message with Padding TLV, it MUST include the Padding TLV with the Padding field of the same length as in the received packet and set the Final flag.

### 3.3. Diagnostic TLV

Diagnostic TLV MAY be used to characterize the result of the last requested operation.



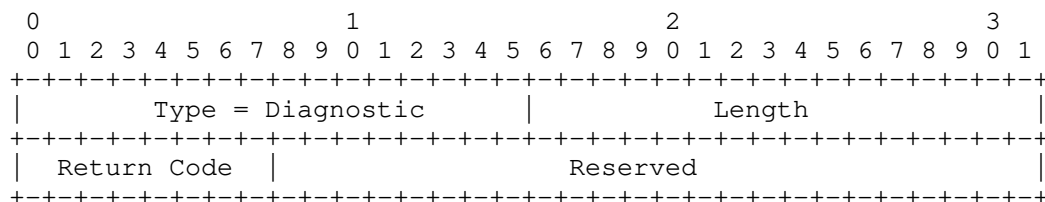


Figure 5: Diagnostic TLV Format

where fields are defined as the following:

- o Type - TBA6 allocated by IANA in Section 4.
- o Length - MUST be set to four.
- o Return Code - eight bits-long field. The responding BFD system can set it to one of the values defined in Section 4.3.
- o Reserved - 24 bits-long field. MUST be zeroed on transmit and ignored on receipt.

### 3.4. Performance Measurement with Extended BFD Control Message

Loss measurement, delay measurement, and loss/delay measurement messages can be used in the Extended BFD control message to support one-way and round-trip measurements. All the messages are encapsulated as TLVs with Type values allocated by IANA, Section 4.

The sender MAY use the Performance Metric TLV (presented in Figure 6) to measure performance metrics and obtain the measurement report from the receiver.

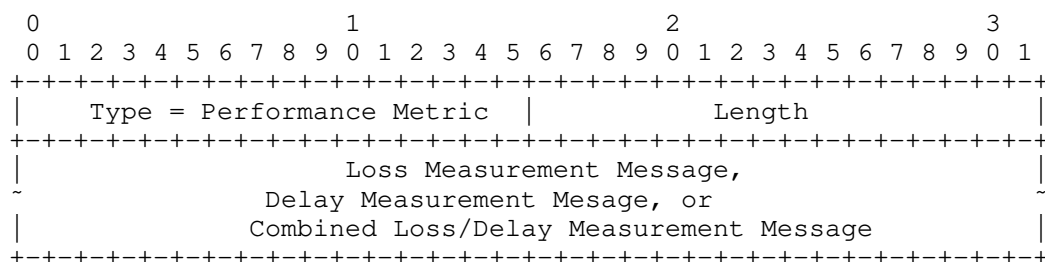


Figure 6: Performance Metric TLV Format

where fields are defined as the following:

- o Type - TBA3 through TBA5 allocated by IANA in Section 4 as follows:
  - \* TBA3 - Loss Measurement Type;
  - \* TBA4 - Delay Measurement Type;
  - \* TBA5 - Combined Loss/Delay Measurement Type
- o Length - two octets long field equals length on the Metric sub-TLVs field in octets. The value of the Length field MUST be a multiple of 4.
- o Value - various performance metrics measured either directly or using synthetic methods accordingly using the messages defined in Sections 3.1 through 3.3 [RFC6374].

To perform one-way loss and/or delay measurement, the BFD node MAY periodically transmit the Extended BFD message with one of the TLVs listed above in Asynchronous mode. To perform synthetic loss measurement, the sender MUST monotonically increment the counter of transmitted test packets. When using Performance Metric TLV for synthetic measurement an Extended BFD Control message MAY also include Padding TLV. In that case, the Padding TLV MUST immediately follow Performance Metric TLV. Also, direct-mode loss measurement, as described in [RFC6374], is supported. Procedures to negotiate and manipulate transmission intervals defined in Sections 6.8.2 and 6.8.3 in [RFC5880] SHOULD be used to control the performance impact of using the Extended BFD for performance measurement in the particular BFD session.

To measure the round-trip loss and/or delay metrics the BFD node transmits the Extended BFD control message with the Performance Metric TLV with the Poll flag set. Before the transmission of the Extended BFD control message with the Performance Metric TLV, the receiver MUST clear the Poll flag and set the Final flag.

### 3.5. Lightweight Authentication

Using BFD without any security measures, for example, by exchanging BFD control packets without authentication, increases the risk of an attack, especially over multiple nodes. Thus, using BFD without security measures may cause false positive as well as false negative defect detection situations. In the former, an attacker may spoof BFD control packets pretending to be a remote peer and can thus view the BFD session operation even though the real path had failed. In the latter, the attacker may spoof altered BFD control message

indicating that the BFD session is un-operational even though the path and the remote BFD peer operate normally.

BFD technology[RFC5880] includes optional authentication protection of BFD control packets to minimize the chances of attacks in a networking system. However, at least some of the supported authentication protocols do not provide sufficient protection in modern networks. Also, current BFD technology requires authentication of each and every BFD control packet. Such an authentication requirement can put a computational burden on networking devices, especially in the Asynchronous mode, at least because authenticating each BFD control packet can require substantial computing resources to support packet exchange at high rates.

This specification defines a lightweight on-demand mode of authentication for a BFD session. The lightweight authentication is an optional mode that can be used when the BFD Authentication [RFC5880] is not in use (bfd.AuthType is zero). The mechanism includes negotiation (Section 3.5.1) and on-demand authentication (Section 3.5.2) phases. During the former, BFD peers advertise supported authentication capabilities and independently select the commonly supported mode of the authentication. In the authentication phase, each BFD system transmits, at certain events and periodically, authenticated BFD control packets in Poll Sequence.

### 3.5.1. Lightweight Authentication Mode Negotiation

Figure 7 displays the format of the Authentication field that is part of the Capability Encoding:

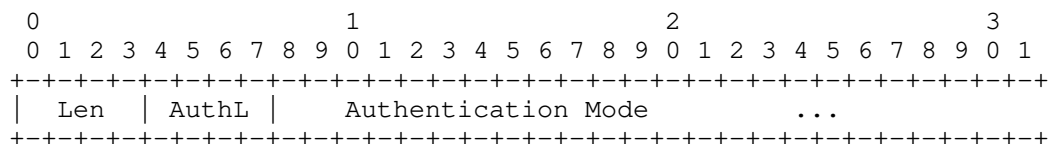


Figure 7: Lightweight Authentication Capability Field Format

where fields are defined as the following:

- o Len (Length) - four-bits long field. The value of the Length field is equal to the length of the Authentication field, including the Length, in octets.
- o AuthL (Authentication Length) - four bits size field. The value of the field is, in four octets long words, the longest

authentication signature the BFD system is capable of supporting for any of the methods advertised in the AuthMode field.

- o Authentication Mode - variable-length field. It is a bit-coded field that a BFD system uses to list modes of lightweight authentication it supports.

A BFD system uses Capability TLV, defined in Section 3.1, to discover the commonly supported mode of the Lightweight Authentication. The system sets the values in the Authentication field according to properly reflect its authentication capabilities. The BFD system transmits the Extended BFD control packet with Capability TLV as the first in a Poll Sequence. The remote BFD system that supports this specification receives the Extended BFD control packet with the advertised Lightweight Authentication modes and stores information locally. The system responds with the advertisement of its Lightweight Authentication capabilities in the Extended BFD control packet with the Final flag set. Each BFD system uses local and received information about Lightweight Authentication capabilities to deduce the commonly supported modes and selects from that set the one that uses the strongest authentication with the longest signature. If the common set is empty, i.e., none of supported by one BFD system authentication method is supported by another, an implementation **MUST** reflect this in its operational state and **SHOULD** notify an operator.

### 3.5.2. Using Lightweight Authentication

After BFD peers select an authentication mode for using in Lightweight Authentication each BFD system **MUST** use it to authenticate each Extended BFD control packet transmitted as part of a Poll Sequence using Lightweight Authentication TLV presented in Figure 8.

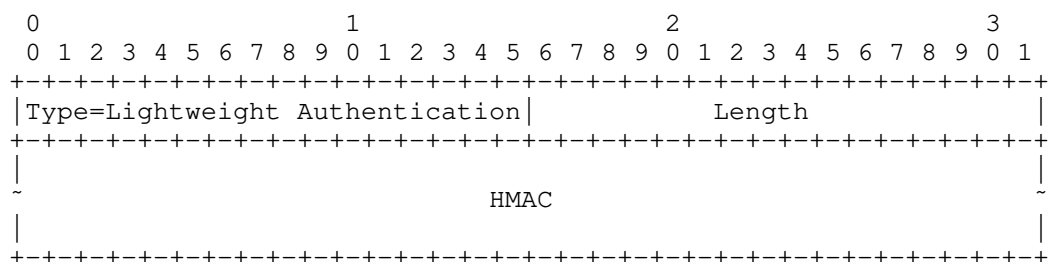


Figure 8: Lightweight Authentication TLV Format

where fields are defined as the following:

- o Type - TBA8 allocated by IANA in Section 4

- o Length - two octets long field equals length on the HMAC (Hashed Message Authentication Code) field in octets. The value of the Length field MUST be a multiple of 4.
- o HMAC - the hash value calculated on the entire preceding Extended BFD control packet data.

The Lightweight Authentication TLV MUST be the last TLV in an Extended BFD control packet. Padding TLV (Section 3.2) MAY be used to align the length of the Extended BFD control packet, excluding the Lightweight Authentication TLV, at multiple of 16 boundary.

The BFD system that receives the Extended BFD control packet with the Lightweight Authentication TLV MUST first validate the authentication by calculating the hash over the Extended BFD control packet. If the validation succeeds, the receiver MUST transmit the Extended BFD control packet with the Final flag set and the value of the Return code field in Diagnostic TLV set to None value (Table 5). If the validation of the lightweight authentication fails, then the BFD system MUST transmit the Extended BFD control packet with the Final flag set and the value of the Return Code field of the Diagnostic TLV set to Lightweight Authentication failed value (Table 5). The BFD system MUST have a control policy that defines actions when the system receives the Lightweight Authentication failed return code.

#### 4. IANA Considerations

##### 4.1. Extended BFD Message Types

IANA is requested to create the Extended BFD Message Types registry. All code points in the range 1 through 32759 in this registry shall be allocated according to the "IETF Review" procedure as specified in [RFC8126]. Code points in the range 32760 through 65279 in this registry shall be allocated according to the "First Come First Served" procedure as specified in [RFC8126]. Remaining code points are allocated according to Table 1:

| Value         | Description                  | Reference               |
|---------------|------------------------------|-------------------------|
| 0             | Reserved                     | This document           |
| 1- 32767      | Mandatory TLV,<br>unassigned | IETF Review             |
| 32768 - 65279 | Optional TLV,<br>unassigned  | First Come First Served |
| 65280 - 65519 | Experimental                 | This document           |
| 65520 - 65534 | Private Use                  | This document           |
| 65535         | Reserved                     | This document           |

Table 1: Extended BFD Type Registry

This document defines the following new values in Extended BFD Type registry:

| Value | Description                     | Reference     |
|-------|---------------------------------|---------------|
| TBA1  | Padding                         | This document |
| TBA2  | Capability                      | This document |
| TBA3  | Loss Measurement                | This document |
| TBA4  | Delay Measurement               | This document |
| TBA5  | Combined Loss/Delay Measurement | This document |
| TBA6  | Diagnostic                      | This document |
| TBA8  | Lightweight Authentication      | This document |

Table 2: Extended BFD Types

#### 4.2. Lightweight Authentication Modes

IANA is requested to create a Lightweight Authentication Modes registry. All code points in this registry shall be allocated according to the "IETF Review" procedure as specified in [RFC8126].

This document defines the following new values in the Lightweight Authentication Modes registry:

| Bit Position | Value | Description            | Reference     |
|--------------|-------|------------------------|---------------|
| 0            | 0x1   | Keyed SHA-1            | This document |
| 1            | 0x2   | Meticulous Keyed SHA-1 | This document |
| 2            | 0x4   | SHA-256                | This document |

Table 3: Lightweight Authentication Modes

#### 4.3. Return Codes

IANA is requested to create the Extended BFD Return Codes registry. All code points in the range 1 through 250 in this registry shall be allocated according to the "IETF Review" procedure as specified in [RFC8126]. Remaining code points are allocated according to Table 4:

| Value   | Description  | Reference     |
|---------|--------------|---------------|
| 0       | Reserved     | This document |
| 1- 250  | Unassigned   | IETF Review   |
| 251-253 | Experimental | This document |
| 254     | Private Use  | This document |
| 255     | Reserved     | This document |

Table 4: Extended BFD Return Codes Registry

This document defines the following new values in Extended BFD Return Codes registry:

| Value | Description                         | Reference     |
|-------|-------------------------------------|---------------|
| 0     | None                                | This document |
| 1     | One or more TLVs was not understood | This document |
| 2     | Lightweight Authentication failed   | This document |

Table 5: Extended BFD Return Codes

#### 5. Security Considerations

This document does not introduce new security aspects but inherits all security considerations from [RFC5880], [RFC6428], and [RFC6374].

## 6. References

### 6.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC5880] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD)", RFC 5880, DOI 10.17487/RFC5880, June 2010, <<https://www.rfc-editor.org/info/rfc5880>>.
- [RFC6374] Frost, D. and S. Bryant, "Packet Loss and Delay Measurement for MPLS Networks", RFC 6374, DOI 10.17487/RFC6374, September 2011, <<https://www.rfc-editor.org/info/rfc6374>>.
- [RFC6428] Allan, D., Ed., Swallow, G., Ed., and J. Drake, Ed., "Proactive Connectivity Verification, Continuity Check, and Remote Defect Indication for the MPLS Transport Profile", RFC 6428, DOI 10.17487/RFC6428, November 2011, <<https://www.rfc-editor.org/info/rfc6428>>.
- [RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <<https://www.rfc-editor.org/info/rfc8126>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.

### 6.2. Informative References

- [RFC8655] Finn, N., Thubert, P., Varga, B., and J. Farkas, "Deterministic Networking Architecture", RFC 8655, DOI 10.17487/RFC8655, October 2019, <<https://www.rfc-editor.org/info/rfc8655>>.

## Appendix A. Acknowledgements

TBD



Authors' Addresses

Greg Mirsky  
ZTE Corp.

Email: gregimirsky@gmail.com

Xiao Min  
ZTE Corp.

Email: xiao.min2@zte.com.cn

BFD Working Group  
Internet-Draft  
Intended status: Informational  
Expires: 8 September 2022

G. Mirsky  
Ericsson  
7 March 2022

BFD in Demand Mode over Point-to-Point MPLS LSP  
draft-mirsky-bfd-mpls-demand-11

Abstract

This document describes procedures for using Bidirectional Forwarding Detection (BFD) in Demand mode to detect data plane failures in Multiprotocol Label Switching (MPLS) point-to-point Label Switched Paths.

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 <https://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 8 September 2022.

Copyright Notice

Copyright (c) 2022 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 (<https://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 Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

## Table of Contents

|                                                                 |   |
|-----------------------------------------------------------------|---|
| 1. Introduction . . . . .                                       | 2 |
| 2. Conventions used in this document . . . . .                  | 2 |
| 2.1. Terminology . . . . .                                      | 2 |
| 3. Use of the BFD Demand Mode . . . . .                         | 2 |
| 3.1. The Applicability of BFD for Multipoint Networks . . . . . | 4 |
| 4. IANA Considerations . . . . .                                | 4 |
| 5. Security Considerations . . . . .                            | 4 |
| 6. Normative References . . . . .                               | 4 |
| 7. Informative References . . . . .                             | 5 |
| Appendix A. Acknowledgements . . . . .                          | 5 |
| Author's Address . . . . .                                      | 6 |

## 1. Introduction

[RFC5884] defined use of the Asynchronous method of Bidirectional Detection (BFD) [RFC5880] to monitor and detect failures in the data path of a Multiprotocol Label Switching (MPLS) Label Switched Path (LSP). Use of the Demand mode, also specified in [RFC5880], has not been defined so far. This document describes procedures for using the Demand mode of BFD protocol to detect data plane failures in MPLS point-to-point (p2p) LSPs.

## 2. Conventions used in this document

## 2.1. Terminology

MPLS: Multiprotocol Label Switching

LSP: Label Switched Path

LER: Label switching Edge Router

BFD: Bidirectional Forwarding Detection

p2p: Point-to-Point

## 3. Use of the BFD Demand Mode

[RFC5880] defines that the Demand mode may be:

- \* asymmetric, i.e. used in one direction of a BFD session;
- \* switched to and from without bringing BFD session to Down state through using a Poll Sequence.

For the case of BFD over MPLS LSP, ingress Label switching Edge Router (LER) usually acts as Active BFD peer and egress LER acts as Passive BFD peer. The Active peer bootstraps the BFD session by using LSP ping. If the BFD session is configured to use the Demand mode, once the BFD session is in Up state the ingress LER switches to the Demand mode as defined in Section 6.6 [RFC5880]. The egress LER also follows procedures defined in Section 6.6 [RFC5880] and ceases further transmission of periodic BFD control packets to the ingress LER.

In this state BFD peers remains as long as the egress LER is in Up state. The ingress LER can periodically check continuity of a bidirectional path between the ingress and egress LERs by using the Poll Sequence, as described in Section 6.6 [RFC5880]. An implementation that supports using the Poll Sequence as the mechanism for bidirectional path continuity check must control the interval between consecutive Poll Sequences. The Rdefault value could be selected as 1 second.

If the Detection timer at the egress LER expires, the BFD system on LER sends BFD Control packet to the ingress LER with the Poll (P) bit set, Status (Sta) field set to the Down value, and the Diagnostic (Diag) field set to Control Detection Time Expired value. The egress LER periodically transmits these Control packets to the ingress LER until either it receives the valid for this BFD session control packet with the Final (F) bit set from the ingress LER or the defect condition clears and the BFD session state reaches Up state at the egress LER. An implementation that supports this specification provides control of the interval between consecutive Poll messages signaling the expiration of the Detection timer. The default value of the interval can be selected as 1 second.

The ingress LER transmits BFD Control packets over the MPLS LSP with the Demand (D) flag set at negotiated interval per [RFC5880], the greater of `bfd.DesiredMinTxInterval` and `bfd.RemoteMinRxInterval`, until it receives the valid BFD packet from the egress LER with the Poll (P) bit and the Diagnostic (Diag) field value Control Detection Time Expired. Reception of such BFD control packet by the ingress LER indicates that the monitored LSP has a failure and sending BFD control packet with the Final flag set to acknowledge failure indication is likely to fail. Instead, the ingress LER transmits the BFD Control packet to the egress LER over the IP network with:

- \* destination IP address is set to the destination IP address of the LSP Ping Echo request message [RFC8029];
- \* destination UDP port set to 4784 [RFC5883];

- \* Final (F) flag in BFD control packet is set;
- \* Demand (D) flag in BFD control packet is cleared.

The ingress LER changes the state of the BFD session to Down and changes rate of BFD Control packets transmission to one packet per second. The ingress LER in Down mode changes to Asynchronous mode until the BFD session comes to Up state once again. Then the ingress LER switches to the Demand mode.

### 3.1. The Applicability of BFD for Multipoint Networks

[RFC8562] defines the use of BFD in multipoint networks. This specification analyzes the case of p2p LSP. In that scenario, the ingress of the LSP acts as the MultipointHead, and the egress - as MultipointTail. The BFD state machines for MultipointHead, MultipointClient, and MultipointTail don't use the three-way handshakes for session establishment and teardown. As a result, the Init state is absent, and the session transitions to the Up state once the BFD session is administratively enabled. Hence, a BFD session over a p2p LSP, using principles of [RFC8562] or [RFC8563], can be established faster if the MultipointTail has been provisioned with the value of My Discriminator used by the MultipointHead for that BFD session. That value can be provided to the MultipointTail using different mechanisms, e.g., an extension to IGP. Description of mechanism to provide the value of My Discriminator used by the MultipointHead for the particular BFD session is outside the scope of this specification.

Unsolicited notification of the detected failure by the MultipointTail to the MultipointClient performs as described above for the case when the ingress BFD system switches the remote peer into the Demand mode.

## 4. IANA Considerations

TBD

## 5. Security Considerations

This document does not introduce new security aspects but inherits all security considerations from [RFC5880], [RFC5884], [RFC7726], [RFC8029], and [RFC6425].

## 6. Normative References

- [RFC5880] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD)", RFC 5880, DOI 10.17487/RFC5880, June 2010, <<https://www.rfc-editor.org/info/rfc5880>>.
- [RFC5883] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD) for Multihop Paths", RFC 5883, DOI 10.17487/RFC5883, June 2010, <<https://www.rfc-editor.org/info/rfc5883>>.
- [RFC5884] Aggarwal, R., Kompella, K., Nadeau, T., and G. Swallow, "Bidirectional Forwarding Detection (BFD) for MPLS Label Switched Paths (LSPs)", RFC 5884, DOI 10.17487/RFC5884, June 2010, <<https://www.rfc-editor.org/info/rfc5884>>.
- [RFC6425] Saxena, S., Ed., Swallow, G., Ali, Z., Farrel, A., Yasukawa, S., and T. Nadeau, "Detecting Data-Plane Failures in Point-to-Multipoint MPLS - Extensions to LSP Ping", RFC 6425, DOI 10.17487/RFC6425, November 2011, <<https://www.rfc-editor.org/info/rfc6425>>.
- [RFC7726] Govindan, V., Rajaraman, K., Mirsky, G., Akiya, N., and S. Aldrin, "Clarifying Procedures for Establishing BFD Sessions for MPLS Label Switched Paths (LSPs)", RFC 7726, DOI 10.17487/RFC7726, January 2016, <<https://www.rfc-editor.org/info/rfc7726>>.
- [RFC8029] Kompella, K., Swallow, G., Pignataro, C., Ed., Kumar, N., Aldrin, S., and M. Chen, "Detecting Multiprotocol Label Switched (MPLS) Data-Plane Failures", RFC 8029, DOI 10.17487/RFC8029, March 2017, <<https://www.rfc-editor.org/info/rfc8029>>.
- [RFC8562] Katz, D., Ward, D., Pallagatti, S., Ed., and G. Mirsky, Ed., "Bidirectional Forwarding Detection (BFD) for Multipoint Networks", RFC 8562, DOI 10.17487/RFC8562, April 2019, <<https://www.rfc-editor.org/info/rfc8562>>.

## 7. Informative References

- [RFC8563] Katz, D., Ward, D., Pallagatti, S., Ed., and G. Mirsky, Ed., "Bidirectional Forwarding Detection (BFD) Multipoint Active Tails", RFC 8563, DOI 10.17487/RFC8563, April 2019, <<https://www.rfc-editor.org/info/rfc8563>>.

## Appendix A. Acknowledgements

TBD

Author's Address

Greg Mirsky  
Ericsson  
Email: gregimirsky@gmail.com

BFD Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: May 27, 2019

X. Min  
G. Mirsky  
ZTE  
November 23, 2018

BFD for Geneve  
draft-xiao-bfd-geneve-00

Abstract

This document describes the use of the Bidirectional Forwarding Detection (BFD) protocol in Generic Network Virtualization Encapsulation (Geneve) overlay networks.

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 <https://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 May 27, 2019.

Copyright Notice

Copyright (c) 2018 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 (<https://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  |
| 1.1. Conventions Used in This Document . . . . .         | 2  |
| 1.1.1. Terminology . . . . .                             | 2  |
| 1.1.2. Requirements Language . . . . .                   | 3  |
| 2. BFD Packet Transmission over Geneve Tunnel . . . . .  | 3  |
| 2.1. BFD Packet Encapsulation in Geneve . . . . .        | 3  |
| 2.1.1. BFD Encapsulation With IP/UDP Header . . . . .    | 3  |
| 2.1.2. BFD Encapsulation Without IP/UDP Header . . . . . | 5  |
| 3. Reception of BFD packet from Geneve Tunnel . . . . .  | 7  |
| 3.1. Demultiplexing of the BFD packet . . . . .          | 8  |
| 4. Security Considerations . . . . .                     | 8  |
| 5. IANA Considerations . . . . .                         | 9  |
| 6. Acknowledgements . . . . .                            | 9  |
| 7. Normative References . . . . .                        | 9  |
| Authors' Addresses . . . . .                             | 10 |

## 1. Introduction

"Generic Network Virtualization Encapsulation" (Geneve) [I-D.ietf-nvo3-geneve] provides a generic tunneling protocol that is applicable to many scenarios, including an encapsulation scheme that allows virtual machines (VMs) to communicate in a data center network.

This document describes the use of Bidirectional Forwarding Detection (BFD) protocol for Geneve to enable monitoring continuity of the path between Network Virtualization Edges (NVEs) and/or availability of a replicator service node using BFD.

The use cases and the deployment of BFD for Geneve are consistent with what's described in Section 3 and Section 4 of [I-D.ietf-bfd-vxlan]. The main difference between Geneve and "Virtual eXtensible Local Area Network" (VXLAN) [RFC7348] encapsulation is that Geneve supports multi-protocol payload and variable length options.

## 1.1. Conventions Used in This Document

## 1.1.1. Terminology

BFD: Bidirectional Forwarding Detection

Geneve: Generic Network Virtualization Encapsulation

NVE: Network Virtualization Edge

VFI: Virtual Forwarding Instance

VM: Virtual Machine

VNI: Virtual Network Identifier

VXLAN: Virtual eXtensible Local Area Network

#### 1.1.2. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

### 2. BFD Packet Transmission over Geneve Tunnel

BFD packet MUST be encapsulated and sent to a remote NVE using one of the options described in Section 2.1. Implementations SHOULD ensure that the BFD packets follow the same lookup path as Geneve data packets within the sender system.

#### 2.1. BFD Packet Encapsulation in Geneve

Concerning whether or not the Geneve data packets include an IP protocol data unit, this document defines three options of BFD packet encapsulation in Geneve.

##### 2.1.1. BFD Encapsulation With IP/UDP Header

If the Protocol Type field (as defined in Section 3.4 of [I-D.ietf-nvo3-geneve]) of data packets indicates that there exists an inner IP header, i.e., the Protocol Type equals to 0x6558 (Ethernet frame), or 0x0800 (IPv4), or 0x86DD (IPv6), or 0x8847 (MPLS), or 0x8848 (MPLS with the upstream-assigned label), then BFD packets are encapsulated in Geneve as described below. The Geneve packet format over IPv4 is defined in Section 3.1 of [I-D.ietf-nvo3-geneve]. The Geneve packet format over IPv6 is defined in Section 3.2 of [I-D.ietf-nvo3-geneve]. The Outer IP/UDP and Geneve headers MUST be encoded by the sender as defined in [I-D.ietf-nvo3-geneve]. Note that the outer IP header and the inner IP header may not be of the same address family, in other words, outer IPv6 header accompanied with inner IPv4 header and outer IPv4 header accompanied with inner IPv6 header are both possible.

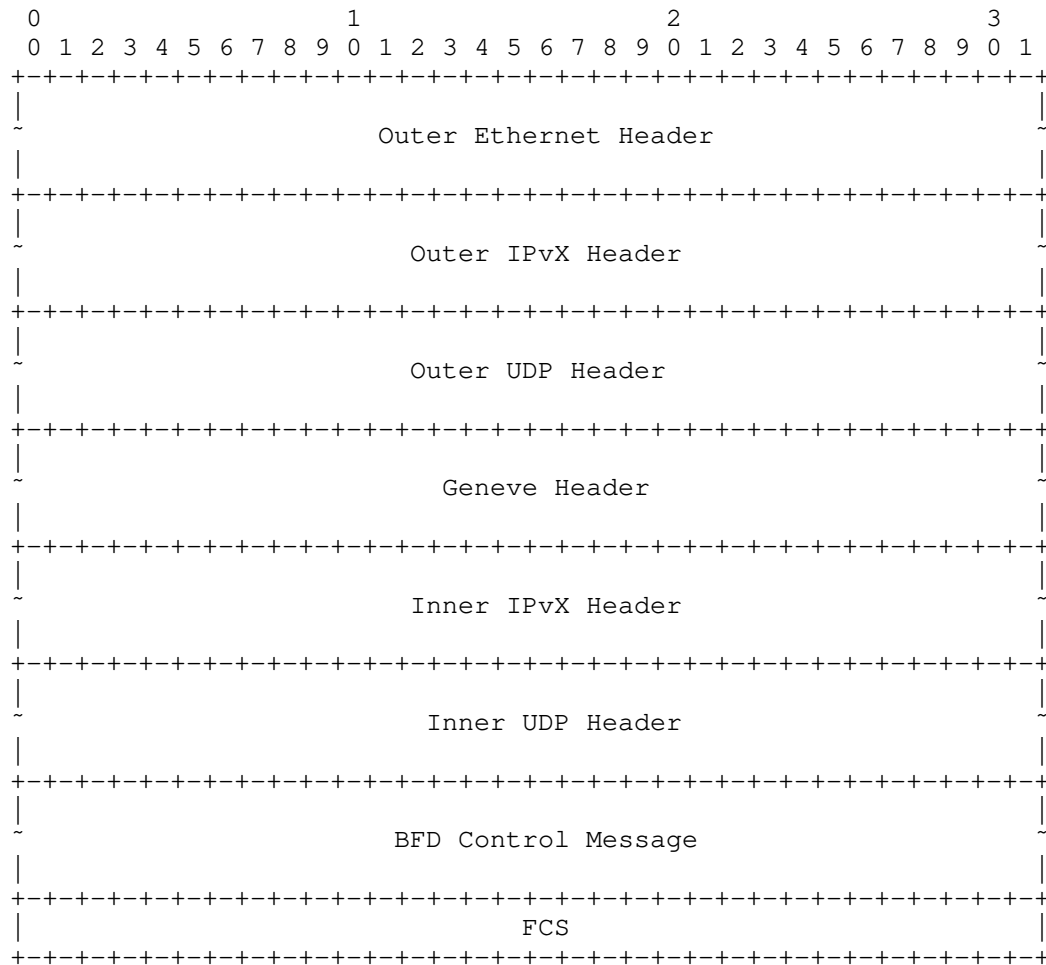


Figure 1: Geneve Encapsulation of BFD Control Message With the Inner IP/UDP Header

When the BFD packets are encapsulated in Geneve in this way, the BFD packet MUST be carried inside the inner IP packet of the Geneve packet. The inner IP packet carrying the BFD payload has the following format:

IP header:

Source IP: IP address of the originating NVE.

Destination IP: IP address of the terminating NVE.

TTL: MUST be set to 1 to ensure that the BFD packet is not routed within the L3 underlay network.

The fields of the UDP header and the BFD control packet are encoded as specified in [RFC5881].

When the BFD packets are encapsulated in Geneve in this way, the Geneve header SHOULD follow the value set below.

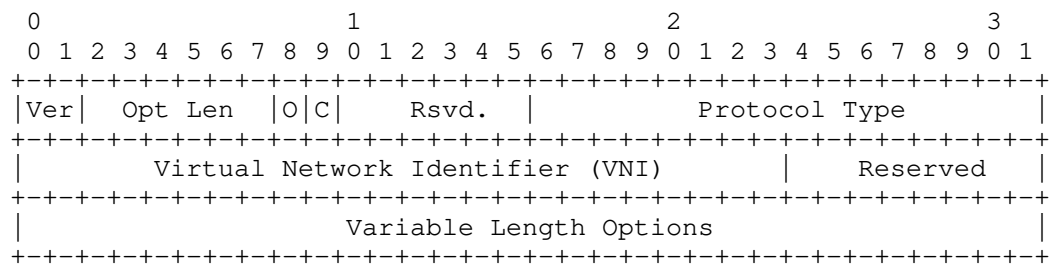


Figure 2: Geneve Header

Opt Len field SHOULD be set to 0, which indicates there isn't any variable length option.

[Ed.Note]: Use of O bit is still being discussed in the NVO3 WG, so the value is undetermined.

C bit SHOULD be set to 0.

Protocol Type field SHOULD be set to 0x0800 (IPv4) or 0x86DD (IPv6).

#### 2.1.2. BFD Encapsulation Without IP/UDP Header

Alternatively to the use of the inner IP/UDP header to demultiplex BFD control packet by the value of the destination UDP port, BFD control packet MAY be encapsulated without the inner IP/UDP header. The BFD control packet MAY be identified directly in the Geneve header or through Geneve OAM shim. In either case, the Outer IP/UDP and Geneve headers MUST be encoded by the sender as defined in [I-D.ietf-nvo3-geneve].

Figure 3 displays the layout of the Ethernet frame with BFD control packet encapsulated in Geneve without the use of IP/UDP header and identified by the value TBA1 (to be assigned by IANA) of the Protocol Type field.

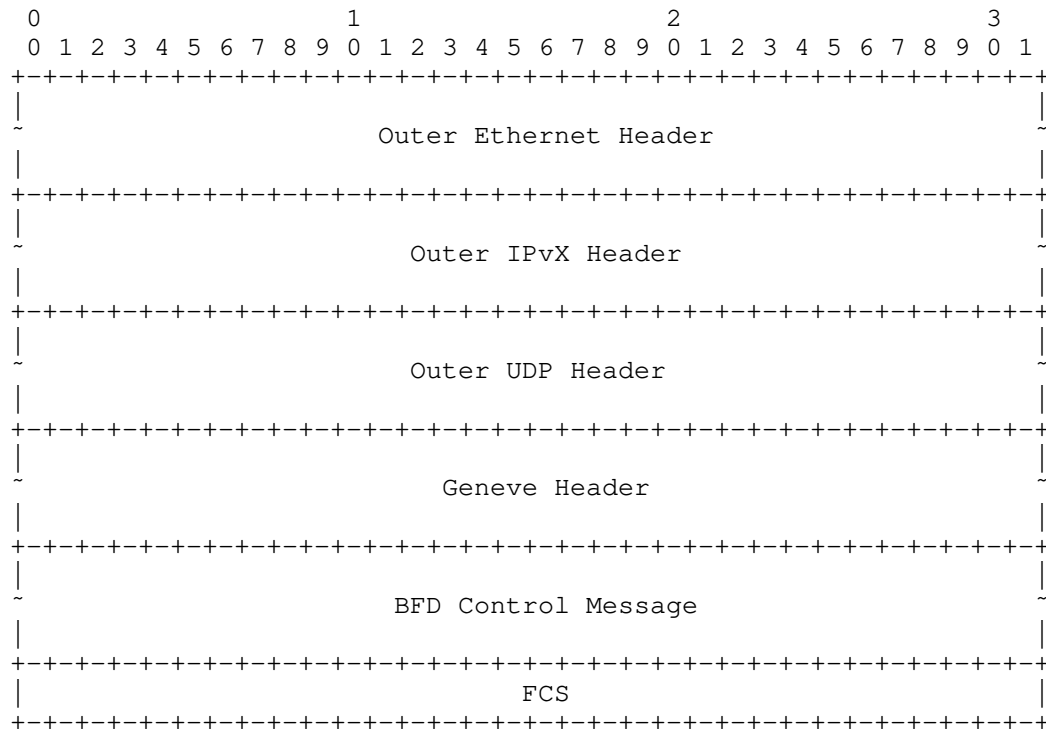


Figure 3: Geneve Encapsulation of BFD Control Message Without the Inner IP/UDP Header

When the BFD packets are encapsulated in Geneve in this way, the BFD packet MUST immediately follow the Geneve header, and the Geneve header SHOULD follow the value set below.

Opt Len field SHOULD be set to 0, which indicates there isn't any variable length option.

[Ed.Note]: Use of O bit is still being discussed in the NVO3 WG, so the value is undetermined.

C bit SHOULD be set to 0.

Also, if BFD control packet is encapsulated in Geneve without the use of IP/UDP header, the BFD control packet MAY be identified through the Geneve OAM shim. The layout of the Ethernet frame is shown in Figure 4. Protocol Type field MUST be set to the value TBA2 (to be assigned by IANA) which indicates a Geneve OAM shim that will have a field to indicate the inner BFD control packet. Definition of the

format of the Geneve OAM shim is outside the scope of this document. The Geneve OAM shim immediately follows the Geneve header, and the BFD control packet immediately follows the Geneve OAM shim.

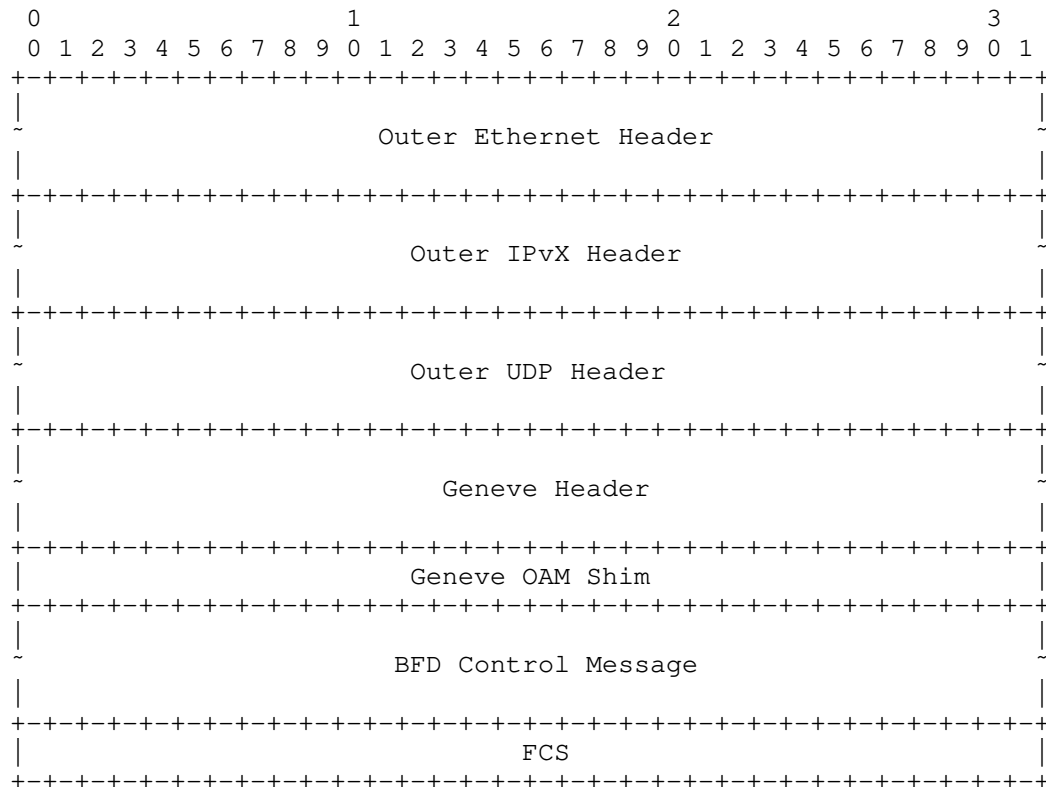


Figure 4: Geneve Encapsulation of BFD Control Message With Geneve OAM Shim

### 3. Reception of BFD packet from Geneve Tunnel

Once a packet is received, NVE MUST validate the packet as described in [I-D.ietf-nvo3-geneve].

If the Protocol Type field equals 0x0800 (IPv4) or 0x86DD (IPv6), and the Destination IP of the inner IP packet matches the IP address of the NVE, the UDP destination port and the TTL of the inner IP packet MUST be validated to determine whether BFD can process the received packet. BFD packet with inner IP set to NVE MUST NOT be forwarded to VMs.

If the Protocol Type field equals the value TBA1 (to be assigned by IANA) which indicates an inner BFD control message, the received packet MUST be processed by BFD and MUST NOT be forwarded to VMs.

If the Protocol Type field equals the value TBA2 (to be assigned by IANA) which indicates a Geneve OAM shim that will have a field to indicate the inner BFD control message, the received packet MUST be processed by BFD and MUST NOT be forwarded to VMs. This case is for further study.

To ensure BFD detects the proper configuration of Virtual Network Identifier (VNI) in a remote NVE, a lookup SHOULD be performed with the MAC-DA/IP-DA/MPLS-Label and VNI as key in the Virtual Forwarding Instance (VFI) table of the originating/terminating NVE to exercise the VFI associated with the VNI.

### 3.1. Demultiplexing of the BFD packet

If the Protocol Type field equals 0x0800 (IPv4) or 0x86DD (IPv6), demultiplexing of IP BFD packet has been defined in Section 3 of [RFC5881]. Since multiple BFD sessions may be running between two NVEs, there needs to be a mechanism for demultiplexing received BFD packets to the proper session. The procedure for demultiplexing packets with Your Discriminator equal to 0 is different from [RFC5880]. For such packets, the BFD session MUST be identified using the inner headers, i.e., the source IP and the destination IP present in the IP header carried by the payload of the Geneve encapsulated packet. The VNI of the packet SHOULD be used to derive interface-related information for demultiplexing the packet. If BFD packet is received with non-zero Your Discriminator, then BFD session MUST be demultiplexed only with Your Discriminator as the key.

If the Protocol Type field equals the value TBA1 (to be assigned by IANA) which indicates an inner BFD control message, or the value TBA2 (to be assigned by IANA) which indicates a Geneve OAM shim that will have a field to indicate the inner BFD control message, the VNI of the packet SHOULD be used to derive interface-related information for demultiplexing the packet, demultiplexing of BFD packet MUST rely on non-zero Your Discriminator as the key.

## 4. Security Considerations

This document does not raise any additional security issues beyond those of the specifications referred to in the list of normative references.

## 5. IANA Considerations

In the Geneve Protocol Type registry defined in [ETYPES], a new BFD Control Message or Geneve OAM Shim is requested from IANA as follows:

| Geneve Protocol Type | Description         | Semantics Definition | Reference     |
|----------------------|---------------------|----------------------|---------------|
| TBA1                 | BFD Control Message | Section 3.1          | This Document |
| TBA2                 | Geneve OAM Shim     | Section 3.1          | This Document |

Table 1: New BFD Control Message or Geneve OAM shim Ethertype

## 6. Acknowledgements

To be added.

## 7. Normative References

- [ETYPES] The IEEE Registration Authority, "IEEE 802 Numbers", 2013, <<http://www.iana.org/assignments/ieee-802-numbers/ieee-802-numbers.xml>>.
- [I-D.ietf-bfd-vxlan] Networks, J., Paragiri, S., Govindan, V., Mudigonda, M., and G. Mirsky, "BFD for VXLAN", draft-ietf-bfd-vxlan-03 (work in progress), October 2018.
- [I-D.ietf-nvo3-geneve] Gross, J., Ganga, I., and T. Sridhar, "Geneve: Generic Network Virtualization Encapsulation", draft-ietf-nvo3-geneve-08 (work in progress), October 2018.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC5880] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD)", RFC 5880, DOI 10.17487/RFC5880, June 2010, <<https://www.rfc-editor.org/info/rfc5880>>.



- [RFC5881] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD) for IPv4 and IPv6 (Single Hop)", RFC 5881, DOI 10.17487/RFC5881, June 2010, <<https://www.rfc-editor.org/info/rfc5881>>.
- [RFC7348] Mahalingam, M., Dutt, D., Duda, K., Agarwal, P., Kreeger, L., Sridhar, T., Bursell, M., and C. Wright, "Virtual eXtensible Local Area Network (VXLAN): A Framework for Overlaying Virtualized Layer 2 Networks over Layer 3 Networks", RFC 7348, DOI 10.17487/RFC7348, August 2014, <<https://www.rfc-editor.org/info/rfc7348>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.

## Authors' Addresses

Xiao Min  
ZTE  
Nanjing  
China

Phone: +86 25 88016574  
Email: [xiao.min2@zte.com.cn](mailto:xiao.min2@zte.com.cn)

Greg Mirsky  
ZTE  
USA

Email: [gregimirsky@gmail.com](mailto:gregimirsky@gmail.com)