Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: February 2, 2019

R. Rahman, Ed.
Cisco Systems
L. Zheng, Ed.
Huawei Technologies
M. Jethanandani, Ed.
Xoriant Corporation
S. Pallagatti
Rtbrick
G. Mirsky
ZTE Corporation

August 1, 2018

YANG Data Model for Bidirectional Forwarding Detection (BFD) draft-ietf-bfd-yang-17

Abstract

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

The YANG modules in this document conform to the Network Management Datastore Architecture (NMDA).

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 February 2, 2019.

Table of Contents

1. Introduction 3
1.1. Requirements Language 4
1.2. Tree Diagrams 4
2. Design of the Data Model 4
2.1. Design of Configuration Model 5
2.1.1. Common BFD configuration parameters 6
2.1.2. Single-hop IP 7
2.1.3. Multihop IP 7
2.1.4. MPLS Traffic Engineering Tunnels 8
2.1.5. MPLS Label Switched Paths 9
2.1.6. Link Aggregation Groups 9
2.2. Design of Operational State Model 9
2.3. Notifications 10
2.4. RPC Operations 10
2.5. BFD top level hierarchy 10
2.6. BFD IP single-hop hierarchy 10
2.7. BFD IP multihop hierarchy 12
2.8. BFD over LAG hierarchy 14
2.9. BFD over MPLS LSPs hierarchy 18
2.10. BFD over MPLS-TE hierarchy 20
2.11. Interaction with other YANG modules 22
2.11.1. Module ietf-interfaces 22
2.11.2. Module ietf-ip 22
2.11.3. Module ietf-mpls 23
2.11.4. Module ietf-te 23
2.12. IANA BFD YANG Module 23
2.13. BFD types YANG Module 26
2.14. BFD top-level YANG Module 39
2.15. BFD IP single-hop YANG Module 41
2.16. BFD IP multihop YANG Module 44
2.17. BFD over LAG YANG Module 47
2.18. BFD over MPLS YANG Module 51
2.19. BFD over MPLS-TE YANG Module 55
3. Data Model examples 58
3.1. IP single-hop 58
3.2. IP multihop 59
3.3. LAG 60
3.4. MPLS 61

4. Security Considerations 62
5. IANA Considerations 66
5.1. IANA-Maintained iana-bfd-types module 70
6. Acknowledgements 70
7. References 70
7.1. Normative References 70
7.2. Informative References 73
Appendix A. Echo function configuration example 73
A.1. Example YANG module for BFD echo function configuration . 74
Appendix B. Change log 76
B.1. Changes between versions -16 and -17 76
B.2. Changes between versions -15 and -16 76
B.3. Changes between versions -14 and -15 76
B.4. Changes between versions -13 and -14 76
B.5. Changes between versions -12 and -13 76
B.6. Changes between versions -11 and -12 76
B.7. Changes between versions -10 and -11 76
B.8. Changes between versions -09 and -10 77
B.9. Changes between versions -08 and -09 77
B.10. Changes between versions -07 and -08 77
B.11. Changes between versions -06 and -07 77
B.12. Changes between versions -05 and -06 77
B.13. Changes between versions -04 and -05 78
B.14. Changes between versions -03 and -04 78
B.15. Changes between versions -02 and -03 78
B.16. Changes between versions -01 and -02 78
B.17. Changes between versions -00 and -01 78
Authors' Addresses 78

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.

The YANG modules in this document conform to the Network Management Datastore Architecture (NMDA) [RFC8342]. This means that the data models do not have separate top-level or sibling containers for configuration and operational state data.

1.1. 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 BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

1.2. Tree Diagrams

This document uses the graphical representation of data models defined in [RFC8340].

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, and 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 (NMDA Version)" [RFC8349]. This new "bfd" container is augmented by all the YANG modules for their respective specific information:

1. ietf-bfd-ip-sh.yang augments "/routing/control-plane-protocols/control-plane-protocol/bfd/" with the "ip-sh" container for BFD sessions over IP single-hop.

2. ietf-bfd-ip-mh.yang augments "/routing/control-plane-protocols/control-plane-protocol/bfd/" with the "ip-mh" container for BFD sessions over IP multi-hop.

3. ietf-bfd-lag.yang augments "/routing/control-plane-protocols/control-plane-protocol/bfd/" with the "lag" container for BFD sessions over LAG.

4. ietf-bfd-mpls.yang augments "/routing/control-plane-protocols/control-plane-protocol/bfd/" with the "mpls" container for BFD over MPLS LSPs.

5. ietf-bfd-mpls-te.yang augments "/routing/control-plane-protocols/control-plane-protocol/bfd/" with the "mpls-te" container for BFD over MPLS-TE.

BFD can operate in the following contexts:

1. At the network device level

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

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

When used at the network device level, the BFD YANG model is used "as-is". When the BFD YANG model is used in a Logical Network Element or in a Network Instance, then the BFD YANG model augments the mounted routing model for the Logical Network Element or the Network Instance.

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. For example, when a new IGP peer is discovered, the IGP would create a BFD session to the newly discovered peer and similarly when an IGP peer goes away, the IGP would remove the BFD session to that peer. The mechanism how the BFD sessions are created and removed by the BFD clients is outside the scope of this document, but typically this would be done by use of an API implemented by the BFD module on the system. 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. Multihop IP

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

Because of multiple paths, there could be multiple multihop 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 are 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 State Model

The operational state 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 state data 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 multihop 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 end-users 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 state data. The operational state data consist of overall BFD session statistics, i.e. for BFD on all types of forwarding paths.

```
module: ietf-bfd
  augment /rt:routing/rt:control-plane-protocols
            /rt:control-plane-protocol:
    +--rw bfd
       +--ro summary
          +--ro number-of-sessions?             yang:gauge32
          +--ro number-of-sessions-up?          yang:gauge32
          +--ro number-of-sessions-down?        yang:gauge32
          +--ro number-of-sessions-admin-down?  yang:gauge32
```

2.6. BFD IP single-hop hierarchy

An "ip-sh" node is added under "bfd" node in control-plane-protocol. The configuration and operational state data for each BFD IP single-hop session is under this "ip-sh" node.

```
module: ietf-bfd-ip-sh
```

```
     augment /rt:routing/rt:control-plane-protocols
               /rt:control-plane-protocol/bfd:bfd:
       +--rw ip-sh
          +--ro summary
          |  +--ro number-of-sessions?              yang:gauge32
          |  +--ro number-of-sessions-up?           yang:gauge32
          |  +--ro number-of-sessions-down?         yang:gauge32
          |  +--ro number-of-sessions-admin-down?   yang:gauge32
          +--rw sessions
          |  +--rw session* [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) {single-minimum-interval}?
          |     |     +--rw min-interval?              uint32
          |     +--rw demand-enabled?                    boolean
          |     |       {demand-mode}?
          |     +--rw admin-down?                        boolean
          |     +--rw authentication! {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-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 session-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?                     yang:counter32
          |        +--ro admin-down-count?               yang:counter32
          |        +--ro receive-packet-count?           yang:counter64
          |        +--ro send-packet-count?              yang:counter64
          |        +--ro receive-invalid-packet-count?   yang:counter64
          |        +--ro send-failed-packet-count?       yang:counter64
          +--rw interfaces* [interface]
             +--rw interface         if:interface-ref
             +--rw authentication! {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 multihop hierarchy

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

```
module: ietf-bfd-ip-mh
```

```
     augment /rt:routing/rt:control-plane-protocols
               /rt:control-plane-protocol/bfd:bfd:
       +--rw ip-mh
          +--ro summary
          |  +--ro number-of-sessions?              yang:gauge32
          |  +--ro number-of-sessions-up?           yang:gauge32
          |  +--ro number-of-sessions-down?         yang:gauge32
          |  +--ro number-of-sessions-admin-down?   yang:gauge32
          +--rw session-groups
             +--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) {single-minimum-interval}?
                |     +--rw min-interval?                uint32
                +--rw demand-enabled?                   boolean
                |       {demand-mode}?
                +--rw admin-down?                       boolean
                +--rw authentication! {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 session-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?
                   |       yang:counter32
                   +--ro admin-down-count?
                   |       yang:counter32
                   +--ro receive-packet-count?
                   |       yang:counter64
                   +--ro send-packet-count?
                   |       yang:counter64
                   +--ro receive-invalid-packet-count?
                   |       yang:counter64
                   +--ro send-failed-packet-count?
                           yang:counter64

  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 state data for each BFD LAG session is under this "lag" node.

```
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 summary
          |     +--ro number-of-sessions?              yang:gauge32
          |     +--ro number-of-sessions-up?           yang:gauge32
          |     +--ro number-of-sessions-down?         yang:gauge32
          |     +--ro number-of-sessions-admin-down?   yang:gauge32
          +--rw micro-bfd-ipv6-session-statistics
          |  +--ro summary
          |     +--ro number-of-sessions?              yang:gauge32
          |     +--ro number-of-sessions-up?           yang:gauge32
          |     +--ro number-of-sessions-down?         yang:gauge32
          |     +--ro number-of-sessions-admin-down?   yang:gauge32
          +--rw sessions
             +--rw session* [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) {single-minimum-interval}?
                |     +--rw min-interval?                uint32
                +--rw demand-enabled?                   boolean
                |       {demand-mode}?
                +--rw admin-down?                       boolean
                +--rw authentication! {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 session-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?
|     |       yang:counter32
|     +--ro admin-down-count?
|     |       yang:counter32
|     +--ro receive-packet-count?
|     |       yang:counter64
|     +--ro send-packet-count?
|     |       yang:counter64
|     +--ro receive-invalid-packet-count?
|     |       yang:counter64
|     +--ro send-failed-packet-count?
|             yang:counter64
+--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 session-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?
                         |       yang:counter32
                         +--ro admin-down-count?
                         |       yang:counter32
                         +--ro receive-packet-count?
                         |       yang:counter64
                         +--ro send-packet-count?
                         |       yang:counter64
                         +--ro receive-invalid-packet-count?
                         |       yang:counter64
                         +--ro send-failed-packet-count?
                                 yang:counter64

    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 state 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 summary
       |  +--ro number-of-sessions?              yang:gauge32
       |  +--ro number-of-sessions-up?           yang:gauge32
       |  +--ro number-of-sessions-down?         yang:gauge32
       |  +--ro number-of-sessions-admin-down?   yang:gauge32
       +--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) {single-minimum-interval}?
       |  |     +--rw min-interval?               uint32
       |  +--rw authentication! {authentication}?
       |     +--rw key-chain?    kc:key-chain-ref
       |     +--rw meticulous?   boolean
       +--rw session-groups
          +--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) {single-minimum-interval}?
             |     +--rw min-interval?               uint32
             +--rw demand-enabled?                   boolean
             |       {demand-mode}?
             +--rw admin-down?                       boolean
             +--rw authentication! {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 session-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?
                |  |       yang:counter32
                |  +--ro admin-down-count?
                |  |       yang:counter32
                |  +--ro receive-packet-count?
                |  |       yang:counter64
                |  +--ro send-packet-count?
                |  |       yang:counter64
                |  +--ro receive-invalid-packet-count?
                |  |       yang:counter64
                |  +--ro send-failed-packet-count?
                |          yang:counter64
                +--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 state 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) {single-minimum-interval}?
       |  |     +--rw min-interval?              uint32
       |  +--rw authentication! {authentication}?
       |     +--rw key-chain?    kc:key-chain-ref
       |     +--rw meticulous?   boolean
       +--ro summary
          +--ro number-of-sessions?             yang:gauge32
          +--ro number-of-sessions-up?          yang:gauge32
          +--ro number-of-sessions-down?        yang:gauge32
          +--ro number-of-sessions-admin-down?  yang:gauge32
  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) {single-minimum-interval}?
    |     +--rw min-interval?              uint32
    +--rw demand-enabled?                  boolean {demand-mode}?
```

```
       +--rw admin-down?                        boolean
       +--rw authentication! {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 session-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?                     yang:counter32
       |  +--ro admin-down-count?               yang:counter32
       |  +--ro receive-packet-count?           yang:counter64
       |  +--ro send-packet-count?              yang:counter64
       |  +--ro receive-invalid-packet-count?   yang:counter64
       |  +--ro send-failed-packet-count?       yang:counter64
       +--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 [RFC8343]:

/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 [RFC8344]:

/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@2018-08-01.yang"

module iana-bfd-types {

  yang-version 1.1;

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

  prefix "iana-bfd-types";

  organization "IANA";

  contact
    "        Internet Assigned Numbers Authority

    Postal: ICANN
            12025 Waterfront Drive, Suite 300
            Los Angeles, CA 90094-2536
            United States of America

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

  description
    "This module defines YANG data types for IANA-registered
     BFD parameters.
```

```
       This YANG module is maintained by IANA and reflects the
       'BFD Diagnostic Codes' and 'BFD Authentication Types' registries.

       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
       (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.";

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

   reference "RFC XXXX";

   revision 2018-08-01 {
     description "Initial revision.";
     reference "RFC XXXX: IANA BFD YANG Data Types.";
   }

   /*
    * Type Definitions
    */
   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 RFC 5880, values are maintained in
        the 'BFD Diagnostic Codes' IANA registry. 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 RFC 5880, values are
         maintained in the 'BFD Authentication Types' IANA registry.
         Range is 0 to 255.";
    }
  }

  <CODE ENDS>
```

2.13. BFD types YANG Module

This YANG module imports typedefs from [RFC6991], [RFC8177] and the "control-plane-protocol" identity from [RFC8349].

```
<CODE BEGINS> file "ietf-bfd-types@2018-08-01.yang"

module ietf-bfd-types {

  yang-version 1.1;

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

  prefix "bfd-types";

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

  import iana-bfd-types {
    prefix "iana-bfd-types";
    reference "RFC XXXX: YANG Data Model for BFD";
  }

  import ietf-inet-types {
    prefix "inet";
    reference "RFC 6991: Common YANG Data Types";
  }

  import ietf-yang-types {
    prefix "yang";
    reference "RFC 6991: Common YANG Data Types";
```

```
  }

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

  import ietf-key-chain {
    prefix "kc";
    reference "RFC 8177: YANG Data Model for Key Chains";
  }

  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 RFC 5880, and also groupings which are common
     to other BFD YANG modules.

     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
     (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.";

  reference "RFC XXXX";

  revision 2018-08-01 {
    description "Initial revision.";
    reference "RFC XXXX: YANG Data Model for BFD";
  }
```

```
  /*
   * Feature definitions
   */
  feature single-minimum-interval {
    description
      "This feature indicates that the server supports configuration
       of one minimum interval value which is used for both transmit and
       receive minimum intervals.";
  }

  feature authentication {
    description
      "This feature indicates that the server supports BFD
       authentication.";
    reference
      "RFC 5880: Bidirectional Forwarding Detection (BFD),
       section 6.7.";
  }

  feature demand-mode {
    description
      "This feature indicates that the server supports BFD demand
       mode.";
    reference
      "RFC 5880: Bidirectional Forwarding Detection (BFD),
       section 6.6.";
  }

  feature echo-mode {
    description
      "This feature indicates that the server supports BFD echo
       mode.";
    reference
      "RFC 5880: Bidirectional Forwarding Detection (BFD),
       section 6.4.";
  }

  /*
   * Identity definitions
   */
  identity bfdv1 {
    base "rt:control-plane-protocol";
    description "BFD protocol version 1.";
    reference
      "RFC 5880: Bidirectional Forwarding Detection (BFD).";
  }

  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.";
    reference
      "RFC 5881: Bidirectional Forwarding Detection (BFD)
       for IPv4 and IPv6 (Single Hop).";
  }
  identity path-ip-mh {
    base path-type;
    description "BFD on IP multihop paths.";
    reference
      "RFC 5883: Bidirectional Forwarding Detection (BFD) for
       Multihop Paths.";
  }
  identity path-mpls-te {
    base path-type;
    description
      "BFD on MPLS Traffic Engineering.";
    reference
      "RFC 5884: Bidirectional Forwarding Detection (BFD)
       for MPLS Label Switched Paths (LSPs).";
  }
  identity path-mpls-lsp {
    base path-type;
    description
      "BFD on MPLS Label Switched Path.";
    reference
      "RFC 5884: Bidirectional Forwarding Detection (BFD)
       for MPLS Label Switched Paths (LSPs).";
  }
  identity path-lag {
    base path-type;
    description
      "Micro-BFD on LAG member links.";
    reference
      "RFC 7130: Bidirectional Forwarding Detection (BFD) on
       Link Aggregation Group (LAG) Interfaces.";
  }

  identity encap-type {
    description
      "Base identity for BFD encapsulation type.";
  }
  identity encap-ip {
```

```
    base encap-type;
    description "BFD with IP encapsulation.";
  }

  /*
   * Type Definitions
   */
  typedef discriminator {
    type uint32;
    description "BFD discriminator as described in RFC 5880.";
  }

  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 RFC 5880.";
  }

  typedef multiplier {
    type uint8 {
      range 1..255;
    }
    description "BFD multiplier as described in RFC 5880.";
  }

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

```
  }

  /*
   * Groupings
   */
  grouping auth-parms {
    description
      "Grouping for BFD authentication parameters
       (see section 6.7 of RFC 5880).";
    container authentication {
      if-feature authentication;
      presence
        "Enables BFD authentication (see section 6.7 of RFC 5880).";
      description "Parameters for BFD authentication.";

      leaf key-chain {
        type kc:key-chain-ref;
        description "Name of the key-chain as per RFC 8177.";
      }

      leaf meticulous {
        type boolean;
        description
          "Enables meticulous mode as described in section 6.7 " +
          "of RFC 5880.";
      }
    }
  }

  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 one value used for both transmit and
         receive.";
      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 {
        if-feature single-minimum-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 configuration 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 configuration 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 session-statistics {
      config "false";
      description "BFD per-session statistics.";

      leaf create-time {
        type yang:date-and-time;
        description
          "Time and date when this session was created.";
      }
      leaf last-down-time {
        type yang:date-and-time;
        description
          "Time and date of last time this session went down.";
      }
      leaf last-up-time {
        type yang:date-and-time;
        description
          "Time and date of last time this session went up.";
      }
      leaf down-count {
        type yang:counter32;
        description
          "The number of times this session has transitioned in the
           down state.";
      }
      leaf admin-down-count {
        type yang:counter32;
        description
          "The number of times this session has transitioned in the
           admin-down state.";
      }
      leaf receive-packet-count {
        type yang:counter64;
```

```
        description
          "Count of received packets in this session. This includes
           valid and invalid received packets.";
      }
      leaf send-packet-count {
        type yang:counter64;
        description "Count of sent packets in this session.";
      }
      leaf receive-invalid-packet-count {
        type yang:counter64;
        description
          "Count of invalid received packets in this session.";
      }
      leaf send-failed-packet-count {
        type yang:counter64;
        description
          "Count of packets which failed to be sent in this session.";
      }
    }
  }

  grouping session-statistics-summary {
    description "Grouping for session statistics summary.";
    container summary {
      config false;
      description "BFD session statistics summary.";
      leaf number-of-sessions {
        type yang:gauge32;
        description "Number of BFD sessions.";
      }
      leaf number-of-sessions-up {
        type yang:gauge32;
        description
          "Number of BFD sessions currently in up state (as defined
           in RFC 5880).";
      }
      leaf number-of-sessions-down {
        type yang:gauge32;
        description
          "Number of BFD sessions currently in down or init state
           but not admin-down (as defined in RFC 5880).";
      }
      leaf number-of-sessions-admin-down {
        type yang:gauge32;
        description
          "Number of BFD sessions currently in admin-down state (as
           defined in RFC 5880).";
      }
```

```
    }
  }

  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

This YANG module imports and augments "/routing/control-plane-protocols/control-plane-protocol" from [RFC8349].

<CODE BEGINS> file "ietf-bfd@2018-08-01.yang"

```
module ietf-bfd {

  yang-version 1.1;

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

  prefix "bfd";

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

  import ietf-bfd-types {
    prefix "bfd-types";
    reference "RFC XXXX: YANG Data Model for 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:   <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 RFC 5880.

       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
       (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.";

    reference "RFC XXXX";

    revision 2018-08-01 {
      description "Initial revision.";
      reference "RFC XXXX: YANG Data Model for BFD";
    }

    augment "/rt:routing/rt:control-plane-protocols/"
          + "rt:control-plane-protocol" {
       when "derived-from-or-self(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-summary;
      }
    }
  }

  <CODE ENDS>
```

2.15. BFD IP single-hop YANG Module

This YANG module imports "interface-ref" from [RFC8343], typedefs from [RFC6991] and augments "/routing/control-plane-protocols/control-plane-protocol" from [RFC8349].

<CODE BEGINS> file "ietf-bfd-ip-sh@2018-08-01.yang"

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

  yang-version 1.1;

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

  prefix "bfd-ip-sh";

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

  import ietf-bfd-types {
    prefix "bfd-types";
    reference "RFC XXXX: YANG Data Model for BFD";
  }

  import ietf-bfd {
    prefix "bfd";
    reference "RFC XXXX: YANG Data Model for BFD";
  }

  import ietf-interfaces {
    prefix "if";
    reference
      "RFC 8343: A YANG Data Model for Interface Management";
  }

  import ietf-inet-types {
    prefix "inet";
    reference "RFC 6991: Common YANG Data Types";
  }

  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:   <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 RFC 5881.

        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
        (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.";

     reference "RFC XXXX";

     revision 2018-08-01 {
       description "Initial revision.";
       reference "RFC XXXX: A YANG data model for BFD IP single-hop";
     }

     /*
      * Augments
      */
     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-summary;

         container sessions {
           description
             "BFD IP single-hop sessions.";
           list session {
             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 IP 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;
        }
      }
    }

    /*
     * Notifications
     */
    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 multihop YANG Module

This YANG module imports typedefs from [RFC6991] and augments "/routing/control-plane-protocols/control-plane-protocol" from [RFC8349].

```
 <CODE BEGINS> file "ietf-bfd-ip-mh@2018-08-01.yang"

 module ietf-bfd-ip-mh {

   yang-version 1.1;

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

   prefix "bfd-ip-mh";

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

   import ietf-bfd-types {
     prefix "bfd-types";
     reference "RFC XXXX: YANG Data Model for BFD";
   }

   import ietf-bfd {
     prefix "bfd";
     reference "RFC XXXX: YANG Data Model for BFD";
   }

   import ietf-inet-types {
     prefix "inet";
     reference "RFC 6991: Common YANG Data Types";
   }

   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:   <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 RFC 5883.

     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
     (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.";

  reference "RFC XXXX";

  revision 2018-08-01 {
    description "Initial revision.";
    reference "RFC XXXX: A YANG data model for BFD IP multihop.";
  }

  /*
   * Augments
   */
  augment "/rt:routing/rt:control-plane-protocols/"
        + "rt:control-plane-protocol/bfd:bfd" {
    description "BFD augmentation for IP multihop.";
    container ip-mh {
      description "BFD IP multihop top level container.";
```

```
      uses bfd-types:session-statistics-summary;

      container session-groups {
        description
          "BFD IP multi-hop session groups.";
        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;
          }
        }
      }
    }
```

```
   }

   /*
    * Notifications
    */
   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

This YANG module imports "interface-ref" from [RFC8343], typedefs from [RFC6991] and augments "/routing/control-plane-protocols/control-plane-protocol" from [RFC8349].

```
  <CODE BEGINS> file "ietf-bfd-lag@2018-08-01.yang"

  module ietf-bfd-lag {

    yang-version 1.1;

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

    prefix "bfd-lag";

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

    import ietf-bfd-types {
      prefix "bfd-types";
      reference "RFC XXXX: YANG Data Model for BFD";
    }

    import ietf-bfd {
      prefix "bfd";
      reference "RFC XXXX: YANG Data Model for BFD";
    }

    import ietf-interfaces {
      prefix "if";
```

```
      reference
        "RFC 8343: A YANG Data Model for Interface Management";
    }

    import ietf-inet-types {
      prefix "inet";
      reference "RFC 6991: Common YANG Data Types";
    }

    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:   <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.

       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
       (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.";

    reference "RFC XXXX";

    revision 2018-08-01 {
      description "Initial revision.";
      reference "RFC XXXX: A YANG data model for BFD over LAG";
```

```
    }

    /*
     * Augments
     */
    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-summary;
        }
        container micro-bfd-ipv6-session-statistics {
          description "Micro-BFD IPv6 session counters.";
          uses bfd-types:session-statistics-summary;
        }

        container sessions {
          description
            "BFD over LAG sessions";
          list session {
            key "lag-name";
            description "List of BFD over LAG sessions.";
            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;
            }
          }
        }
      }
    }

    /*
     * Notifications
     */
    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

This YANG module imports typedefs from [RFC6991] and augments "/routing/control-plane-protocols/control-plane-protocol" from [RFC8349].

```
  <CODE BEGINS> file "ietf-bfd-mpls@2018-08-01.yang"

  module ietf-bfd-mpls {

    yang-version 1.1;

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

    prefix "bfd-mpls";

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

    import ietf-bfd-types {
      prefix "bfd-types";
      reference "RFC XXXX: YANG Data Model for BFD";
    }

    import ietf-bfd {
      prefix "bfd";
      reference "RFC XXXX: YANG Data Model for BFD";
    }

    import ietf-inet-types {
      prefix "inet";
      reference "RFC 6991: Common YANG Data Types";
    }

    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:   <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 RFC 5884.

       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
       (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.";

    reference "RFC XXXX";

    revision 2018-08-01 {
      description "Initial revision.";
      reference "RFC XXXX: A YANG data model for BFD over MPLS LSPs";
    }

    /*
     * Identity definitions
     */
    identity encap-gach {
      base bfd-types:encap-type;
      description
        "BFD with G-ACh encapsulation as per RFC 5586.";
    }
```

```
    identity encap-ip-gach {
      base bfd-types:encap-type;
      description
        "BFD with IP and G-ACh encapsulation as per RFC 5586.";
    }

    /*
     * Groupings
     */
    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 RFC 5884.";

      leaf mpls-dest-address {
        type inet:ip-address;
        config "false";
        description
          "Destination address as per RFC 5884.
           Needed if IP encapsulation is used.";
      }
    }

    /*
     * Augments
     */
    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-summary;

        container egress {
          description "Egress configuration.";

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

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

        container session-groups {
          description
            "BFD over MPLS session groups.";
          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;
            }
          }
        }
      }

      /*
       * Notifications
       */
      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 RFC 5884.
           Needed if IP encapsulation is used.";
      }
    }
  }
```

<CODE ENDS>

2.19. BFD over MPLS-TE YANG Module

This YANG module imports and augments "/te/tunnels/tunnel" from [I-D.ietf-teas-yang-te].

<CODE BEGINS> file "ietf-bfd-mpls-te@2018-08-01.yang"

```
module ietf-bfd-mpls-te {

  yang-version 1.1;

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

  prefix "bfd-mpls-te";

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

  import ietf-bfd-types {
    prefix "bfd-types";
    reference "RFC XXXX: YANG Data Model for BFD";
  }

  import ietf-bfd {
    prefix "bfd";
    reference "RFC XXXX: YANG Data Model for BFD";
  }

  import ietf-bfd-mpls {
    prefix "bfd-mpls";
    reference "RFC XXXX: YANG Data Model for BFD";
  }

  import ietf-te {
    prefix "te";
    // RFC Ed.: replace YYYY with actual RFC number of
    // draft-ietf-teas-yang-te and remove this note.
    reference
      "RFC YYYY: A YANG Data Model for Traffic Engineering Tunnels and
       Interfaces";
```

```
    }

    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:   <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 RFC 5884.

       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
       (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.";

    reference "RFC XXXX";

    revision 2018-08-01 {
      description "Initial revision.";
      reference "RFC XXXX: A YANG data model for BFD over MPLS-TE";
    }

    /*
     * Augments
     */
    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-summary;
      }
    }

    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;
    }

    /*
     * Notifications
     */
    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>

3. Data Model examples

This section presents some simple and illustrative examples on how to configure BFD.

3.1. IP single-hop

The following is an example configuration for a BFD IP single-hop session. The desired transmit interval and the required receive interval are both set to 10ms.

```
<?xml version="1.0" encoding="UTF-8"?>
<config xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <interfaces xmlns="urn:ietf:params:xml:ns:yang:ietf-interfaces">
    <interface>
      <name>eth0</name>
      <type xmlns:ianaift="urn:ietf:params:xml:ns:yang:iana-if-type">
        ianaift:ethernetCsmacd
      </type>
    </interface>
  </interfaces>
  <routing xmlns="urn:ietf:params:xml:ns:yang:ietf-routing">
    <control-plane-protocols>
      <control-plane-protocol>
        <type xmlns:bfd-types=
            "urn:ietf:params:xml:ns:yang:ietf-bfd-types">
          bfd-types:bfdv1
        </type>
        <name>name:BFD</name>
        <bfd xmlns="urn:ietf:params:xml:ns:yang:ietf-bfd">
          <ip-sh xmlns="urn:ietf:params:xml:ns:yang:ietf-bfd-ip-sh">
            <sessions>
              <session>
                <interface>eth0</interface>
                <dest-addr>2001:db8:0:113::101</dest-addr>
                <desired-min-tx-interval>10000</desired-min-tx-interval>
                <required-min-rx-interval>
                  10000
                </required-min-rx-interval>
              </session>
            </sessions>
          </ip-sh>
        </bfd>
      </control-plane-protocol>
    </control-plane-protocols>
  </routing>
</config>
```

3.2. IP multihop

The following is an example configuration for a BFD IP multihop session group. The desired transmit interval and the required receive interval are both set to 150ms.

```
<?xml version="1.0" encoding="UTF-8"?>
<config xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <routing xmlns="urn:ietf:params:xml:ns:yang:ietf-routing">
    <control-plane-protocols>
      <control-plane-protocol>
        <type xmlns:bfd-types=
            "urn:ietf:params:xml:ns:yang:ietf-bfd-types">
          bfd-types:bfdv1
        </type>
        <name>name:BFD</name>
        <bfd xmlns="urn:ietf:params:xml:ns:yang:ietf-bfd">
          <ip-mh xmlns="urn:ietf:params:xml:ns:yang:ietf-bfd-ip-mh">
            <session-groups>
              <session-group>
                <source-addr>2001:db8:0:113::103</source-addr>
                <dest-addr>2001:db8:0:114::100</dest-addr>
                <desired-min-tx-interval>
                  150000
                </desired-min-tx-interval>
                <required-min-rx-interval>
                  150000
                </required-min-rx-interval>
                <rx-ttl>240</rx-ttl>
              </session-group>
            </session-groups>
          </ip-mh>
        </bfd>
      </control-plane-protocol>
    </control-plane-protocols>
  </routing>
</config>
```

3.3. LAG

The following is an example of BFD configuration for a LAG session. In this case, an interface named "Bundle-Ether1" of interface type "ieee802eadLag" has a desired transmit and required receive interval set to 10ms.

```
<?xml version="1.0" encoding="UTF-8"?>
<config xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <interfaces xmlns="urn:ietf:params:xml:ns:yang:ietf-interfaces">
    <interface>
      <name>Bundle-Ether1</name>
      <type xmlns:ianaift="urn:ietf:params:xml:ns:yang:iana-if-type">
        ianaift:ieee8023adLag
      </type>
    </interface>
  </interfaces>
  <routing xmlns="urn:ietf:params:xml:ns:yang:ietf-routing">
    <control-plane-protocols>
      <control-plane-protocol>
        <type xmlns:bfd-types=
            "urn:ietf:params:xml:ns:yang:ietf-bfd-types">
          bfd-types:bfdv1
        </type>
        <name>name:BFD</name>
        <bfd xmlns="urn:ietf:params:xml:ns:yang:ietf-bfd">
          <lag xmlns="urn:ietf:params:xml:ns:yang:ietf-bfd-lag">
            <sessions>
              <session>
                <lag-name>Bundle-Ether1</lag-name>
                <ipv6-dest-addr>2001:db8:112::16</ipv6-dest-addr>
                <desired-min-tx-interval>
                  100000
                </desired-min-tx-interval>
                <required-min-rx-interval>
                  100000
                </required-min-rx-interval>
                <use-ipv6>true</use-ipv6>
              </session>
            </sessions>
          </lag>
        </bfd>
      </control-plane-protocol>
    </control-plane-protocols>
  </routing>
</config>
```

3.4. MPLS

The following is an example of BFD configured for an MPLS LSP. In this case, the desired transmit and required receive interval set to 250ms.

```
<?xml version="1.0" encoding="UTF-8"?>
<config xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <routing xmlns="urn:ietf:params:xml:ns:yang:ietf-routing">
    <control-plane-protocols>
      <control-plane-protocol>
        <type xmlns:bfd-types=
            "urn:ietf:params:xml:ns:yang:ietf-bfd-types">
          bfd-types:bfdv1
        </type>
        <name>name:BFD</name>
        <bfd xmlns="urn:ietf:params:xml:ns:yang:ietf-bfd">
          <mpls xmlns="urn:ietf:params:xml:ns:yang:ietf-bfd-mpls">
            <session-groups>
              <session-group>
                <mpls-fec>2001:db8:114::/116</mpls-fec>
                <desired-min-tx-interval>
                  250000
                </desired-min-tx-interval>
                <required-min-rx-interval>
                  250000
                </required-min-rx-interval>
              </session-group>
            </session-groups>
          </mpls>
        </bfd>
      </control-plane-protocol>
    </control-plane-protocols>
  </routing>
</config>
```

4. 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 [RFC5246].

The NETCONF access control model [RFC6536] 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/sessions: the list specifies the IP single-hop BFD sessions.

/routing/control-plane-protocols/control-plane-protocol/bfd/ip-sh/sessions: data nodes local-multiplier, desired-min-tx-interval, required-min-rx-interval and min-interval all impact the BFD IP single-hop session. The source-addr and dest-addr data nodes can be used to send BFD packets to unwitting recipients, [RFC5880] describes how BFD mitigates against such threats. Authentication data nodes key-chain and meticulous impact the security of the BFD IP single-hop session.

/routing/control-plane-protocols/control-plane-protocol/bfd/ip-mh/session-group: the list specifies the IP multi-hop BFD session groups.

/routing/control-plane-protocols/control-plane-protocol/bfd/ip-mh/session-group: data nodes local-multiplier, desired-min-tx-interval, required-min-rx-interval and min-interval all impact the BFD IP multi-hop session. The source-addr and dest-addr data nodes can be used to send BFD packets to unwitting recipients, [RFC5880] describes how BFD mitigates against such threats. Authentication data nodes key-chain and meticulous impact the security of the BFD IP multi-hop session.

/routing/control-plane-protocols/control-plane-protocol/bfd/lag/sessions: the list specifies the BFD sessions over LAG.

/routing/control-plane-protocols/control-plane-protocol/bfd/lag/sessions: data nodes local-multiplier, desired-min-tx-interval, required-min-rx-interval and min-interval all impact the BFD over LAG session. The ipv4-dest-addr and ipv6-dest-addr data nodes can be used to send BFD packets to unwitting recipients, [RFC5880] describes how BFD mitigates against such threats. Authentication data nodes key-chain and meticulous impact the security of the BFD over LAG session.

/routing/control-plane-protocols/control-plane-protocol/bfd/mpls/session-group: the list specifies the session groups for BFD over MPLS.

/routing/control-plane-protocols/control-plane-protocol/bfd/mpls/session-group: data nodes local-multiplier, desired-min-tx-interval, required-min-rx-interval, and min-interval all impact the BFD over MPLS LSPs session. Authentication data nodes key-chain and meticulous impact the security of the BFD over MPLS LSPs session.

/routing/control-plane-protocols/control-plane-protocol/bfd/mpls/egress: data nodes local-multiplier, desired-min-tx-interval, required-min-rx-interval and min-interval all impact the BFD over MPLS LSPs sessions for which this device is an MPLS LSP egress node. Authentication data nodes key-chain and meticulous impact the security of the BFD over MPLS LSPs sessions for which this device is an MPLS LSP egress node

/te/tunnels/tunnel: data nodes local-multiplier, desired-min-tx-interval, required-min-rx-interval and min-interval all impact the BFD session over the MPLS-TE tunnel. Authentication data nodes key-chain and meticulous impact the security of the BFD session over the MPLS-TE tunnel.

/routing/control-plane-protocols/control-plane-protocol/bfd/mpls-te/egress: data nodes local-multiplier, desired-min-tx-interval, required-min-rx-interval and min-interval all impact the BFD over MPLS-TE sessions for which this device is an MPLS-TE egress node. Authentication data nodes key-chain and meticulous impact the security of the BFD over MPLS-TE sessions for which this device is an MPLS-TE egress node.

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.

When BFD clients are used to modify BFD configuration (as described in Section 2.1), the BFD clients need to be included in an analysis of the security properties of the BFD-using system (e.g., when considering the authentication and authorization of control actions). In many cases, BFD is not the most vulnerable portion of such a composite system, since BFD is limited to generating well-defined traffic at a fixed rate on a given path; in the case of an IGP as BFD client, attacking the IGP could cause more broad-scale disruption than (de)configuring a BFD session could cause.

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/ summary: access to this information discloses the number of BFD IP single-hop sessions which are up, down and admin-down. The counters include BFD sessions for which the user does not have read-access.

/routing/control-plane-protocols/control-plane-protocol/bfd/ip-sh/sessions/session/: access to data nodes local-discriminator and remote-discriminator (combined with the data nodes in the authentication container) provides the ability to spoof BFD IP single-hop packets.

/routing/control-plane-protocols/control-plane-protocol/bfd/ip-mh/ summary: access to this information discloses the number of BFD IP multi-hop sessions which are up, down and admin-down. The counters include BFD sessions for which the user does not have read-access.

/routing/control-plane-protocols/control-plane-protocol/bfd/ip-mh/ session-groups/session-group/sessions: access to data nodes local-discriminator and remote-discriminator (combined with the data nodes in the session-group's authentication container) provides the ability to spoof BFD IP multi-hop packets.

/routing/control-plane-protocols/control-plane-protocol/bfd/lag/ micro-bfd-ipv4-session-statistics/summary: access to this information discloses the number of micro BFD IPv4 LAG sessions which are up, down and admin-down. The counters include BFD sessions for which the user does not have read-access.

/routing/control-plane-protocols/control-plane-protocol/bfd/lag/sessions/session/member-links/member-link/micro-bfd-ipv4: access to data nodes local-discriminator and remote-discriminator (combined with the data nodes in the session's authentication container) provides the ability to spoof BFD IPv4 LAG packets.

/routing/control-plane-protocols/control-plane-protocol/bfd/lag/ micro-bfd-ipv6-session-statistics/summary: access to this information discloses the number of micro BFD IPv6 LAG sessions which are up, down and admin-down. The counters include BFD sessions for which the user does not have read-access.

/routing/control-plane-protocols/control-plane-protocol/bfd/lag/sessions/session/member-links/member-link/micro-bfd-ipv6: access to data nodes local-discriminator and remote-discriminator (combined with the data nodes in the session's

authentication container) provides the ability to spoof BFD IPv6 LAG packets.

/routing/control-plane-protocols/control-plane-protocol/bfd/mpls/ summary: access to this information discloses the number of BFD sessions over MPLS LSPs which are up, down and admin-down. The counters include BFD sessions for which the user does not have read-access.

/routing/control-plane-protocols/control-plane-protocol/bfd/mpls/ session-groups/session-group/sessions: access to data nodes local-discriminator and remote-discriminator (combined with the data nodes in the session-group's authentication container) provides the ability to spoof BFD over MPLS LSPs packets.

/routing/control-plane-protocols/control-plane-protocol/bfd/mpls-te/ summary: access to this information discloses the number of BFD sessions over MPLS-TE which are up, down and admin-down. The counters include BFD sessions for which the user does not have read-access.

/te/lsps-state/lsp: access to data nodes local-discriminator and remote-discriminator (combined with the data nodes in the tunnel's authentication container) provides the ability to spoof BFD over MPLS-TE packets.

5. IANA Considerations

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

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

Registrant Contact: The IESG.

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

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

Registrant Contact: The IESG.

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

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.

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

RFC Editor: Replace RFC XXXX with actual RFC number and remove this note.

Name: iana-bfd-types

Namespace: urn:ietf:params:xml:ns:yang:iana-bfd-types

Prefix: iana-bfd-types

Reference: RFC XXXX

Name: ietf-bfd-types

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

Prefix: bfd-types

Reference: RFC XXXX

Name: ietf-bfd

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

Prefix: bfd

Reference: RFC XXXX

Name: ietf-bfd-ip-sh

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

Prefix: bfd-ip-sh

Reference: RFC XXXX

Name: ietf-bfd-ip-mh

Namespace: urn:ietf:params:xml:ns:yang:ietf-bfd-ip-mh

Prefix: bfd-ip-mh

Reference: RFC XXXX

Name: ietf-bfd-lag

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

Prefix: bfd-lag

Reference: RFC XXXX

Name: ietf-bfd-mpls

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

Prefix: bfd-mpls

Reference: RFC XXXX

Name: ietf-bfd-mpls-te

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

Prefix: bfd-mpls-te

Reference: RFC XXXX

5.1. 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 mirrors the "BFD Diagnostic Codes" registry and "BFD Authentication Types" registry at https://www.iana.org/assignments/bfd-parameters/bfd-parameters.xhtml. Whenever that registry changes, IANA must update the iana-bfd-types YANG module.

6. 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.

7. References

7.1. Normative References

[I-D.ietf-mpls-base-yang]
Saad, T., Raza, K., Gandhi, R., Liu, X., and V. Beeram, "A YANG Data Model for MPLS Base", draft-ietf-mpls-base-yang-06 (work in progress), February 2018.

[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-16 (work in progress), July 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>.

[RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.17487/RFC3688, January 2004, <https://www.rfc-editor.org/info/rfc3688>.

[RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, DOI 10.17487/RFC5246, August 2008, <https://www.rfc-editor.org/info/rfc5246>.

[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>.

[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>.

[RFC6536] Bierman, A. and M. Bjorklund, "Network Configuration Protocol (NETCONF) Access Control Model", RFC 6536, DOI 10.17487/RFC6536, March 2012, <https://www.rfc-editor.org/info/rfc6536>.

[RFC6991] Schoenwaelder, J., Ed., "Common YANG Data Types", RFC 6991, DOI 10.17487/RFC6991, July 2013, <https://www.rfc-editor.org/info/rfc6991>.

[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>.

[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>.

[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>.

[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>.

[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>.

[RFC8344] Bjorklund, M., "A YANG Data Model for IP Management", RFC 8344, DOI 10.17487/RFC8344, March 2018, <https://www.rfc-editor.org/info/rfc8344>.

[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>.

7.2. Informative References

[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-rtgwg-lne-model] Berger, L., Hopps, C., Lindem, A., Bogdanovic, D., and X. Liu, "YANG Model for Logical Network Elements", draft-ietf-rtgwg-lne-model-10 (work in progress), March 2018.

[I-D.ietf-rtgwg-ni-model] Berger, L., Hopps, C., Lindem, A., Bogdanovic, D., and X. Liu, "YANG Model for Network Instances", draft-ietf-rtgwg-ni-model-12 (work in progress), March 2018.

[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>.

Appendix A. Echo function configuration example

As mentioned in Section 2.1.2, the mechanism to start and stop the echo function, as defined in [RFC5880] and [RFC5881], is implementation specific. In this section we provide an example of how the echo function can be implemented via configuration.

```
   module: example-bfd-echo
     augment /rt:routing/rt:control-plane-protocols
               /rt:control-plane-protocol/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 configuration

```
module example-bfd-echo {
  namespace "tag:example.com,2018: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.

     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
     (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 2018-08-01 {
    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
        "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.";
    }

    leaf required-min-echo-rx-interval {
      type uint32;
      units microseconds;
      default 0;
      description
        "This is the Required Min Echo RX Interval as defined in BFD
        [RFC5880].";
    }
  }

  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 function.";

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

      description "BFD echo function container";

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

Appendix B. Change log

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

B.1. Changes between versions -16 and -17

- Addressed IESG comments.

B.2. Changes between versions -15 and -16

- Added list of modules for YANG module registry.

B.3. Changes between versions -14 and -15

- Added missing ietf-bfd-types in XML registry.

B.4. Changes between versions -13 and -14

- Addressed missing/incorrect references in import statements.

B.5. Changes between versions -12 and -13

- Updated references for drafts which became RFCs recently.

B.6. Changes between versions -11 and -12

- Addressed comments from YANG Doctor review of rev11.

B.7. Changes between versions -10 and -11

- Added 2 examples.
- Added a container around some lists.
- Fixed some indentation nits.

B.8. Changes between versions -09 and -10

- Addressed comments from YANG Doctor review.
- Addressed comments from WGLC.

B.9. Changes between versions -08 and -09

- Mostly cosmetic changes to abide by draft-ietf-netmod-rfc6087bis.
- Specified yang-version 1.1.
- Added data model examples.
- Some minor changes.

B.10. 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.

B.11. 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.

B.12. 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.

B.13. Changes between versions -04 and -05

- "bfd" node in augment of control-plane-protocol.
- Removed augment of network-instance. Replaced by schema-mount.
- Added information on interaction with other YANG modules.

B.14. 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.

B.15. Changes between versions -02 and -03

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

B.16. Changes between versions -01 and -02

- Replace routing-instance with network-instance from YANG Network Instances [I-D.ietf-rtgwg-ni-model]

B.17. 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)
Xoriant Corporation
1248 Reamwood Ave
Sunnyvale, California 94089
USA

Email: mjethanandani@gmail.com

Santosh Pallagatti
Rtbrick
India

Email: santosh.pallagatti@gmail.com

Greg Mirsky
ZTE Corporation

Email: gregimirsky@gmail.com