Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: July 15, 2018

R. Rahman, Ed.
Cisco Systems
L. Zheng, Ed.
Huawei Technologies
M. Jethanandani, Ed.
Cisco Systems
S. Pallagatti

G. Mirsky
ZTE Corporation

January 11, 2018

YANG Data Model for Bidirectional Forwarding Detection (BFD)
draft-ietf-bfd-yang-08.txt

Abstract

This document defines a YANG data model that can be used to configure and manage Bidirectional Forwarding Detection (BFD).

Requirements Language

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

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 July 15, 2018.

Table of Contents

1. Introduction

This document defines a YANG data model that can be used to configure and manage Bidirectional Forwarding Detection (BFD) [RFC5880]. BFD is a network protocol which is used for liveness detection of arbitrary paths between systems. Some examples of different types of paths over which we have BFD:

1) Two systems directly connected via IP. This is known as BFD over single-hop IP, a.k.a. BFD for IPv4 and IPv6 [RFC5881]

2) Two systems connected via multiple hops as described in BFD for Multiple Hops. [RFC5883]

3) Two systems connected via MPLS Label Switched Paths (LSPs) as described in BFD for MPLS LSP [RFC5884]

4) Two systems connected via a Link Aggregation Group (LAG) interface as described in BFD on LAG Interfaces [RFC7130]

5) Two systems connected via pseudowires (PWs), this is known as Virtual Circuit Connectivity Verification (VCCV) as described in BFD for PW VCCV [RFC5885]. This is not addressed in this document.

BFD typically does not operate on its own. Various control protocols, also known as BFD clients, use the services provided by

BFD for their own operation as described in Generic Application of BFD [RFC5882]. The obvious candidates which use BFD are those which do not have "hellos" to detect failures, e.g. static routes, and routing protocols whose "hellos" do not support sub-second failure detection, e.g. OSPF and IS-IS.

2. Design of the Data Model

Since BFD is used for liveliness detection of various forwarding paths, there is no uniform key to identify a BFD session. So the BFD data model is split in multiple YANG modules where each module corresponds to one type of forwarding path. For example, BFD for IP single-hop is in one YANG module and BFD for MPLS-TE is in another YANG module. The main difference between these modules is how a BFD session is uniquely identified, i.e the key for the list containing the BFD sessions for that forwarding path. To avoid duplication of BFD definitions, we have common types and groupings which are used by all the modules.

A new control-plane protocol "bfdv1" is defined and a "bfd" container is created under control-plane-protocol as specified in A YANG Data Model for Routing Management [RFC8022]. This new "bfd" node is augmented by all the YANG modules for their respective specific information.

BFD can operate in the following contexts:

1. Network devices as described in Network Device YANG Organizational Models [I-D.ietf-rtgwg-device-model]

2. Logical Network Elements as described in YANG Logical Network Element [I-D.ietf-rtgwg-lne-model]

3. Network instances as described in YANG Logical Network Element [I-D.ietf-rtgwg-ni-model]

The approach taken is to do a schema-mount (see Schema Mount [I-D.ietf-netmod-schema-mount]) of the BFD model in the appropriate locations. For example, if an implementation supports BFD IP multi-hop in network instances, the implementation would do schema-mount of the BFD IP multi-hop model in a mount-point which resides in a network instance.

The data models in this document strive to follow the "Network Management Datastore Architecture" (NMDA) guidelines described in [I-D.dsdt-nmda-guidelines]. This means that the data models do NOT have separate top-level or sibling containers for configuration and operational data.

2.1. Design of Configuration Model

The configuration model consists mainly of the parameters specified in BFD [RFC5880]. Some examples are desired minimum transmit interval, required minimum receive interval, detection multiplier, etc

BFD clients are applications that use BFD for fast detection of failures. Some implementations have BFD session configuration under the BFD clients. For example, BFD session configuration under routing applications such as OSPF, IS-IS, BGP etc. Other implementations have BFD session configuration centralized under BFD, i.e. outside the multiple BFD clients.

The BFD parameters of interest to a BFD client are mainly the multiplier and interval(s) since those parameters impact the convergence time of the BFD clients when a failure occurs. Other parameters such as BFD authentication are not specific to the requirements of the BFD client. Ideally all configuration should be centralized under BFD. However, this is a problem for clients of BFD which auto-discover their peers. For example, IGPs do not have the peer address configured, instead the IGP is enabled on an interface and the IGP peers are auto-discovered. So for an operator to configure BFD to an IGP peer, the operator would first have to determine the peer addresses. And when a new peer is discovered, BFD configuration would need to be added. To avoid this issue, we define grouping client-cfg-parms in Section 2.13 for BFD clients to configure BFD: this allows BFD clients such as the IGPs to have configuration (multiplier and intervals) for the BFD sessions they need. The mechanism how the BFD sessions are created by the BFD clients is outside the scope of this document. For BFD clients which create BFD sessions via their own configuration, authentication parameters (if required) are still specified in BFD.

2.1.1. Common BFD configuration parameters

The basic BFD configuration parameters are:

local-multiplier
: This is the detection time multiplier as defined in BFD [RFC5880].

desired-min-tx-interval
: This is the Desired Min TX Interval as defined in BFD [RFC5880].

required-min-rx-interval

This is the Required Min RX Interval as defined in BFD [RFC5880].

Although BFD [RFC5880] allows for different values for transmit and receive intervals, some implementations allow users to specify just one interval which is used for both transmit and receive intervals or separate values for transmit and receive intervals. The BFD YANG model supports this: there is a choice between "min-interval", used for both transmit and receive intervals, and "desired-min-tx-interval" and "required-min-rx-interval". This is supported via a grouping which is used by the YANG modules for the various forwarding paths.

For BFD authentication we have:

key-chain
: This is a reference to key-chain defined in YANG Data Model for Key Chains [RFC8177]. The keys, cryptographic algorithms, key lifetime etc are all defined in the key-chain model.

meticulous
: This enables meticulous mode as per BFD [RFC5880].

2.1.2. Single-hop IP

For single-hop IP, there is an augment of the "bfd" data node in Section 2. The "ip-sh" node contains a list of IP single-hop sessions where each session is uniquely identified by the interface and destination address pair. For the configuration parameters we use what is defined in Section 2.1.1. The "ip-sh" node also contains a list of interfaces, this is used to specify authentication parameters for BFD sessions which are created by BFD clients, see Section 2.1.

[RFC5880] and [RFC5881] do not specify whether echo function is continuous or on demand. Therefore the mechanism used to start and stop echo function is implementation specific and should be done by augmentation:

1) Configuration. This is suitable for continuous echo function. An example is provided in Appendix A.

2) RPC. This is suitable for on-demand echo function.

2.1.3. Multi-hop IP

For multi-hop IP, there is an augment of the "bfd" data node in Section 2.

Because of multiple paths, there could be multiple multi-hop IP sessions between a source and a destination address. We identify this as a "session-group". The key for each "session-group" consists of:

source address
: Address belonging to the local system as per BFD for Multiple Hops [RFC5883]

destination address
: Address belonging to the remote system as per BFD for Multiple Hops [RFC5883]

For the configuration parameters we use what is defined in Section 2.1.1

Here are some extra parameters:

tx-ttl
: TTL of outgoing BFD control packets.

rx-ttl
: Minimum TTL of incoming BFD control packets.

2.1.4. MPLS Traffic Engineering Tunnels

For MPLS-TE tunnels, BFD is configured under the MPLS-TE tunnel since the desired failure detection parameters is a property of the MPLS-TE tunnel. This is achieved by augmenting the MPLS-TE data model in YANG Data Model for TE Topologies [I-D.ietf-teas-yang-te]. For BFD parameters which are specific to the TE application, e.g. whether to tear down the tunnel in the event of a BFD session failure, these parameters will be defined in the YANG model of the MPLS-TE application.

On top of the usual BFD parameters, we have the following per MPLS-TE tunnel:

encap
: Encapsulation for the BFD packets: choice between IP, G-ACh and IP with G-ACh as per MPLS Generic Associated Channel [RFC5586]

For general MPLS-TE data, "mpls-te" data node is added under the "bfd" node in Section 2. Since some MPLS-TE tunnels are uni-directional there is no MPLS-TE configuration for these tunnels on the egress node (note that this does not apply to bi-directional MPLS-TP tunnels). The BFD parameters for the egress node are added under "mpls-te".

2.1.5. MPLS Label Switched Paths

Here we address MPLS LSPs whose FEC is an IP address. The "bfd" node in Section 2 is augmented with "mpls" which contains a list of sessions uniquely identified by an IP prefix. Because of multiple paths, there could be multiple MPLS sessions to an MPLS FEC. We identify this as a "session-group".

Since these LSPs are uni-directional there is no LSP configuration on the egress node.

The BFD parameters for the egress node are added under "mpls".

2.1.6. Link Aggregation Groups

Per BFD on LAG Interfaces [RFC7130], configuring BFD on LAG consists of having micro-BFD sessions on each LAG member link. Since the BFD parameters are an attribute of the LAG, they should be under the LAG. However there is no LAG YANG model which we can augment. So a "lag" data node is added to the "bfd" node in Section 2, the configuration is per-LAG: we have a list of LAGs. The destination IP address of the micro-BFD sessions is configured per-LAG and per address-family (IPv4 and IPv6)

2.2. Design of Operational Model

The operational model contains both the overall statistics of BFD sessions running on the device and the per session operational information.

The overall statistics of BFD sessions consist of number of BFD sessions, number of BFD sessions up etc. This information is available globally (i.e. for all BFD sessions) under the "bfd" node in Section 2 and also per type of forwarding path.

For each BFD session, mainly three categories of operational items are shown. The fundamental information of a BFD session such as the local discriminator, remote discriminator and the capability of supporting demand detect mode are shown in the first category. The second category includes a BFD session running information, e.g. the remote BFD state and the diagnostic code received. Another example

is the actual transmit interval between the control packets, which may be different from the desired minimum transmit interval configured, is shown in this category. Similar examples are actual received interval between the control packets and the actual transmit interval between the echo packets. The third category contains the detailed statistics of the session, e.g. when the session transitioned up/down and how long it has been in that state.

For some path types, there may be more than 1 session on the virtual path to the destination. For example, with IP multi-hop and MPLS LSPs, there could be multiple BFD sessions from the source to the same destination to test the various paths (ECMP) to the destination. This is represented by having multiple "sessions" under each "session-group".

2.3. Notifications

This YANG model defines notifications to inform clients of BFD of important events detected during the protocol operation. Pair of local and remote discriminator identifies a BFD session on local system. Notifications also give more important details about BFD sessions; e.g. new state, time in previous state, network-instance and the reason that the BFD session state changed. The notifications are defined for each type of forwarding path but use groupings for common information.

2.4. RPC Operations

None.

2.5. BFD top level hierarchy

At the "bfd" node under control-plane-protocol, there is no configuration data, only operational data. The operational data consist of overall BFD session statistics, i.e. for BFD on all types of forwarding paths. The "bfd" node under control-plane-protocol can be used in a network device (top-level), or mounted in an LNE or in a network instance.

```
module: ietf-bfd
  augment /rt:routing/rt:control-plane-protocols
          /rt:control-plane-protocol:
    +--rw bfd
       +--ro session-statistics
          +--ro session-count?              uint32
          +--ro session-up-count?           uint32
          +--ro session-down-count?         uint32
          +--ro session-admin-down-count?   uint32
```

2.6. BFD IP single-hop hierarchy

An "ip-sh" node is added under "bfd" node in control-plane-protocol. The configuration and operational data for each BFD IP single-hop session is under this "ip-sh" node. The "ip-sh" node can be used in a network device (top-level), or mounted in an LNE or in a network instance.

```
module: ietf-bfd-ip-sh
  augment /rt:routing/rt:control-plane-protocols
          /rt:control-plane-protocol/bfd:bfd:
    +--rw ip-sh
       +--ro session-statistics
       |  +--ro session-count?              uint32
       |  +--ro session-up-count?           uint32
       |  +--ro session-down-count?         uint32
       |  +--ro session-admin-down-count?   uint32
       +--rw sessions* [interface dest-addr]
       |  +--rw interface                   if:interface-ref
       |  +--rw dest-addr                   inet:ip-address
       |  +--rw source-addr?                inet:ip-address
       |  +--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)
       |  |     +--rw min-interval?               uint32
       |  +--rw demand-enabled?             boolean {demand-mode}?
       |  +--rw admin-down?                 boolean
       |  +--rw authentication-parms! {authentication}?
       |  |  +--rw key-chain?    kc:key-chain-ref
       |  |  +--rw meticulous?   boolean
       |  +--ro path-type?                  identityref
       |  +--ro ip-encapsulation?           boolean
       |  +--ro local-discriminator?        discriminator
       |  +--ro remote-discriminator?       discriminator
       |  +--ro remote-multiplier?          multiplier
       |  +--ro demand-capability?          boolean {demand-mode}?
       |  +--ro source-port?                inet:port-number
       |  +--ro dest-port?                  inet:port-number
       |  +--ro session-running
       |  |  +--ro session-index?               uint32
       |  |  +--ro local-state?                 state
       |  |  +--ro remote-state?                state
       |  |  +--ro local-diagnostic?
       |  |  |       iana-bfd-types:diagnostic
       |  |  +--ro remote-diagnostic?
```

```
        |  |  |       iana-bfd-types:diagnostic
        |  |  +--ro remote-authenticated?         boolean
        |  |  +--ro remote-authentication-type?   iana-bfd-types:auth-ty
 pe
        |  |  |       {authentication}?
        |  |  +--ro detection-mode?               enumeration
        |  |  +--ro negotiated-tx-interval?       uint32
        |  |  +--ro negotiated-rx-interval?       uint32
        |  |  +--ro detection-time?               uint32
        |  |  +--ro echo-tx-interval-in-use?      uint32 {echo-mode}?
        |  +--ro sesssion-statistics
        |     +--ro create-time?            yang:date-and-time
        |     +--ro last-down-time?         yang:date-and-time
        |     +--ro last-up-time?           yang:date-and-time
        |     +--ro down-count?             uint32
        |     +--ro admin-down-count?       uint32
        |     +--ro receive-packet-count?   uint64
        |     +--ro send-packet-count?      uint64
        |     +--ro receive-bad-packet?     uint64
        |     +--ro send-failed-packet?     uint64
        +--rw interfaces* [interface]
           +--rw interface               if:interface-ref
           +--rw authentication-parms! {authentication}?
              +--rw key-chain?    kc:key-chain-ref
              +--rw meticulous?   boolean

   notifications:
     +---n singlehop-notification
        +--ro local-discr?                 discriminator
        +--ro remote-discr?                discriminator
        +--ro new-state?                   state
        +--ro state-change-reason?         iana-bfd-types:diagnostic
        +--ro time-of-last-state-change?   yang:date-and-time
        +--ro dest-addr?                   inet:ip-address
        +--ro source-addr?                 inet:ip-address
        +--ro session-index?               uint32
        +--ro path-type?                   identityref
        +--ro interface?                   if:interface-ref
        +--ro echo-enabled?                boolean
```

2.7. BFD IP multi-hop hierarchy

An "ip-mh" node is added under the "bfd" node in cntrol-plane-protocol. The configuration and operational data for each BFD IP multi-hop session is under this "ip-mh" node. In the operational model we support multiple BFD multi-hop sessions per remote address (ECMP), the local discriminator is used as key. The "ip-mh" node can

be used in a network device (top-level), or mounted in an LNE or in a network instance.

```
module: ietf-bfd-ip-mh
  augment /rt:routing/rt:control-plane-protocols
          /rt:control-plane-protocol/bfd:bfd:
    +--rw ip-mh
       +--ro session-statistics
       |  +--ro session-count?              uint32
       |  +--ro session-up-count?           uint32
       |  +--ro session-down-count?         uint32
       |  +--ro session-admin-down-count?   uint32
       +--rw session-group* [source-addr dest-addr]
          +--rw source-addr                 inet:ip-address
          +--rw dest-addr                   inet:ip-address
          +--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)
          |     +--rw min-interval?               uint32
          +--rw demand-enabled?             boolean {demand-mode}?
          +--rw admin-down?                 boolean
          +--rw authentication-parms! {authentication}?
          |  +--rw key-chain?    kc:key-chain-ref
          |  +--rw meticulous?   boolean
          +--rw tx-ttl?                     bfd-types:hops
          +--rw rx-ttl                      bfd-types:hops
          +--ro sessions*
             +--ro path-type?               identityref
             +--ro ip-encapsulation?        boolean
             +--ro local-discriminator?     discriminator
             +--ro remote-discriminator?    discriminator
             +--ro remote-multiplier?       multiplier
             +--ro demand-capability?       boolean {demand-mode}?
             +--ro source-port?             inet:port-number
             +--ro dest-port?               inet:port-number
             +--ro session-running
             |  +--ro session-index?                uint32
             |  +--ro local-state?                  state
             |  +--ro remote-state?                 state
             |  +--ro local-diagnostic?
             |  |       iana-bfd-types:diagnostic
             |  +--ro remote-diagnostic?
             |  |       iana-bfd-types:diagnostic
             |  +--ro remote-authenticated?         boolean
             |  +--ro remote-authentication-type?
```

```
            |  |       iana-bfd-types:auth-type {authentication}?
            |  +--ro detection-mode?               enumeration
            |  +--ro negotiated-tx-interval?       uint32
            |  +--ro negotiated-rx-interval?       uint32
            |  +--ro detection-time?               uint32
            |  +--ro echo-tx-interval-in-use?      uint32
            |          {echo-mode}?
            +--ro sesssion-statistics
               +--ro create-time?            yang:date-and-time
               +--ro last-down-time?         yang:date-and-time
               +--ro last-up-time?           yang:date-and-time
               +--ro down-count?             uint32
               +--ro admin-down-count?       uint32
               +--ro receive-packet-count?   uint64
               +--ro send-packet-count?      uint64
               +--ro receive-bad-packet?     uint64
               +--ro send-failed-packet?     uint64

  notifications:
    +---n multihop-notification
       +--ro local-discr?                discriminator
       +--ro remote-discr?               discriminator
       +--ro new-state?                  state
       +--ro state-change-reason?        iana-bfd-types:diagnostic
       +--ro time-of-last-state-change?  yang:date-and-time
       +--ro dest-addr?                  inet:ip-address
       +--ro source-addr?                inet:ip-address
       +--ro session-index?              uint32
       +--ro path-type?                  identityref
```

2.8. BFD over LAG hierarchy

A "lag" node is added under the "bfd" node in control-plane-protocol. The configuration and operational data for each BFD LAG session is under this "lag" node. The "lag" node can be used in a network device (top-level), or mounted in an LNE or in a network instance.

```
module: ietf-bfd-lag
  augment /rt:routing/rt:control-plane-protocols
          /rt:control-plane-protocol/bfd:bfd:
    +--rw lag
       +--rw micro-bfd-ipv4-session-statistics
       |  +--ro session-statistics
       |     +--ro session-count?              uint32
       |     +--ro session-up-count?           uint32
       |     +--ro session-down-count?         uint32
       |     +--ro session-admin-down-count?   uint32
       +--rw micro-bfd-ipv6-session-statistics
```

```
|  +--ro session-statistics
|     +--ro session-count?              uint32
|     +--ro session-up-count?           uint32
|     +--ro session-down-count?         uint32
|     +--ro session-admin-down-count?   uint32
+--rw sessions* [lag-name]
   +--rw lag-name                    if:interface-ref
   +--rw ipv4-dest-addr?             inet:ipv4-address
   +--rw ipv6-dest-addr?             inet:ipv6-address
   +--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)
   |     +--rw min-interval?               uint32
   +--rw demand-enabled?             boolean {demand-mode}?
   +--rw admin-down?                 boolean
   +--rw authentication-parms! {authentication}?
   |  +--rw key-chain?    kc:key-chain-ref
   |  +--rw meticulous?   boolean
   +--rw use-ipv4?                   boolean
   +--rw use-ipv6?                   boolean
   +--ro member-links* [member-link]
      +--ro member-link       if:interface-ref
      +--ro micro-bfd-ipv4
      |  +--ro path-type?              identityref
      |  +--ro ip-encapsulation?       boolean
      |  +--ro local-discriminator?    discriminator
      |  +--ro remote-discriminator?   discriminator
      |  +--ro remote-multiplier?      multiplier
      |  +--ro demand-capability?      boolean {demand-mode}?
      |  +--ro source-port?            inet:port-number
      |  +--ro dest-port?              inet:port-number
      |  +--ro session-running
      |  |  +--ro session-index?               uint32
      |  |  +--ro local-state?                 state
      |  |  +--ro remote-state?                state
      |  |  +--ro local-diagnostic?
      |  |  |       iana-bfd-types:diagnostic
      |  |  +--ro remote-diagnostic?
      |  |  |       iana-bfd-types:diagnostic
      |  |  +--ro remote-authenticated?        boolean
      |  |  +--ro remote-authentication-type?
      |  |  |       iana-bfd-types:auth-type {authentication}?
      |  |  +--ro detection-mode?              enumeration
      |  |  +--ro negotiated-tx-interval?      uint32
      |  |  +--ro negotiated-rx-interval?      uint32
```

```
|  |  +--ro detection-time?              uint32
|  |  +--ro echo-tx-interval-in-use?     uint32
|  |          {echo-mode}?
|  +--ro sesssion-statistics
|     +--ro create-time?            yang:date-and-time
|     +--ro last-down-time?         yang:date-and-time
|     +--ro last-up-time?           yang:date-and-time
|     +--ro down-count?             uint32
|     +--ro admin-down-count?       uint32
|     +--ro receive-packet-count?   uint64
|     +--ro send-packet-count?      uint64
|     +--ro receive-bad-packet?     uint64
|     +--ro send-failed-packet?     uint64
+--ro micro-bfd-ipv6
   +--ro path-type?              identityref
   +--ro ip-encapsulation?       boolean
   +--ro local-discriminator?    discriminator
   +--ro remote-discriminator?   discriminator
   +--ro remote-multiplier?      multiplier
   +--ro demand-capability?      boolean {demand-mode}?
   +--ro source-port?            inet:port-number
   +--ro dest-port?              inet:port-number
   +--ro session-running
   |  +--ro session-index?               uint32
   |  +--ro local-state?                 state
   |  +--ro remote-state?                state
   |  +--ro local-diagnostic?
   |  |       iana-bfd-types:diagnostic
   |  +--ro remote-diagnostic?
   |  |       iana-bfd-types:diagnostic
   |  +--ro remote-authenticated?        boolean
   |  +--ro remote-authentication-type?
   |  |       iana-bfd-types:auth-type {authentication}?
   |  +--ro detection-mode?              enumeration
   |  +--ro negotiated-tx-interval?      uint32
   |  +--ro negotiated-rx-interval?      uint32
   |  +--ro detection-time?              uint32
   |  +--ro echo-tx-interval-in-use?     uint32
   |          {echo-mode}?
   +--ro sesssion-statistics
      +--ro create-time?            yang:date-and-time
      +--ro last-down-time?         yang:date-and-time
      +--ro last-up-time?           yang:date-and-time
      +--ro down-count?             uint32
      +--ro admin-down-count?       uint32
      +--ro receive-packet-count?   uint64
      +--ro send-packet-count?      uint64
      +--ro receive-bad-packet?     uint64
```

```
                  +--ro send-failed-packet?     uint64

   notifications:
     +---n lag-notification
        +--ro local-discr?                discriminator
        +--ro remote-discr?               discriminator
        +--ro new-state?                  state
        +--ro state-change-reason?        iana-bfd-types:diagnostic
        +--ro time-of-last-state-change?  yang:date-and-time
        +--ro dest-addr?                  inet:ip-address
        +--ro source-addr?                inet:ip-address
        +--ro session-index?              uint32
        +--ro path-type?                  identityref
        +--ro lag-name?                   if:interface-ref
        +--ro member-link?                if:interface-ref
```

2.9. BFD over MPLS LSPs hierarchy

An "mpls" node is added under the "bfd" node in control-plane-protocol. The configuration is per MPLS FEC under this "mpls" node. In the operational model we support multiple BFD sessions per MPLS FEC (ECMP), the local discriminator is used as key. The "mpls" node can be used in a network device (top-level), or mounted in an LNE or in a network instance.

```
 module: ietf-bfd-mpls
   augment /rt:routing/rt:control-plane-protocols
           /rt:control-plane-protocol/bfd:bfd:
     +--rw mpls
        +--ro session-statistics
        |  +--ro session-count?              uint32
        |  +--ro session-up-count?           uint32
        |  +--ro session-down-count?         uint32
        |  +--ro session-admin-down-count?   uint32
        +--rw egress
        |  +--rw enable?                     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)
        |  |     +--rw min-interval?               uint32
        |  +--rw authentication-parms! {authentication}?
        |     +--rw key-chain?    kc:key-chain-ref
        |     +--rw meticulous?   boolean
        +--rw session-group* [mpls-fec]
           +--rw mpls-fec                  inet:ip-prefix
```

```
      +--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)
      |     +--rw min-interval?               uint32
      +--rw demand-enabled?             boolean {demand-mode}?
      +--rw admin-down?                 boolean
      +--rw authentication-parms! {authentication}?
      |  +--rw key-chain?    kc:key-chain-ref
      |  +--rw meticulous?   boolean
      +--ro sessions*
         +--ro path-type?              identityref
         +--ro ip-encapsulation?       boolean
         +--ro local-discriminator?    discriminator
         +--ro remote-discriminator?   discriminator
         +--ro remote-multiplier?      multiplier
         +--ro demand-capability?      boolean {demand-mode}?
         +--ro source-port?            inet:port-number
         +--ro dest-port?              inet:port-number
         +--ro session-running
         |  +--ro session-index?               uint32
         |  +--ro local-state?                 state
         |  +--ro remote-state?                state
         |  +--ro local-diagnostic?
         |  |       iana-bfd-types:diagnostic
         |  +--ro remote-diagnostic?
         |  |       iana-bfd-types:diagnostic
         |  +--ro remote-authenticated?        boolean
         |  +--ro remote-authentication-type?
         |  |       iana-bfd-types:auth-type {authentication}?
         |  +--ro detection-mode?              enumeration
         |  +--ro negotiated-tx-interval?      uint32
         |  +--ro negotiated-rx-interval?      uint32
         |  +--ro detection-time?              uint32
         |  +--ro echo-tx-interval-in-use?     uint32
         |          {echo-mode}?
         +--ro sesssion-statistics
         |  +--ro create-time?           yang:date-and-time
         |  +--ro last-down-time?        yang:date-and-time
         |  +--ro last-up-time?          yang:date-and-time
         |  +--ro down-count?            uint32
         |  +--ro admin-down-count?      uint32
         |  +--ro receive-packet-count?  uint64
         |  +--ro send-packet-count?     uint64
         |  +--ro receive-bad-packet?    uint64
         |  +--ro send-failed-packet?    uint64
```

```
            +--ro mpls-dest-address?      inet:ip-address

  notifications:
    +---n mpls-notification
       +--ro local-discr?                discriminator
       +--ro remote-discr?               discriminator
       +--ro new-state?                  state
       +--ro state-change-reason?        iana-bfd-types:diagnostic
       +--ro time-of-last-state-change?  yang:date-and-time
       +--ro dest-addr?                  inet:ip-address
       +--ro source-addr?                inet:ip-address
       +--ro session-index?              uint32
       +--ro path-type?                  identityref
       +--ro mpls-dest-address?          inet:ip-address
```

2.10. BFD over MPLS-TE hierarchy

YANG Data Model for TE Topologies [I-D.ietf-teas-yang-te] is augmented. BFD is configured per MPLS-TE tunnel, and BFD session operational data is provided per MPLS-TE LSP.

```
module: ietf-bfd-mpls-te
  augment /rt:routing/rt:control-plane-protocols
          /rt:control-plane-protocol/bfd:bfd:
    +--rw mpls-te
       +--rw egress
       |  +--rw enable?                    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)
       |  |     +--rw min-interval?               uint32
       |  +--rw authentication-parms! {authentication}?
       |     +--rw key-chain?    kc:key-chain-ref
       |     +--rw meticulous?   boolean
       +--ro session-statistics
          +--ro session-count?             uint32
          +--ro session-up-count?          uint32
          +--ro session-down-count?        uint32
          +--ro session-admin-down-count?  uint32
  augment /te:te/te:tunnels/te:tunnel:
    +--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)
     |     +--rw min-interval?              uint32
     +--rw demand-enabled?             boolean {demand-mode}?
     +--rw admin-down?                 boolean
     +--rw authentication-parms! {authentication}?
     |  +--rw key-chain?    kc:key-chain-ref
     |  +--rw meticulous?   boolean
     +--rw encap?                      identityref
   augment /te:te/te:lsps-state/te:lsp:
     +--ro path-type?              identityref
     +--ro ip-encapsulation?       boolean
     +--ro local-discriminator?    discriminator
     +--ro remote-discriminator?   discriminator
     +--ro remote-multiplier?      multiplier
     +--ro demand-capability?      boolean {demand-mode}?
     +--ro source-port?            inet:port-number
     +--ro dest-port?              inet:port-number
     +--ro session-running
     |  +--ro session-index?               uint32
     |  +--ro local-state?                 state
     |  +--ro remote-state?                state
     |  +--ro local-diagnostic?            iana-bfd-types:diagnostic
     |  +--ro remote-diagnostic?           iana-bfd-types:diagnostic
     |  +--ro remote-authenticated?        boolean
     |  +--ro remote-authentication-type?   iana-bfd-types:auth-type
     |  |       {authentication}?
     |  +--ro detection-mode?              enumeration
     |  +--ro negotiated-tx-interval?      uint32
     |  +--ro negotiated-rx-interval?      uint32
     |  +--ro detection-time?              uint32
     |  +--ro echo-tx-interval-in-use?     uint32 {echo-mode}?
     +--ro sesssion-statistics
     |  +--ro create-time?           yang:date-and-time
     |  +--ro last-down-time?        yang:date-and-time
     |  +--ro last-up-time?          yang:date-and-time
     |  +--ro down-count?            uint32
     |  +--ro admin-down-count?      uint32
     |  +--ro receive-packet-count?   uint64
     |  +--ro send-packet-count?     uint64
     |  +--ro receive-bad-packet?    uint64
     |  +--ro send-failed-packet?    uint64
     +--ro mpls-dest-address?     inet:ip-address

   notifications:
     +---n mpls-te-notification
        +--ro local-discr?               discriminator
        +--ro remote-discr?              discriminator
        +--ro new-state?                 state
```

```
      +--ro state-change-reason?         iana-bfd-types:diagnostic
      +--ro time-of-last-state-change?   yang:date-and-time
      +--ro dest-addr?                   inet:ip-address
      +--ro source-addr?                 inet:ip-address
      +--ro session-index?               uint32
      +--ro path-type?                   identityref
      +--ro mpls-dest-address?           inet:ip-address
      +--ro tunnel-name?                 string
```

2.11. Interaction with other YANG modules

Generic YANG Data Model for Connectionless OAM protocols [I-D.ietf-lime-yang-connectionless-oam] describes how the LIME connectionless OAM model could be extended to support BFD.

Also, the operation of the BFD data model depends on configuration parameters that are defined in other YANG modules.

2.11.1. Module ietf-interfaces

The following boolean configuration is defined in A YANG Data Model for Interface Management [RFC7223]:

/if:interfaces/if:interface/if:enabled
: If this configuration is set to "false", no BFD packets can be transmitted or received on that interface.

2.11.2. Module ietf-ip

The following boolean configuration is defined in A YANG Data Model for IP Management [RFC7277]:

/if:interfaces/if:interface/ip:ipv4/ip:enabled
: If this configuration is set to "false", no BFD IPv4 packets can be transmitted or received on that interface.

/if:interfaces/if:interface/ip:ipv4/ip:forwarding
: If this configuration is set to "false", no BFD IPv4 packets can be transmitted or received on that interface.

/if:interfaces/if:interface/ip:ipv6/ip:enabled
: If this configuration is set to "false", no BFD IPv6 packets can be transmitted or received on that interface.

/if:interfaces/if:interface/ip:ipv6/ip:forwarding
: If this configuration is set to "false", no BFD IPv6 packets can be transmitted or received on that interface.

2.11.3. Module ietf-mpls

The following boolean configuration is defined in A YANG Data Model for MPLS Base [I-D.ietf-mpls-base-yang]:

/rt:routing/mpls:mpls/mpls:interface/mpls:config/mpls:enabled
If this configuration is set to "false", no BFD MPLS packets can be transmitted or received on that interface.

2.11.4. Module ietf-te

The following configuration is defined in the "ietf-te" YANG module YANG Data Model for TE Topology [I-D.ietf-teas-yang-te]:

/ietf-te:te/ietf-te:tunnels/ietf-te:tunnel/ietf-te:config/ietf-te:admin-status
If this configuration is not set to "state-up", no BFD MPLS packets can be transmitted or received on that tunnel.

2.12. IANA BFD YANG Module

```
<CODE BEGINS> file "iana-bfd-types@2018-01-11.yang"

module iana-bfd-types {
  namespace "urn:ietf:params:xml:ns:yang:iana-bfd-types";

  prefix "iana-bfd-types";

  organization "IANA";

  contact
    "        Internet Assigned Numbers Authority

    Postal: ICANN
            4676 Admiralty Way, Suite 330
            Marina del Rey, CA 90292

    Tel:    +1 310 823 9358
    <mailto:iana@iana.org>";

  description
    "This module contains a collection of YANG data types
     considered defined by IANA and used for BFD.

     Copyright (c) 2017 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 XXXX; see
       the RFC itself for full legal notices.";

    revision 2018-01-11 {
      description "Initial revision.";
      reference "RFC XXXX: IANA BFD YANG Data Types.";
    }

    // RFC Ed.: replace XXXX with actual RFC number and remove this
    // note


    typedef diagnostic {
      type enumeration {
        enum none {
          value 0;
          description "None";
        }
        enum control-expiry {
          value 1;
          description "Control timer expiry";
        }
        enum echo-failed {
          value 2;
          description "Echo failure";
        }
        enum neighbor-down {
          value 3;
          description "Neighbor down";
        }
        enum forwarding-reset {
          value 4;
          description "Forwarding reset";
        }
        enum path-down {
          value 5;
          description "Path down";
        }
        enum concatenated-path-down {
          value 6;
          description "Concatenated path down";
        }
```

```
        enum admin-down {
          value 7;
          description "Admin down";
        }
        enum reverse-concatenated-path-down {
          value 8;
          description "Reverse concatenated path down";
        }
        enum mis-connectivity-defect {
          value 9;
          description "Mis-connectivity defect as specified in RFC6428";
        }
      }
      description
        "BFD diagnostic as defined in RFC5880. Range is 0 to 31.";
    }

    typedef auth-type {
      type enumeration {
        enum reserved {
          value 0;
          description "Reserved";
        }
        enum simple-password {
          value 1;
          description "Simple password";
        }
        enum keyed-md5 {
          value 2;
          description "Keyed MD5";
        }
        enum meticulous-keyed-md5 {
          value 3;
          description "Meticulous keyed MD5";
        }
        enum keyed-sha1 {
          value 4;
          description "Keyed SHA1";
        }
        enum meticulous-keyed-sha1 {
          value 5;
          description "Meticulous keyed SHA1";
        }
      }
      description
        "BFD authentication type as defined in RFC5880. Range is 0 to
         255.";
    }
```

```
  }

  <CODE ENDS>
```

2.13. BFD types YANG Module

```
 <CODE BEGINS> file "ietf-bfd-types@2018-01-11.yang"

 module ietf-bfd-types {
   namespace "urn:ietf:params:xml:ns:yang:ietf-bfd-types";

   prefix "bfd-types";

   import iana-bfd-types {
     prefix "iana-bfd-types";
   }

   import ietf-inet-types {
     prefix "inet";
   }

   import ietf-yang-types {
     prefix "yang";
   }

   import ietf-routing {
     prefix "rt";
   }

   import ietf-key-chain {
     prefix "kc";
   }

   organization "IETF BFD Working Group";

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

      Editors:  Reshad Rahman (rrahman@cisco.com),
                Lianshu Zheng (vero.zheng@huawei.com),
                Mahesh Jethanandani (mjethanandani@gmail.com)";

   description
     "This module contains a collection of BFD specific YANG data type
      definitions, as per RFC5880.

      Copyright (c) 2017 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 XXXX; see
      the RFC itself for full legal notices.";

   revision 2018-01-11 {
     description "Initial revision.";
     reference "RFC XXXX: A YANG data model for BFD";
   }

   // RFC Ed.: replace XXXX with actual RFC number and remove this
   // note

   identity bfdv1 {
     base "rt:control-plane-protocol";
     description "BFD protocol version 1 as per RFC5880.";
   }

   typedef discriminator {
     type uint32 {
       range 1..4294967295;
     }
     description "BFD discriminator";
   }

   typedef state {
     type enumeration {
       enum adminDown {
         value 0;
         description "admindown";
       }
       enum down {
         value 1;
         description "down";
       }
       enum init {
         value 2;
         description "init";
       }
       enum up {
         value 3;
```

```
        description "up";
      }
    }
    description "BFD state as defined in RFC5880";
  }

  typedef multiplier {
    type uint8 {
      range 1..255;
    }
    description "Multiplier";
  }

  typedef hops {
    type uint8 {
      range 1..255;
    }
    description
      "This corresponds to Time To Live for IPv4 and corresponds to hop
       limit for IPv6";
  }

  /*
   * Identity definitions
   */
  identity path-type {
    description
      "Base identity for BFD path type. The path type indicates
       the type of path on which BFD is running";
  }
  identity path-ip-sh {
    base path-type;
    description "BFD on IP single hop";
  }
  identity path-ip-mh {
    base path-type;
    description "BFD on IP multi hop";
  }
  identity path-mpls-te {
    base path-type;
    description "BFD on MPLS Traffic Engineering";
  }
  identity path-mpls-lsp {
    base path-type;
    description "BFD on MPLS Label Switched Path";
  }
  identity path-lag {
    base path-type;
```

```
    description "Micro-BFD on LAG member links";
  }

  identity encap-type {
    description
      "Base identity for BFD encapsulation type.";
  }
  identity encap-ip {
    base encap-type;
    description "BFD with IP encapsulation.";
  }

  /*
   * Feature definitions.
   */
  feature authentication {
    description "BFD authentication supported";
  }

  feature demand-mode {
    description "BFD demand mode supported";
  }

  feature echo-mode {
    description "BFD echo mode supported";
  }

  /*
   * Groupings
   */
  grouping auth-parms {
    description
      "Grouping for BFD authentication parameters
       (see section 6.7 of RFC5880).";
    container authentication-parms {
      if-feature authentication;
      presence
        "Enables BFD authentication (see section 6.7 of RFC5880).";
      description "Parameters for BFD authentication";

      leaf key-chain {
        type kc:key-chain-ref;
        description "Name of key-chain";
      }

      leaf meticulous {
        type boolean;
        description
```

```
          "Enables meticulous mode as described in section 6.7 " +
          "of RFC5880";
      }
    }
  }

  grouping base-cfg-parms {
    description "BFD grouping for base config parameters";
    leaf local-multiplier {
      type multiplier;
      default 3;
      description "Multiplier transmitted by local system";
    }

    choice interval-config-type {
      description
        "Two interval values or 1 value used for both tx and rx";
      case tx-rx-intervals {
        leaf desired-min-tx-interval {
          type uint32;
          units microseconds;
          default 1000000;
          description
            "Desired minimum transmit interval of control packets";
        }

        leaf required-min-rx-interval {
          type uint32;
          units microseconds;
          default 1000000;
          description
            "Required minimum receive interval of control packets";
        }
      }
      case single-interval {
        leaf min-interval {
          type uint32;
          units microseconds;
          default 1000000;
          description
            "Desired minimum transmit interval and required " +
            "minimum receive interval of control packets";
        }
      }
    }
  }

  grouping client-cfg-parms {
```

```
    description
      "BFD grouping for config parameters
       used by clients of BFD, e.g. IGP or MPLS";

    leaf enable {
      type boolean;
      default false;
      description
        "Indicates whether the BFD is enabled.";
    }
    uses base-cfg-parms;
  }

  grouping common-cfg-parms {
    description
      "BFD grouping for common config parameters";

    uses base-cfg-parms;

    leaf demand-enabled {
      if-feature demand-mode;
      type boolean;
      default false;
      description
        "To enable demand mode";
    }

    leaf admin-down {
      type boolean;
      default false;
      description
        "Is the BFD session administratively down";
    }
    uses auth-parms;
  }

  grouping all-session {
    description "BFD session operational information";
    leaf path-type {
      type identityref {
        base path-type;
      }
      config "false";
      description
        "BFD path type, this indicates the path type that BFD is
        running on";
    }
    leaf ip-encapsulation {
```

```
      type boolean;
      config "false";
      description "Whether BFD encapsulation uses IP";
    }
    leaf local-discriminator {
      type discriminator;
      config "false";
      description "Local discriminator";
    }
    leaf remote-discriminator {
      type discriminator;
      config "false";
      description "Remote discriminator";
    }
    leaf remote-multiplier {
      type multiplier;
      config "false";
      description "Remote multiplier";
    }
    leaf demand-capability {
      if-feature demand-mode;
      type boolean;
      config "false";
      description "Local demand mode capability";
    }
    leaf source-port {
      when "../ip-encapsulation = 'true'" {
        description
          "Source port valid only when IP encapsulation is used";
      }
      type inet:port-number;
      config "false";
      description "Source UDP port";
    }
    leaf dest-port {
      when "../ip-encapsulation = 'true'" {
        description
          "Destination port valid only when IP encapsulation is used";
      }
      type inet:port-number;
      config "false";
      description "Destination UDP port";
    }

    container session-running {
      config "false";
      description "BFD session running information";
      leaf session-index {
```

```
          type uint32;
          description
            "An index used to uniquely identify BFD sessions";
        }
        leaf local-state {
          type state;
          description "Local state";
        }
        leaf remote-state {
          type state;
          description "Remote state";
        }
        leaf local-diagnostic {
          type iana-bfd-types:diagnostic;
          description "Local diagnostic";
        }
        leaf remote-diagnostic {
          type iana-bfd-types:diagnostic;
          description "Remote diagnostic";
        }
        leaf remote-authenticated {
          type boolean;
          description
            "Indicates whether incoming BFD control packets are
            authenticated";
        }
        leaf remote-authentication-type {
          when "../remote-authenticated = 'true'" {
            description
              "Only valid when incoming BFD control packets are
               authenticated";
          }
          if-feature authentication;
          type iana-bfd-types:auth-type;
          description
            "Authentication type of incoming BFD control packets";
        }
        leaf detection-mode {
          type enumeration {
            enum async-with-echo {
              value "1";
              description "Async with echo";
            }
            enum async-without-echo {
              value "2";
              description "Async without echo";
            }
            enum demand-with-echo {
```

```
            value "3";
            description "Demand with echo";
          }
          enum demand-without-echo {
            value "4";
            description "Demand without echo";
          }
        }
        description "Detection mode";
      }
      leaf negotiated-tx-interval {
        type uint32;
        units microseconds;
        description "Negotiated transmit interval";
      }
      leaf negotiated-rx-interval {
        type uint32;
        units microseconds;
        description "Negotiated receive interval";
      }
      leaf detection-time {
        type uint32;
        units microseconds;
        description "Detection time";
      }
      leaf echo-tx-interval-in-use {
        when "../../path-type = 'bfd-types:path-ip-sh'" {
          description
            "Echo is supported for IP single-hop only.";
        }
        if-feature echo-mode;
        type uint32;
        units microseconds;
        description "Echo transmit interval in use";
      }
    }

    container sesssion-statistics {
      config "false";
      description "BFD per-session statistics";

      leaf create-time {
        type yang:date-and-time;
        description
          "Time and date when session was created";
      }
      leaf last-down-time {
        type yang:date-and-time;
```

```
        description
          "Time and date of last time the session went down";
      }
      leaf last-up-time {
        type yang:date-and-time;
        description
          "Time and date of last time the session went up";
      }
      leaf down-count {
        type uint32;
        description "Session Down Count";
      }
      leaf admin-down-count {
        type uint32;
        description "Session Admin-Down Count";
      }
      leaf receive-packet-count {
        type uint64;
        description "Received Packet Count";
      }
      leaf send-packet-count {
        type uint64;
        description "Sent Packet Count";
      }
      leaf receive-bad-packet {
        type uint64;
        description "Received bad packet count";
      }
      leaf send-failed-packet {
        type uint64;
        description "Packet Failed to Send Count";
      }
    }
  }

  grouping session-statistics {
    description "Grouping for session counters";
    container session-statistics {
      config false;
      description "BFD session counters";
      leaf session-count {
        type uint32;
        description "Number of sessions";
      }
      leaf session-up-count {
        type uint32;
        description "Count of sessions which are up";
      }
```

```
      leaf session-down-count {
        type uint32;
        description "Count of sessions which are down";
      }
      leaf session-admin-down-count {
        type uint32;
        description "Count of sessions which are admin-down";
      }
    }
  }

  grouping notification-parms {
    description
      "This group describes common parameters that will be sent " +
      "as part of BFD notification";

    leaf local-discr {
      type discriminator;
      description "BFD local discriminator";
    }

    leaf remote-discr {
      type discriminator;
      description "BFD remote discriminator";
    }

    leaf new-state {
      type state;
      description "Current BFD state";
    }

    leaf state-change-reason {
      type iana-bfd-types:diagnostic;
      description "BFD state change reason";
    }

    leaf time-of-last-state-change {
      type yang:date-and-time;
      description
        "Calendar time of previous state change";
    }

    leaf dest-addr {
      type inet:ip-address;
      description "BFD peer address";
    }

    leaf source-addr {
```

```
      type inet:ip-address;
      description "BFD local address";
    }

    leaf session-index {
      type uint32;
      description "An index used to uniquely identify BFD sessions";
    }

    leaf path-type {
      type identityref {
        base path-type;
      }
      description "BFD path type";
    }
  }
}

<CODE ENDS>
```

2.14. BFD top-level YANG Module

```
  <CODE BEGINS> file "ietf-bfd@2018-01-11.yang"

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

    prefix "bfd";

    import ietf-bfd-types {
      prefix "bfd-types";
    }

    import ietf-routing {
      prefix "rt";
    }

    organization "IETF BFD Working Group";

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

       Editors:  Reshad Rahman (rrahman@cisco.com),
                 Lianshu Zheng (vero.zheng@huawei.com),
                 Mahesh Jethanandani (mjethanandani@gmail.com)";

    description
```

```
       "This module contains the YANG definition for BFD parameters as
        per RFC5880.

        Copyright (c) 2017 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 XXXX; see
        the RFC itself for full legal notices.";

     revision 2018-01-11 {
       description "Initial revision.";
       reference "RFC XXXX: A YANG data model for BFD";
     }

     // RFC Ed.: replace XXXX with actual RFC number and remove this
     // note

     augment "/rt:routing/rt:control-plane-protocols/"
           + "rt:control-plane-protocol" {
        when "rt:type = 'bfd-types:bfdv1'" {
         description
           "This augmentation is only valid for a control-plane protocol
            instance of BFD (type 'bfdv1').";
       }
       description "BFD augmentation.";

       container bfd {
         description "BFD top level container";

         uses bfd-types:session-statistics;
       }
     }
   }

   <CODE ENDS>
```

2.15. BFD IP single-hop YANG Module

```
   <CODE BEGINS> file "ietf-bfd-ip-sh@2018-01-11.yang"

   module ietf-bfd-ip-sh {
```

```
  namespace "urn:ietf:params:xml:ns:yang:ietf-bfd-ip-sh";

  prefix "bfd-ip-sh";

  import ietf-bfd-types {
    prefix "bfd-types";
  }

  import ietf-bfd {
    prefix "bfd";
  }

  import ietf-interfaces {
    prefix "if";
  }

  import ietf-inet-types {
    prefix "inet";
  }

  import ietf-routing {
    prefix "rt";
  }

  organization "IETF BFD Working Group";

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

     Editors:  Reshad Rahman (rrahman@cisco.com),
               Lianshu Zheng (vero.zheng@huawei.com),
               Mahesh Jethanandani (mjethanandani@gmail.com)";

  description
    "This module contains the YANG definition for BFD IP single-hop
     as per RFC5881.

     Copyright (c) 2017 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 XXXX; see
        the RFC itself for full legal notices.";

    revision 2018-01-11 {
      description "Initial revision.";
      reference "RFC XXXX: A YANG data model for BFD IP single-hop";
    }

    // RFC Ed.: replace XXXX with actual RFC number and remove this
    // note

    augment "/rt:routing/rt:control-plane-protocols/"
          + "rt:control-plane-protocol/bfd:bfd" {
      description "BFD augmentation for IP single-hop";
      container ip-sh {
        description "BFD IP single-hop top level container";

        uses bfd-types:session-statistics;

        list sessions {
          key "interface dest-addr";
          description "List of IP single-hop sessions";
          leaf interface {
            type if:interface-ref;
            description
              "Interface on which the BFD session is running.";
          }
          leaf dest-addr {
            type inet:ip-address;
            description "IP address of the peer";
          }
          leaf source-addr {
            type inet:ip-address;
            description "Local address";
          }

          uses bfd-types:common-cfg-parms;

          uses bfd-types:all-session;
        }
        list interfaces {
          key "interface";
          description "List of interfaces";
          leaf interface {
            type if:interface-ref;
            description
              "BFD information for this interface.";
          }
```

```
          uses bfd-types:auth-parms;
        }
      }
    }

    notification singlehop-notification {
      description
        "Notification for BFD single-hop session state change. An " +
        "implementation may rate-limit notifications, e.g. when a" +
        "session is continuously changing state.";

      uses bfd-types:notification-parms;

      leaf interface {
        type if:interface-ref;
        description "Interface to which this BFD session belongs to";
      }

      leaf echo-enabled {
        type boolean;
        description "Was echo enabled for BFD";
      }
    }

  }

  <CODE ENDS>
```

2.16. BFD IP multi-hop YANG Module

```
<CODE BEGINS> file "ietf-bfd-ip-mh@2018-01-11.yang"

module ietf-bfd-ip-mh {
  namespace "urn:ietf:params:xml:ns:yang:ietf-bfd-ip-mh";
  // replace with IANA namespace when assigned
  prefix "bfd-ip-mh";

  import ietf-bfd-types {
    prefix "bfd-types";
  }

  import ietf-bfd {
    prefix "bfd";
  }

  import ietf-inet-types {
    prefix "inet";
  }
```

```
  import ietf-routing {
    prefix "rt";
  }

  organization "IETF BFD Working Group";

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

     Editors:  Reshad Rahman (rrahman@cisco.com),
               Lianshu Zheng (vero.zheng@huawei.com),
               Mahesh Jethanandani (mjethanandani@gmail.com)";

  description
    "This module contains the YANG definition for BFD IP multi-hop
     as per RFC5883.

     Copyright (c) 2017 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 XXXX; see
     the RFC itself for full legal notices.";

  revision 2018-01-11 {
    description "Initial revision.";
    reference "RFC XXXX: A YANG data model for BFD IP multi-hop";
  }

  // RFC Ed.: replace XXXX with actual RFC number and remove this
  // note

  augment "/rt:routing/rt:control-plane-protocols/"
        + "rt:control-plane-protocol/bfd:bfd" {
    description "BFD augmentation for IP multi-hop";
    container ip-mh {
      description "BFD IP multi-hop top level container";

      uses bfd-types:session-statistics;

      list session-group {
```

```
        key "source-addr dest-addr";
              description
                "Group of BFD IP multi-hop sessions (for ECMP). A " +
          "group of sessions is between 1 source and 1  " +
          "destination, each session has a different field " +
          "in UDP/IP hdr for ECMP.";

        leaf source-addr {
          type inet:ip-address;
          description
            "Local IP address";
        }
        leaf dest-addr {
          type inet:ip-address;
          description
            "IP address of the peer";
        }
        uses bfd-types:common-cfg-parms;

        leaf tx-ttl {
          type bfd-types:hops;
          default 255;
          description "Hop count of outgoing BFD control packets";
        }
        leaf rx-ttl {
          type bfd-types:hops;
          mandatory true;
          description
            "Minimum allowed hop count value for incoming BFD control
             packets. Control packets whose hop count is lower than this
                   value are dropped.";
        }
        list sessions {
                config false;
          description
            "The multiple BFD sessions between a source and a " +
            "destination.";
          uses bfd-types:all-session;
        }
      }
    }
  }

  notification multihop-notification {
    description
      "Notification for BFD multi-hop session state change. An " +
      "implementation may rate-limit notifications, e.g. when a" +
      "session is continuously changing state.";
```

```
    uses bfd-types:notification-parms;
  }
}

<CODE ENDS>
```

2.17. BFD over LAG YANG Module

```
   <CODE BEGINS> file "ietf-bfd-lag@2018-01-11.yang"

   module ietf-bfd-lag {
     namespace "urn:ietf:params:xml:ns:yang:ietf-bfd-lag";
     // replace with IANA namespace when assigned
     prefix "bfd-lag";

     import ietf-bfd-types {
       prefix "bfd-types";
     }

     import ietf-bfd {
       prefix "bfd";
     }

     import ietf-interfaces {
       prefix "if";
     }

     import ietf-inet-types {
       prefix "inet";
     }

     import ietf-routing {
       prefix "rt";
     }

     organization "IETF BFD Working Group";

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

        Editors:  Reshad Rahman (rrahman@cisco.com),
                  Lianshu Zheng vero.zheng@huawei.com),
                  Mahesh Jethanandani (mjethanandani@gmail.com)";

     description
       "This module contains the YANG definition for BFD over LAG
        interfaces as per RFC7130.
```

```
  revision 2018-01-11 {
    description "Initial revision.";
    reference "RFC XXXX: A YANG data model for BFD over LAG";
  }

  // RFC Ed.: replace XXXX with actual RFC number and remove this
  // note

  augment "/rt:routing/rt:control-plane-protocols/"
        + "rt:control-plane-protocol/bfd:bfd" {
    description "BFD augmentation for LAG";
    container lag {
      description "BFD over LAG top level container";

      container micro-bfd-ipv4-session-statistics {
        description "Micro-BFD IPv4 session counters";
        uses bfd-types:session-statistics;
      }
      container micro-bfd-ipv6-session-statistics {
        description "Micro-BFD IPv6 session counters";
        uses bfd-types:session-statistics;
      }

      list sessions {
        key "lag-name";
        description "A LAG interface on which BFD is running";
        leaf lag-name {
          type if:interface-ref ;
          description "Name of the LAG";
        }
        leaf ipv4-dest-addr {
          type inet:ipv4-address;
          description
            "IPv4 address of the peer, for IPv4 micro-BFD.";
        }
```

```
          leaf ipv6-dest-addr {
            type inet:ipv6-address;
            description
              "IPv6 address of the peer, for IPv6 micro-BFD.";
          }
          uses bfd-types:common-cfg-parms;

          leaf use-ipv4 {
            type boolean;
            description "Using IPv4 micro-BFD.";
          }
          leaf use-ipv6 {
            type boolean;
            description "Using IPv6 micro-BFD.";
          }

          list member-links {
            key "member-link";
            config false;
            description
              "Micro-BFD over LAG. This represents one member link";

            leaf member-link {
              type if:interface-ref;
              description
                "Member link on which micro-BFD is running";
            }
            container micro-bfd-ipv4 {
              when "../../use-ipv4 = 'true'" {
                description "Needed only if IPv4 is used.";
              }
              description
                "Micro-BFD IPv4 session state on member link";
              uses bfd-types:all-session;
            }
            container micro-bfd-ipv6 {
              when "../../use-ipv6 = 'true'" {
                description "Needed only if IPv6 is used.";
              }
              description
                "Micro-BFD IPv6 session state on member link";
              uses bfd-types:all-session;
            }
          }
        }
      }
    }
```

```
    notification lag-notification {
      description
        "Notification for BFD over LAG session state change. " +
        "An implementation may rate-limit notifications, e.g. when a" +
        "session is continuously changing state.";

      uses bfd-types:notification-parms;

      leaf lag-name {
        type if:interface-ref;
        description "LAG interface name";
      }

      leaf member-link {
        type if:interface-ref;
        description "Member link on which BFD is running";
      }
    }
  }

  <CODE ENDS>
```

2.18. BFD over MPLS YANG Module

```
  <CODE BEGINS> file "ietf-bfd-mpls@2018-01-11.yang"

  module ietf-bfd-mpls {
    namespace "urn:ietf:params:xml:ns:yang:ietf-bfd-mpls";
    // replace with IANA namespace when assigned
    prefix "bfd-mpls";

    import ietf-bfd-types {
      prefix "bfd-types";
    }

    import ietf-bfd {
      prefix "bfd";
    }

    import ietf-inet-types {
      prefix "inet";
    }

    import ietf-routing {
      prefix "rt";
    }

    organization "IETF BFD Working Group";
```

```
    contact
      "WG Web:   <http://tools.ietf.org/wg/bfd>
       WG List:  <rtg-bfd@ietf.org>

       Editors:   Reshad Rahman (rrahman@cisco.com),
                  Lianshu Zheng (vero.zheng@huawei.com),
                  Mahesh Jethanandani (mjethanandani@gmail.com)";

    description
      "This module contains the YANG definition for BFD parameters for
       MPLS LSPs as per RFC5884.

       Copyright (c) 2017 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 XXXX; see
       the RFC itself for full legal notices.";

    revision 2018-01-11 {
      description "Initial revision.";
      reference "RFC XXXX: A YANG data model for BFD over MPLS LSPs";
    }

    // RFC Ed.: replace XXXX with actual RFC number and remove this
    // note

    identity encap-gach {
      base bfd-types:encap-type;
      description
        "BFD with G-ACh encapsulation as per RFC5586.";
    }

    identity encap-ip-gach {
      base bfd-types:encap-type;
      description
        "BFD with IP and G-ACh encapsulation as per RFC5586.";
    }

    grouping encap-cfg {
      description "Configuration for BFD encapsulation";
```

```
      leaf encap {
        type identityref {
          base bfd-types:encap-type;
        }
        default bfd-types:encap-ip;
        description "BFD encapsulation";
      }
    }

    grouping mpls-dest-address {
      description "Destination address as per RFC5884";

      leaf mpls-dest-address {
        type inet:ip-address;
        config "false";
        description
          "Destination address as per RFC5884.
           Needed if IP encapsulation is used";
      }
    }


    augment "/rt:routing/rt:control-plane-protocols/"
          + "rt:control-plane-protocol/bfd:bfd" {
      description "BFD augmentation for MPLS";
      container mpls {
        description "BFD MPLS top level container";

        uses bfd-types:session-statistics;

        container egress {
          description "Egress configuration";

          uses bfd-types:client-cfg-parms;

          uses bfd-types:auth-parms;
        }

        list session-group {
          key "mpls-fec";
          description
            "Group of BFD MPLS sessions (for ECMP). A group of " +
            "sessions is for 1 FEC, each session has a different " +
            "field in UDP/IP hdr for ECMP.";
          leaf mpls-fec {
            type inet:ip-prefix;
            description "MPLS FEC";
          }
```

```
          uses bfd-types:common-cfg-parms;

          list sessions {
                  config false;
            description
              "The BFD sessions for an MPLS FEC. Local " +
              "discriminator is unique for each session in the " +
              "group.";
            uses bfd-types:all-session;

            uses bfd-mpls:mpls-dest-address;
          }
        }
      }
    }

    notification mpls-notification {
      description
        "Notification for BFD over MPLS FEC session state change. " +
        "An implementation may rate-limit notifications, e.g. when a" +
        "session is continuously changing state.";

      uses bfd-types:notification-parms;

      leaf mpls-dest-address {
        type inet:ip-address;
        description
          "Destination address as per RFC5884.
           Needed if IP encapsulation is used";
      }
    }
  }

  <CODE ENDS>
```

2.19. BFD over MPLS-TE YANG Module

```
  <CODE BEGINS> file "ietf-bfd-mpls-te@2018-01-11.yang"

  module ietf-bfd-mpls-te {
    namespace "urn:ietf:params:xml:ns:yang:ietf-bfd-mpls-te";
    // replace with IANA namespace when assigned
    prefix "bfd-mpls-te";

    import ietf-bfd-types {
      prefix "bfd-types";
    }
```

```
  import ietf-bfd {
    prefix "bfd";
  }

  import ietf-bfd-mpls {
    prefix "bfd-mpls";
  }

  import ietf-te {
    prefix "te";
  }

  import ietf-routing {
    prefix "rt";
  }

  organization "IETF BFD Working Group";

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

     Editors:  Reshad Rahman (rrahman@cisco.com),
               Lianshu Zheng (vero.zheng@huawei.com),
               Mahesh Jethanandani (mjethanandani@gmail.com)";

  description
    "This module contains the YANG definition for BFD parameters for
     MPLS Traffic Engineering as per RFC5884.

     Copyright (c) 2017 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 XXXX; see
     the RFC itself for full legal notices.";

  revision 2018-01-11 {
    description "Initial revision.";
    reference "RFC XXXX: A YANG data model for BFD over MPLS-TE";
  }
```

```
    // RFC Ed.: replace XXXX with actual RFC number and remove this
    // note

    augment "/rt:routing/rt:control-plane-protocols/"
          + "rt:control-plane-protocol/bfd:bfd" {
      description "BFD augmentation for MPLS-TE";
      container mpls-te {
        description "BFD MPLS-TE top level container";

        container egress {
          description "Egress configuration";

          uses bfd-types:client-cfg-parms;

          uses bfd-types:auth-parms;
        }

        uses bfd-types:session-statistics;
      }
    }

    augment "/te:te/te:tunnels/te:tunnel" {
      description "BFD configuration on MPLS-TE tunnel.";

      uses bfd-types:common-cfg-parms;

      uses bfd-mpls:encap-cfg;
    }

    augment "/te:te/te:lsps-state/te:lsp" {
      when "/te:te/te:lsps-state/te:lsp/te:origin-type != 'transit'" {
        description "BFD information not needed at transit points";
      }
      description "BFD state information on MPLS-TE LSP.";

      uses bfd-types:all-session;

      uses bfd-mpls:mpls-dest-address;
    }


    notification mpls-te-notification {
      description
        "Notification for BFD over MPLS-TE session state change. " +
        "An implementation may rate-limit notifications, e.g. when a" +
        "session is continuously changing state.";

      uses bfd-types:notification-parms;
```

```
      uses bfd-mpls:mpls-dest-address;

      leaf tunnel-name {
        type string;
        description "MPLS-TE tunnel on which BFD was running.";
      }
    }
  }
```

<CODE ENDS>

2.20. Security Considerations

The YANG module defined in this memo is designed to be accessed via the NETCONF protocol [RFC6241]. The lowest NETCONF layer is the secure transport layer and the mandatory to implement secure transport is SSH [RFC6242]. The NETCONF access control model [RFC6536] provides the means to restrict access for particular NETCONF users to a pre-configured subset of all available NETCONF protocol operations and content.

The YANG module has writeable data nodes which can be used for creation of BFD sessions and modification of BFD session parameters. The system should "police" creation of BFD sessions to prevent new sessions from causing existing BFD sessions to fail. For BFD session modification, the BFD protocol has mechanisms in place which allow for in service modification.

2.21. IANA Considerations

The IANA is requested to as assign a new namespace URI from the IETF XML registry.

This document registers the following namesace URIs in the IETF XML registry [RFC3688]:

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

Registrant Contact: The IESG.

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

URI: urn:ietf:params:xml:ns:yang:ietf-bfd-ip-sh

Registrant Contact: The IESG.

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

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

Registrant Contact: The IESG.

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

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

Registrant Contact: The IESG.

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

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

Registrant Contact: The IESG.

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

URI: urn:ietf:params:xml:ns:yang:ietf-bfd-mpls-te

Registrant Contact: The IESG.

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

2.21.1. IANA-Maintained iana-bfd-types module

This document defines the initial version of the IANA-maintained iana-bfd-types YANG module.

The iana-bfd-types YANG module is intended to reflect the "BFD Diagnostic Codes" registry and "BFD Authentication Types" registry at https://www.iana.org/assignments/bfd-parameters/bfd-parameters.xhtml

2.22. Acknowledgements

We would also like to thank Nobo Akiya and Jeff Haas for their encouragement on this work. We would also like to thank Rakesh Gandhi and Tarek Saad for their help on the MPLS-TE model. We would also like to thank Acee Lindem for his guidance.

3. References

3.1. 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>.

[RFC5586] Bocci, M., Ed., Vigoureux, M., Ed., and S. Bryant, Ed., "MPLS Generic Associated Channel", RFC 5586, DOI 10.17487/RFC5586, June 2009, <https://www.rfc-editor.org/info/rfc5586>.

[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>.

[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>.

[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>.

[RFC5885] Nadeau, T., Ed. and C. Pignataro, Ed., "Bidirectional Forwarding Detection (BFD) for the Pseudowire Virtual Circuit Connectivity Verification (VCCV)", RFC 5885, DOI 10.17487/RFC5885, June 2010, <https://www.rfc-editor.org/info/rfc5885>.

[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>.

[RFC7223] Bjorklund, M., "A YANG Data Model for Interface Management", RFC 7223, DOI 10.17487/RFC7223, May 2014, <https://www.rfc-editor.org/info/rfc7223>.

[RFC7277] Bjorklund, M., "A YANG Data Model for IP Management", RFC 7277, DOI 10.17487/RFC7277, June 2014, <https://www.rfc-editor.org/info/rfc7277>.

[RFC8022] Lhotka, L. and A. Lindem, "A YANG Data Model for Routing Management", RFC 8022, DOI 10.17487/RFC8022, November 2016, <https://www.rfc-editor.org/info/rfc8022>.

3.2. Informative References

[I-D.dsdt-nmda-guidelines]
Bjorklund, M., Schoenwaelder, J., Shafer, P., Watsen, K., and R. Wilton, "Guidelines for YANG Module Authors (NMDA)", draft-dsdt-nmda-guidelines-01 (work in progress), May 2017.

[I-D.ietf-lime-yang-connectionless-oam]
Kumar, D., Wang, Z., Wu, Q., Rahman, R., and S. Raghavan, "Generic YANG Data Model for the Management of Operations, Administration, and Maintenance (OAM) Protocols that use Connectionless Communications", draft-ietf-lime-yang-connectionless-oam-18 (work in progress), November 2017.

[I-D.ietf-mpls-base-yang]
Raza, K., Gandhi, R., Liu, X., Beeram, V., Saad, T., Bryskin, I., Chen, X., Jones, R., and B. Wen, "A YANG Data Model for MPLS Base", draft-ietf-mpls-base-yang-05 (work in progress), July 2017.

[I-D.ietf-netmod-schema-mount]
Bjorklund, M. and L. Lhotka, "YANG Schema Mount", draft-ietf-netmod-schema-mount-08 (work in progress), October 2017.

[I-D.ietf-rtgwg-device-model]
Lindem, A., Berger, L., Bogdanovic, D., and C. Hopps, "Network Device YANG Logical Organization", draft-ietf-rtgwg-device-model-02 (work in progress), March 2017.

[I-D.ietf-rtgwg-lne-model]
Berger, L., Hopps, C., Lindem, A., Bogdanovic, D., and X. Liu, "YANG Logical Network Elements", draft-ietf-rtgwg-lne-model-05 (work in progress), December 2017.

[I-D.ietf-rtgwg-ni-model]
Berger, L., Hopps, C., Lindem, A., Bogdanovic, D., and X. Liu, "YANG Network Instances", draft-ietf-rtgwg-ni-model-05 (work in progress), December 2017.

[I-D.ietf-teas-yang-te]
Saad, T., Gandhi, R., Liu, X., Beeram, V., Shah, H., and I. Bryskin, "A YANG Data Model for Traffic Engineering Tunnels and Interfaces", draft-ietf-teas-yang-te-10 (work in progress), December 2017.

[RFC8177] Lindem, A., Ed., Qu, Y., Yeung, D., Chen, I., and J. Zhang, "YANG Data Model for Key Chains", RFC 8177, DOI 10.17487/RFC8177, June 2017, <https://www.rfc-editor.org/info/rfc8177>.

Appendix A. Echo function configuration example

The following intervals are added for the echo function (if supported):

desired-min-echo-tx-interval
This is the minimum interval that the local system would like to use when transmitting BFD echo packets. If 0, the echo function as defined in BFD [RFC5880] is disabled.

required-min-echo-rx-interval

This is the Required Min Echo RX Interval as defined in BFD [RFC5880].

```
 module: example-bfd-echo
   augment /rt:routing/rt:control-plane-protocols/rt:control-plane-proto
 col/bfd:bfd/bfd-ip-sh:ip-sh/bfd-ip-sh:sessions:
     +--rw echo {bfd-types:echo-mode}?
        +--rw desired-min-echo-tx-interval?    uint32
        +--rw required-min-echo-rx-interval?   uint32
```

A.1. Example YANG module for BFD echo function

```
module example-bfd-echo {
  namespace "tag:example.com,2017:example-bfd-echo";

  prefix "example-bfd-echo";

  import ietf-bfd-types {
    prefix "bfd-types";
  }

  import ietf-bfd {
    prefix "bfd";
  }

  import ietf-bfd-ip-sh {
    prefix "bfd-ip-sh";
  }

  import ietf-routing {
    prefix "rt";
  }

  organization "IETF BFD Working Group";

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

     Editors:  Reshad Rahman (rrahman@cisco.com),
               Lianshu Zheng (vero.zheng@huawei.com),
               Mahesh Jethanandani (mjethanandani@gmail.com)";

  description
    "This module contains an example YANG augmentation for configuration
     of BFD echo function.
```

```
  revision 2018-01-11 {
    description "Initial revision.";
    reference
      "RFC XXXX: A YANG data model example augmentation for BFD echo
       function";
  }

  // RFC Ed.: replace XXXX with actual RFC number and remove this
  // note

  /*
   * Groupings
   */
  grouping echo-cfg-parms {
    description "BFD grouping for echo config parameters";
    leaf desired-min-echo-tx-interval {
      type uint32;
      units microseconds;
      default 0;
      description "Desired minumum transmit interval for echo";
    }

    leaf required-min-echo-rx-interval {
      type uint32;
      units microseconds;
      default 0;
      description "Required minimum receive interval for echo";
    }
  }

  augment "/rt:routing/rt:control-plane-protocols/"
        + "rt:control-plane-protocol/bfd:bfd/bfd-ip-sh:ip-sh/"
        + "bfd-ip-sh:sessions" {
    description "Augmentation for BFD echo fucntion.";
```

```
    container echo {
      if-feature bfd-types:echo-mode;

      description "BFD echo function container";

      uses echo-cfg-parms;
    }
  }
}
```

Appendix B. BFD client configuration example

The following is an example of how a BFD client could use the grouping client-cfg-parms.

```
module: example-bfd-client
    +--rw area* [area-id]
       +--rw area-id      uint32
       +--rw bfd
       |  +--rw enable?                     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)
       |        +--rw min-interval?               uint32
       +--rw interface* [interface]
          +--rw interface    if:interface-ref
          +--rw bfd
             +--rw enable?                     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)
                   +--rw min-interval?               uint32
```

B.1. Example YANG module for BFD client

```
module example-bfd-client {
  namespace "tag:example.com,2017:example-bfd-client";

  prefix "example-bfd-client";

  import ietf-bfd-types {
```

```
      prefix "bfd-types";
    }

    import ietf-interfaces {
      prefix "if";
    }

    organization "IETF BFD Working Group";

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

       Editors:  Reshad Rahman (rrahman@cisco.com),
                 Lianshu Zheng (vero.zheng@huawei.com),
                 Mahesh Jethanandani (mjethanandani@gmail.com)";

    description
      "This module contains an example of how a protocol which is a
       client of BFD would use BFD parameters.

       Copyright (c) 2017 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 XXXX; see
       the RFC itself for full legal notices.";

    revision 2018-01-11 {
      description "Initial revision.";
      reference
        "RFC XXXX: A YANG data model example for BFD client.";
    }

    // RFC Ed.: replace XXXX with actual RFC number and remove this
    // note

    feature routing-app-bfd {
      description "BFD configuration under routing-app";
    }

    list area {
```

```
        key "area-id";

        description "Specify a routing area.";

        leaf area-id {
          type uint32;
          description "Area";
        }

        container bfd {
          description "BFD configuration for the area.";
          uses bfd-types:client-cfg-parms {
            if-feature routing-app-bfd;
          }
        }

        list interface {
          key "interface";
          description "List of interfaces";
          leaf interface {
            type if:interface-ref;
            description "Interface";
          }
          container bfd {
            description "BFD configuration for the interface.";
            uses bfd-types:client-cfg-parms {
              if-feature routing-app-bfd;
            }
          }
        }
      }
    }
```

Appendix C. Change log

RFC Editor: Remove this section upon publication as an RFC.

C.1. Changes between versions -07 and -08

- Timer intervals in client-cfg-parms are not mandatory anymore.
- Added list of interfaces under "ip-sh" node for authentication parameters.
- Renamed replay-protection to meticulous.

C.2. Changes between versions -06 and -07

- New ietf-bfd-types module.
- Grouping for BFD clients to have BFD multiplier and interval values.
- Change in ietf-bfd-mpls-te since MPLS-TE model changed.
- Removed bfd- prefix from many names.

C.3. Changes between versions -05 and -06

- Adhere to NMDA-guidelines.
- Echo function config moved to appendix as example.
- Added IANA YANG modules.
- Addressed various comments.

C.4. Changes between versions -04 and -05

- "bfd" node in augment of control-plane-protocol as per A YANG Data Model for Routing Management [RFC8022].
- Removed augment of network-instance. Replaced by schema-mount.
- Added information on interaction with other YANG modules.

C.5. Changes between versions -03 and -04

- Updated author information.
- Fixed YANG compile error in ietf-bfd-lag.yang which was due to incorrect when statement.

C.6. Changes between versions -02 and -03

- Fixed YANG compilation warning due to incorrect revision date in ietf-bfd-ip-sh module.

C.7. Changes between versions -01 and -02

- Replace routing-instance, which has been removed from A YANG Data Model for Routing Management [RFC8022], with network-instance from YANG Network Instances [I-D.ietf-rtgwg-ni-model]

C.8. Changes between versions -00 and -01

- Remove BFD configuration parameters from BFD clients, all BFD configuration parameters in BFD
- YANG module split in multiple YANG modules (one per type of forwarding path)
- For BFD over MPLS-TE we augment MPLS-TE model
- For BFD authentication we now use YANG Data Model for Key Chains [RFC8177]

Authors' Addresses

Reshad Rahman (editor)
Cisco Systems
Canada

Email: rrahman@cisco.com

Lianshu Zheng (editor)
Huawei Technologies
China

Email: vero.zheng@huawei.com

Mahesh Jethanandani (editor)
Cisco Systems

Email: mjethanandani@gmail.com

Santosh Pallagatti
India

Email: santosh.pallagatti@gmail.com

Greg Mirsky
ZTE Corporation

Email: gregimirsky@gmail.com