Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: July 12, 2017

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

M. Jethanandani
Cisco Systems
G. Mirsky
Ericsson

January 8, 2017

Yang Data Model for Bidirectional Forwarding Detection (BFD)
draft-ietf-bfd-yang-04.txt

Abstract

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

Requirements Language

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

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

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

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

This Internet-Draft will expire on July 12, 2017.

Table of Contents

1. Introduction

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

This document 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 [RFC5881]

2) Two systems connected via multiple hops [RFC5883]

3) Two systems connected via MPLS Label Switched Paths (LSPs) [RFC5884]

4) Two systems connected via pseudowires (PWs), this is known as Virtual Circuit Connectivity Verification (VCCV) [RFC5885]

5) Two systems connected via a Link Aggregation Group (LAG) interface [RFC7130]

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 [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.

1.1. Contributors

2. Design of the Data Model

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

The new top-level container "bfd" is augmented by all the YANG modules for their respective specific information.

The "network-instance" data node in [I-D.ietf-rtgwg-ni-model] has been augmented with the "bfd" data node. Where appropriate for specific types of forwarding path, the respective BFD YANG modules follow a VRF-centric model by augmenting that new "bfd" node under "network-instance".

Each node in every "bfd" sub-tree has a "config" node and an "oper" node.

2.1. Design of Configuration Model

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

Some implementations have BFD session configuration under the BFD clients. For example, BFD session configuration is 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. To avoid splitting the BFD configuration between BFD clients and BFD, all the configuration is under BFD. The only BFD configuration under BFD clients should be an "enable" knob which makes those clients react to BFD liveliness detection events.

2.1.1. Common BFD configuration parameters

The basic BFD configuration parameters are:

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

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

required-min-rx-interval
: This is the Required Min RX Interval as defined in [RFC5880].

Although [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. There are also intervals for the echo function (if supported):

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

required-min-echo-rx-interval
: This is the Required Min Echo RX Interval as defined in [RFC5880].

For BFD authentication we have:

key-chain
: This is a reference to key-chain defined in [I-D.ietf-rtgwg-yang-key-chain]. The keys, cryptographic algotihms, key lifetime etc are all defined in the key-chain model.

replay-protection
: This specifies meticulous v/s non-meticulous mode as per [RFC5880].

2.1.2. Single-hop IP

For single-hop IP, there is an augment of the "bfd" data node under "network-instance" in Section 2. We have 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

2.1.3. Multi-hop IP

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

We have a list for BFD sessions over multi-hop IP. The key consists of:

source address
: Address belonging to the local system as per [RFC5883]

destination address
: Address belonging to the remote system as per [RFC5883]

Since we are following a VRF-centric model we do not need a VRF field in the key.

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

Here are some extra parameters:

tx-ttl
: TTL of outgoing BFD control packets.

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

2.1.4. MPLS Traffic Engineering Tunnels

For MPLS-TE tunnels, BFD is configured under the MPLS-TE tunnel since the desired failure detection parameters is a property of the MPLS-TE tunnel. This is achieved by augmenting the MPLS-TE data model in [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 [RFC5586]

For general MPLS-TE data, "mpls-te" data node is added under the top-level "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 top-level "bfd" node in Section 2 is augmented with "mpls" which contains a list of sessions uniquely identified by an IP address.

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 [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 top-level "bfd" node in Section 2, the configuration is per-LAG: we have a list of LAGs. The destination IP address of the micro-BFD sessions is configured per-LAG and per address-family (IPv4 and IPv6)

2.2. Design of Operational Model

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

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

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

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

For some session types, there may be more than 1 session on the virtual path to the destination. For example, with IP multi-hop there could be multiple BFD sessions from the source to the same destination to test the various paths (ECMP) to the destination. Each of the BFD sessions on the same virtual path is uniquely identified by the local discriminator.

2.3. Notifications

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

2.4. RPC Operations

TBD

2.5. BFD top level hierarchy

At the top level there is no configuration data, only operational data. The operational data consist of overall BFD session statistics globally and per network-instance

```
   module: ietf-bfd
      +--rw bfd!
         +--rw config
         +--ro oper
            +--ro bfd-session-statistics
               +--ro session-count?              uint32
               +--ro session-up-count?           uint32
               +--ro session-down-count?         uint32
               +--ro session-admin-down-count?   uint32
   augment /ni:network-instances/ni:network-instance:
      +--rw bfd!
         +--rw config
         +--ro oper
            +--ro bfd-session-statistics
               +--ro session-count?              uint32
               +--ro session-up-count?           uint32
               +--ro session-down-count?         uint32
               +--ro session-admin-down-count?   uint32
```

2.6. BFD IP single-hop hierarchy

The configuration and operational data for each BFD IP single-hop session is under "ip-sh" node under network-instance. Overall BFD IP single-hop session statistics are available per network-instance and globally (for all network instances).

```
  module: ietf-bfd-ip-sh
  augment /bfd:bfd:
     +--rw ip-sh
        +--rw config
        +--ro oper
           +--ro bfd-session-statistics
              +--ro session-count?              uint32
              +--ro session-up-count?           uint32
              +--ro session-down-count?         uint32
              +--ro session-admin-down-count?   uint32
  augment /ni:network-instances/ni:network-instance/bfd:bfd:
     +--rw ip-sh
        +--rw config
        |  +--rw session-cfg
        |     +--rw sessions* [interface dest-addr]
        |        +--rw interface                        if:interface-ref
        |        +--rw dest-addr                        inet:ip-address
        |        +--rw source-addr?                     inet:ip-address
        |        +--rw local-multiplier?                bfd-multiplier
        |        +--rw (interval-config-type)?
        |        |  +--:(tx-rx-intervals)
        |        |  |  +--rw desired-min-tx-interval          uint32
```

```
|       |  |  +--rw required-min-rx-interval        uint32
|       |  +--:(single-interval)
|       |     +--rw min-interval                    uint32
|       +--rw demand-enabled?                  boolean
|       +--rw admin-down?                      boolean
|       +--rw authentication-parms! {bfd-authentication}?
|       |  +--rw key-chain?           kc:key-chain-ref
|       |  +--rw replay-protection?   identityref
|       +--rw desired-min-echo-tx-interval?    uint32
|       +--rw required-min-echo-rx-interval?   uint32
+--ro oper
   +--ro bfd-session-statistics
   |  +--ro session-count?              uint32
   |  +--ro session-up-count?           uint32
   |  +--ro session-down-count?         uint32
   |  +--ro session-admin-down-count?   uint32
   +--ro sessions* [interface dest-addr]
      +--ro interface               if:interface-ref
      +--ro dest-addr               inet:ip-address
      +--ro source-addr?            inet:ip-address
      +--ro path-type?              identityref
      +--ro local-discriminator?    bfd-discriminator
      +--ro remote-discriminator?   bfd-discriminator
      +--ro remote-multiplier?      bfd-multiplier
      +--ro out-interface?          if:interface-ref
      +--ro demand-capability?      boolean
      +--ro source-port?            inet:port-number
      +--ro dest-port?              inet:port-number
      +--ro session-running
      |  +--ro session-index?             uint32
      |  +--ro local-state?               bfd-state
      |  +--ro remote-state?              bfd-state
      |  +--ro local-diagnostic?          bfd-diagnostic
      |  +--ro remote-diagnostic?         bfd-diagnostic
      |  +--ro detection-mode?            enumeration
      |  +--ro negotiated-tx-interval?    uint32
      |  +--ro negotiated-rx-interval?    uint32
      |  +--ro echo-tx-interval-in-use?   uint32
      |  +--ro detection-time?            uint32
      +--ro sesssion-statistics
         +--ro create-time?           yang:date-and-time
         +--ro last-down-time?        yang:date-and-time
         +--ro last-up-time?          yang:date-and-time
         +--ro down-count?            uint32
         +--ro admin-down-count?      uint32
         +--ro receive-packet-count?  uint64
         +--ro send-packet-count?     uint64
         +--ro receive-bad-packet?    uint64
```

```
               +--ro send-failed-packet?     uint64
  notifications:
     +---n bfd-singlehop-notification
        +--ro local-discr?              bfd-discriminator
        +--ro remote-discr?             bfd-discriminator
        +--ro new-state?                bfd-state
        +--ro state-change-reason?      string
        +--ro time-in-previous-state?   string
        +--ro dest-addr?                inet:ip-address
        +--ro source-addr?              inet:ip-address
        +--ro session-index?            uint32
        +--ro path-type?                identityref
        +--ro interface?                if:interface-ref
        +--ro echo-enabled?             boolean
```

2.7. BFD IP multi-hop hierarchy

The configuration and operational data for each BFD IP multi-hop session is under "ip-mh" node under network-instance. In the operational model we support multiple BFD multi-hop sessions per remote address (ECMP), the local discriminator is used as key. Overall BFD IP multi-hop session statistics are available per network-instance and globally (for all network instances).

```
module: ietf-bfd-ip-mh
augment /bfd:bfd:
   +--rw ip-mh
      +--rw config
      +--ro oper
         +--ro bfd-session-statistics
            +--ro session-count?              uint32
            +--ro session-up-count?           uint32
            +--ro session-down-count?         uint32
            +--ro session-admin-down-count?   uint32
augment /ni:network-instances/ni:network-instance/bfd:bfd:
   +--rw ip-mh
      +--rw config
      |  +--rw session-cfg
      |     +--rw sessions* [source-addr dest-addr]
      |        +--rw source-addr                inet:ip-address
      |        +--rw dest-addr                  inet:ip-address
      |        +--rw local-multiplier?          bfd-multiplier
      |        +--rw (interval-config-type)?
      |        |  +--:(tx-rx-intervals)
      |        |  |  +--rw desired-min-tx-interval     uint32
      |        |  |  +--rw required-min-rx-interval    uint32
      |        |  +--:(single-interval)
      |        |     +--rw min-interval                uint32
```

```
|       +--rw demand-enabled?            boolean
|       +--rw admin-down?                boolean
|       +--rw authentication-parms! {bfd-authentication}?
|       |  +--rw key-chain?           kc:key-chain-ref
|       |  +--rw replay-protection?   identityref
|       +--rw tx-ttl?                    bfd:ttl
|       +--rw rx-ttl                     bfd:ttl
+--ro oper
   +--ro bfd-session-statistics
   |  +--ro bfd-session-statistics
   |     +--ro session-count?              uint32
   |     +--ro session-up-count?           uint32
   |     +--ro session-down-count?         uint32
   |     +--ro session-admin-down-count?   uint32
   +--ro session-group* [source-addr dest-addr]
      +--ro source-addr    inet:ip-address
      +--ro dest-addr      inet:ip-address
      +--ro sessions* [local-discriminator]
         +--ro ttl?                    bfd:ttl
         +--ro path-type?              identityref
         +--ro local-discriminator     bfd-discriminator
         +--ro remote-discriminator?   bfd-discriminator
         +--ro remote-multiplier?      bfd-multiplier
         +--ro out-interface?          if:interface-ref
         +--ro demand-capability?      boolean
         +--ro source-port?            inet:port-number
         +--ro dest-port?              inet:port-number
         +--ro session-running
         |  +--ro session-index?             uint32
         |  +--ro local-state?               bfd-state
         |  +--ro remote-state?              bfd-state
         |  +--ro local-diagnostic?          bfd-diagnostic
         |  +--ro remote-diagnostic?         bfd-diagnostic
         |  +--ro detection-mode?            enumeration
         |  +--ro negotiated-tx-interval?    uint32
         |  +--ro negotiated-rx-interval?    uint32
         |  +--ro echo-tx-interval-in-use?   uint32
         |  +--ro detection-time?            uint32
         +--ro sesssion-statistics
            +--ro create-time?           yang:date-and-time
            +--ro last-down-time?        yang:date-and-time
            +--ro last-up-time?          yang:date-and-time
            +--ro down-count?            uint32
            +--ro admin-down-count?      uint32
            +--ro receive-packet-count?  uint64
            +--ro send-packet-count?     uint64
            +--ro receive-bad-packet?    uint64
            +--ro send-failed-packet?    uint64
```

```
   notifications:
      +---n bfd-multihop-notification
         +--ro local-discr?              bfd-discriminator
         +--ro remote-discr?             bfd-discriminator
         +--ro new-state?                bfd-state
         +--ro state-change-reason?      string
         +--ro time-in-previous-state?   string
         +--ro dest-addr?                inet:ip-address
         +--ro source-addr?              inet:ip-address
         +--ro session-index?            uint32
         +--ro path-type?                identityref
         +--ro network-instance?         string
```

2.8. BFD over LAG hierarchy

The configuration and operational data for each BFD LAG session is under "lag" node. Overall BFD LAG session statistics are available globally.

```
module: ietf-bfd-lag
augment /bfd:bfd:
   +--rw lag
      +--rw config
      |  +--rw session-cfg
      |     +--rw sessions* [lag-name]
      |        +--rw lag-name                    if:interface-ref
      |        +--rw ipv4-dest-addr?             inet:ipv4-address
      |        +--rw ipv6-dest-addr?             inet:ipv6-address
      |        +--rw local-multiplier?           bfd-multiplier
      |        +--rw (interval-config-type)?
      |        |  +--:(tx-rx-intervals)
      |        |  |  +--rw desired-min-tx-interval     uint32
      |        |  |  +--rw required-min-rx-interval    uint32
      |        |  +--:(single-interval)
      |        |     +--rw min-interval                uint32
      |        +--rw demand-enabled?             boolean
      |        +--rw admin-down?                 boolean
      |        +--rw authentication-parms! {bfd-authentication}?
      |           +--rw key-chain?           kc:key-chain-ref
      |           +--rw replay-protection?   identityref
      +--ro oper
         +--ro micro-bfd-ipv4-session-statistics
         |  +--ro bfd-session-statistics
         |     +--ro session-count?              uint32
         |     +--ro session-up-count?           uint32
         |     +--ro session-down-count?         uint32
         |     +--ro session-admin-down-count?   uint32
         +--ro micro-bfd-ipv6-session-statistics
```

```
|  +--ro bfd-session-statistics
|     +--ro session-count?              uint32
|     +--ro session-up-count?           uint32
|     +--ro session-down-count?         uint32
|     +--ro session-admin-down-count?   uint32
+--ro session-lag* [lag-name]
   +--ro lag-name        if:interface-ref
   +--ro use-ipv4?       boolean
   +--ro use-ipv6?       boolean
   +--ro member-links* [member-link]
      +--ro member-link       if:interface-ref
      +--ro micro-bfd-ipv4
      |  +--ro path-type?              identityref
      |  +--ro local-discriminator?    bfd-discriminator
      |  +--ro remote-discriminator?   bfd-discriminator
      |  +--ro remote-multiplier?      bfd-multiplier
      |  +--ro out-interface?          if:interface-ref
      |  +--ro demand-capability?      boolean
      |  +--ro source-port?            inet:port-number
      |  +--ro dest-port?              inet:port-number
      |  +--ro session-running
      |  |  +--ro session-index?             uint32
      |  |  +--ro local-state?               bfd-state
      |  |  +--ro remote-state?              bfd-state
      |  |  +--ro local-diagnostic?          bfd-diagnostic
      |  |  +--ro remote-diagnostic?         bfd-diagnostic
      |  |  +--ro detection-mode?            enumeration
      |  |  +--ro negotiated-tx-interval?    uint32
      |  |  +--ro negotiated-rx-interval?    uint32
      |  |  +--ro echo-tx-interval-in-use?   uint32
      |  |  +--ro detection-time?            uint32
      |  +--ro sesssion-statistics
      |     +--ro create-time?            yang:date-and-time
      |     +--ro last-down-time?         yang:date-and-time
      |     +--ro last-up-time?           yang:date-and-time
      |     +--ro down-count?             uint32
      |     +--ro admin-down-count?       uint32
      |     +--ro receive-packet-count?   uint64
      |     +--ro send-packet-count?      uint64
      |     +--ro receive-bad-packet?     uint64
      |     +--ro send-failed-packet?     uint64
      +--ro micro-bfd-ipv6
         +--ro path-type?              identityref
         +--ro local-discriminator?    bfd-discriminator
         +--ro remote-discriminator?   bfd-discriminator
         +--ro remote-multiplier?      bfd-multiplier
         +--ro out-interface?          if:interface-ref
         +--ro demand-capability?      boolean
```

```
                    +--ro source-port?           inet:port-number
                    +--ro dest-port?             inet:port-number
                    +--ro session-running
                    |  +--ro session-index?            uint32
                    |  +--ro local-state?              bfd-state
                    |  +--ro remote-state?             bfd-state
                    |  +--ro local-diagnostic?         bfd-diagnostic
                    |  +--ro remote-diagnostic?        bfd-diagnostic
                    |  +--ro detection-mode?           enumeration
                    |  +--ro negotiated-tx-interval?   uint32
                    |  +--ro negotiated-rx-interval?   uint32
                    |  +--ro echo-tx-interval-in-use?  uint32
                    |  +--ro detection-time?           uint32
                    +--ro sesssion-statistics
                       +--ro create-time?            yang:date-and-time
                       +--ro last-down-time?         yang:date-and-time
                       +--ro last-up-time?           yang:date-and-time
                       +--ro down-count?             uint32
                       +--ro admin-down-count?       uint32
                       +--ro receive-packet-count?   uint64
                       +--ro send-packet-count?      uint64
                       +--ro receive-bad-packet?     uint64
                       +--ro send-failed-packet?     uint64
   notifications:
      +---n bfd-lag-notification
         +--ro local-discr?              bfd-discriminator
         +--ro remote-discr?             bfd-discriminator
         +--ro new-state?                bfd-state
         +--ro state-change-reason?      string
         +--ro time-in-previous-state?   string
         +--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

The configuration is per MPLS FEC under "mpls". In the operational model we support multiple BFD sessions per MPLS FEC (ECMP), the local discriminator is used as key.

```
                     module: ietf-bfd-mpls
   module: ietf-bfd-mpls
   augment /bfd:bfd:
      +--rw mpls
         +--rw config
```

```
         |  +--rw egress
         |     +--rw local-multiplier?           bfd-multiplier
         |     +--rw (interval-config-type)?
         |     |  +--:(tx-rx-intervals)
         |     |  |  +--rw desired-min-tx-interval     uint32
         |     |  |  +--rw required-min-rx-interval    uint32
         |     |  +--:(single-interval)
         |     |     +--rw min-interval                uint32
         |     +--rw authentication-parms! {bfd-authentication}?
         |        +--rw key-chain?           kc:key-chain-ref
         |        +--rw replay-protection?   identityref
         +--ro oper
            +--ro bfd-session-statistics
               +--ro session-count?              uint32
               +--ro session-up-count?           uint32
               +--ro session-down-count?         uint32
               +--ro session-admin-down-count?   uint32
   augment /ni:network-instances/ni:network-instance/bfd:bfd:
      +--rw mpls
         +--rw config
         |  +--rw session-cfg
         |     +--rw sessions* [mpls-fec]
         |        +--rw mpls-fec                    inet:ip-address
         |        +--rw local-multiplier?           bfd-multiplier
         |        +--rw (interval-config-type)?
         |        |  +--:(tx-rx-intervals)
         |        |  |  +--rw desired-min-tx-interval     uint32
         |        |  |  +--rw required-min-rx-interval    uint32
         |        |  +--:(single-interval)
         |        |     +--rw min-interval                uint32
         |        +--rw demand-enabled?             boolean
         |        +--rw admin-down?                 boolean
         |        +--rw authentication-parms! {bfd-authentication}?
         |           +--rw key-chain?           kc:key-chain-ref
         |           +--rw replay-protection?   identityref
         +--ro oper
            +--ro bfd-session-statistics
            |  +--ro session-count?              uint32
            |  +--ro session-up-count?           uint32
            |  +--ro session-down-count?         uint32
            |  +--ro session-admin-down-count?   uint32
            +--ro session-group* [mpls-fec]
               +--ro mpls-fec    inet:ip-address
               +--ro sessions* [local-discriminator]
                  +--ro path-type?              identityref
                  +--ro local-discriminator     bfd-discriminator
                  +--ro remote-discriminator?   bfd-discriminator
                  +--ro remote-multiplier?      bfd-multiplier
```

```
                +--ro out-interface?          if:interface-ref
                +--ro demand-capability?      boolean
                +--ro source-port?            inet:port-number
                +--ro dest-port?              inet:port-number
                +--ro session-running
                |  +--ro session-index?             uint32
                |  +--ro local-state?               bfd-state
                |  +--ro remote-state?              bfd-state
                |  +--ro local-diagnostic?          bfd-diagnostic
                |  +--ro remote-diagnostic?         bfd-diagnostic
                |  +--ro detection-mode?            enumeration
                |  +--ro negotiated-tx-interval?    uint32
                |  +--ro negotiated-rx-interval?    uint32
                |  +--ro echo-tx-interval-in-use?   uint32
                |  +--ro detection-time?            uint32
                +--ro sesssion-statistics
                |  +--ro create-time?            yang:date-and-time
                |  +--ro last-down-time?         yang:date-and-time
                |  +--ro last-up-time?           yang:date-and-time
                |  +--ro down-count?             uint32
                |  +--ro admin-down-count?       uint32
                |  +--ro receive-packet-count?   uint64
                |  +--ro send-packet-count?      uint64
                |  +--ro receive-bad-packet?     uint64
                |  +--ro send-failed-packet?     uint64
                +--ro mpls-dest-address?      inet:ip-address
   notifications:
      +---n bfd-mpls-notification
         +--ro local-discr?              bfd-discriminator
         +--ro remote-discr?             bfd-discriminator
         +--ro new-state?                bfd-state
         +--ro state-change-reason?      string
         +--ro time-in-previous-state?   string
         +--ro dest-addr?                inet:ip-address
         +--ro source-addr?              inet:ip-address
         +--ro session-index?            uint32
         +--ro path-type?                identityref
         +--ro network-instance?         string
         +--ro mpls-dest-address?        inet:ip-address
```

2.10. BFD over MPLS-TE hierarchy

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

```
module: ietf-bfd-mpls-te
augment /bfd:bfd:
```

```
      +--rw mpls-te
         +--rw config
         |  +--rw egress
         |     +--rw local-multiplier?           bfd-multiplier
         |     +--rw (interval-config-type)?
         |     |  +--:(tx-rx-intervals)
         |     |  |  +--rw desired-min-tx-interval     uint32
         |     |  |  +--rw required-min-rx-interval    uint32
         |     |  +--:(single-interval)
         |     |     +--rw min-interval                uint32
         |     +--rw authentication-parms! {bfd-authentication}?
         |        +--rw key-chain?           kc:key-chain-ref
         |        +--rw replay-protection?   identityref
         +--ro oper
            +--ro bfd-session-statistics
               +--ro session-count?              uint32
               +--ro session-up-count?           uint32
               +--ro session-down-count?         uint32
               +--ro session-admin-down-count?   uint32
   augment /te:te/te:tunnels/te:tunnel/te:config:
      +--rw local-multiplier?           bfd-multiplier
      +--rw (interval-config-type)?
      |  +--:(tx-rx-intervals)
      |  |  +--rw desired-min-tx-interval     uint32
      |  |  +--rw required-min-rx-interval    uint32
      |  +--:(single-interval)
      |     +--rw min-interval                uint32
      +--rw demand-enabled?             boolean
      +--rw admin-down?                 boolean
      +--rw authentication-parms! {bfd-authentication}?
      |  +--rw key-chain?           kc:key-chain-ref
      |  +--rw replay-protection?   identityref
      +--rw encap?                      identityref
   augment /te:te/te:lsps-state/te:lsp:
      +--ro path-type?              identityref
      +--ro local-discriminator?    bfd-discriminator
      +--ro remote-discriminator?   bfd-discriminator
      +--ro remote-multiplier?      bfd-multiplier
      +--ro out-interface?          if:interface-ref
      +--ro demand-capability?      boolean
      +--ro source-port?            inet:port-number
      +--ro dest-port?              inet:port-number
      +--ro session-running
      |  +--ro session-index?             uint32
      |  +--ro local-state?               bfd-state
      |  +--ro remote-state?              bfd-state
      |  +--ro local-diagnostic?          bfd-diagnostic
      |  +--ro remote-diagnostic?         bfd-diagnostic
```

```
      |  +--ro detection-mode?            enumeration
      |  +--ro negotiated-tx-interval?    uint32
      |  +--ro negotiated-rx-interval?    uint32
      |  +--ro echo-tx-interval-in-use?   uint32
      |  +--ro detection-time?            uint32
      +--ro sesssion-statistics
      |  +--ro create-time?            yang:date-and-time
      |  +--ro last-down-time?         yang:date-and-time
      |  +--ro last-up-time?           yang:date-and-time
      |  +--ro down-count?             uint32
      |  +--ro admin-down-count?       uint32
      |  +--ro receive-packet-count?   uint64
      |  +--ro send-packet-count?      uint64
      |  +--ro receive-bad-packet?     uint64
      |  +--ro send-failed-packet?     uint64
      +--ro mpls-dest-address?      inet:ip-address
   notifications:
      +---n bfd-mpls-te-notification
         +--ro local-discr?              bfd-discriminator
         +--ro remote-discr?             bfd-discriminator
         +--ro new-state?                bfd-state
         +--ro state-change-reason?      string
         +--ro time-in-previous-state?   string
         +--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. Examples

2.12. Interaction with other YANG modules

TBD.

2.13. BFD top-level Yang Module

```
<CODE BEGINS> file "ietf-bfd@2016-07-04.yang"
module ietf-bfd {
  namespace "urn:ietf:params:xml:ns:yang:ietf-bfd";
  // replace with IANA namespace when assigned
  prefix "bfd";

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

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

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

import ietf-routing {
  prefix "rt";
}

import ietf-network-instance {
  prefix "ni";
}

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

organization "IETF BFD Working Group";

contact
  "WG Web:   <http://tools.ietf.org/wg/bfd>
   WG List:  <rtg-bfd@ietf.org>
   WG Chair: Jeff Haas
   WG Chair: Reshad Rahman
   Editor:   Lianshu Zheng and Reshad Rahman";

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

revision 2016-07-04 {
  description "Initial revision.";
  reference "RFC XXXX: A YANG data model for BFD";
}

identity bfd {
  base "rt:routing-protocol";
  description "BFD protocol";
}

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

```
    }

    typedef bfd-diagnostic {
      type enumeration {
        enum none {
          value 0;
          description "None";
        }
        enum controlExpiry {
          value 1;
          description "Control timer expiry";
        }
        enum echoFailed {
          value 2;
          description "Echo failure";
        }
        enum nborDown {
          value 3;
          description "Neighbor down";
        }
        enum fwdingReset {
          value 4;
          description "Forwarding reset";
        }
        enum pathDown {
          value 5;
          description "Path down";
        }
        enum concPathDown {
          value 6;
          description "Concatenated path down";
        }
        enum adminDown {
          value 7;
          description "Admin down";
        }
        enum reverseConcPathDown {
          value 8;
          description "Reverse concatenated path down";
        }
      }
      description "BFD diagnostic";
    }

    typedef bfd-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";
    }

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

    typedef ttl {
      type uint8 {
        range 1..255;
      }
      description "Time To Live";
    }

    identity bfd-path-type {
      description
        "Base identity for BFD path type. The session type indicates
         the type of path on which BFD is running";
    }
    identity bfd-path-ip-sh {
      base bfd-path-type;
      description "BFD on IP single hop";
    }
    identity bfd-path-ip-mh {
      base bfd-path-type;
      description "BFD on IP multi hop";
    }
    identity bfd-path-mpls-te {
      base bfd-path-type;
      description "BFD on MPLS Traffic Engineering";
```

```
    }
    identity bfd-path-mpls-lsp {
      base bfd-path-type;
      description "BFD on MPLS Label Switched Path";
    }
    identity bfd-path-lag {
      base bfd-path-type;
      description "Micro-BFD on LAG member links";
    }

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

    feature bfd-authentication {
      description "BFD authentication supported";
    }

    identity bfd-auth-replay-protection {
      description
        "Base identity for BFD authentication replay protection";
    }
    identity bfd-auth-replay-protection-non-meticulous {
      base bfd-auth-replay-protection;
      description "Non-meticulous (see RFC5880)";
    }
    identity bfd-auth-replay-protection-meticulous {
      base bfd-auth-replay-protection;
      description "Meticulous (see RFC5880)";
    }

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

        leaf key-chain {
          type kc:key-chain-ref;
```

```
          description "Name of key-chain";
        }

        leaf replay-protection {
          type identityref {
            base bfd-auth-replay-protection;
          }
          description
            "Protection against replays";
        }
      }
    }


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

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

          leaf required-min-rx-interval {
            type uint32;
            units microseconds;
            mandatory true;
            description
              "Required minimum receive interval of control packets";
          }
        }
        case single-interval {
          leaf min-interval {
            type uint32;
            units microseconds;
            mandatory true;
            description
```

```
              "Desired minimum transmit interval and required " +
              "minimum receive interval of control packets";
          }
        }
      }
    }

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

      uses bfd-grouping-base-cfg-parms;

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

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

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

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

    grouping bfd-client-base-cfg-parms {
      description
        "BFD grouping which could be used by a protocol which
         is a client of BFD to enable its use of BFD";
```

```
       container bfd-cfg {
         description "BFD configuration";
         leaf enabled {
           type boolean;
           default false;
           description "True if BFD is enabled";
         }
       }
     }

     grouping bfd-all-session {
       description "BFD session operational information";
       leaf path-type {
         type identityref {
           base bfd-path-type;
         }
         description
           "BFD session type, this indicates the path type that BFD is
           running on";
       }
       leaf local-discriminator {
         type bfd-discriminator;
         description "Local discriminator";
       }
       leaf remote-discriminator {
         type bfd-discriminator;
         description "Remote discriminator";
       }
       leaf remote-multiplier {
         type bfd-multiplier;
         description "Remote multiplier";
       }
       leaf out-interface {
         type if:interface-ref;
         description "Outgoing physical interface name";
       }
       leaf demand-capability {
         type boolean;
         description "Local demand mode capability";
       }
       leaf source-port {
         type inet:port-number;
         description "Source UDP port";
       }
       leaf dest-port {
         type inet:port-number;
         description "Destination UDP port";
       }
```

```
       container session-running {
         description "BFD session running information";
         leaf session-index {
           type uint32;
           description
             "An index used to uniquely identify BFD sessions";
         }
         leaf local-state {
           type bfd-state;
           description "Local state";
         }
         leaf remote-state {
           type bfd-state;
           description "Remote state";
         }
         leaf local-diagnostic {
           type bfd-diagnostic;
           description "Local diagnostic";
         }
         leaf remote-diagnostic {
           type bfd-diagnostic;
           description "Remote diagnostic";
         }
         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 echo-tx-interval-in-use {
          when "../../path-type = 'bfd-path-ip-sh'" {
            description
              "Echo is supported for IP single-hop only.";
          }
          type uint32;
          units microseconds;
          description "Echo transmit interval in use";
        }
        leaf detection-time {
          type uint32;
          units microseconds;
          description "Detection time";
        }
      }

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

        leaf create-time {
          type yang:date-and-time;
          description
            "Time and date when session was created";
        }
        leaf last-down-time {
          type yang:date-and-time;
          description
            "Time and date of last time the session went down";
        }
        leaf last-up-time {
          type yang:date-and-time;
          description
            "Time and date of last time the session went up";
        }
        leaf down-count {
          type uint32;
          description "Session Down Count";
        }
        leaf admin-down-count {
          type uint32;
          description "Session Admin-Down Count";
        }
```

```
        leaf receive-packet-count {
          type uint64;
          description "Received Packet Count";
        }
        leaf send-packet-count {
          type uint64;
          description "Sent Packet Count";
        }
        leaf receive-bad-packet {
          type uint64;
          description "Received bad packet count";
        }
        leaf send-failed-packet {
          type uint64;
          description "Packet Failed to Send Count";
        }
      }
    }

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

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

      leaf local-discr {
```

```
        type bfd-discriminator;
        description "BFD local discriminator";
      }

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

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

      leaf state-change-reason {
        type string;
        description "BFD state change reason";
      }

      leaf time-in-previous-state {
        type string;
        description
          "How long the BFD session was in the previous state";
      }

      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 bfd-path-type;
        }
        description "BFD path type";
      }
    }
```

```
    augment "/ni:network-instances/ni:network-instance" {
      description "BFD augmentation.";

      container bfd {
        presence "BFD";
        description "BFD top level container";
        container config {
          description "BFD configuration container";
        }

        container oper {
          config "false";
          description
            "BFD operational container for this routing instance";
          uses bfd-session-statistics;
        }
      }
    }

    container bfd {
      presence "BFD";
      description "BFD top level container";
      container config {
        description "BFD configuration container";
      }

      container oper {
        config "false";
        description "BFD operational container.";
        uses bfd-session-statistics;
      }
    }
  }
  <CODE ENDS>
```

2.14. BFD IP single-hop Yang Module

```
  <CODE BEGINS> file "ietf-bfd-ip-sh@2016-07-04.yang"
  module ietf-bfd-ip-sh {
    namespace "urn:ietf:params:xml:ns:yang:ietf-bfd-ip-sh";
    // replace with IANA namespace when assigned
    prefix "bfd-ip-sh";

    import ietf-bfd {
      prefix "bfd";
    }

    import ietf-interfaces {
```

```
      prefix "if";
    }

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

    import ietf-network-instance {
      prefix "ni";
    }

    organization "IETF BFD Working Group";

    contact
      "WG Web:   <http://tools.ietf.org/wg/bfd>
       WG List:  <rtg-bfd@ietf.org>
       WG Chair: Jeff Haas
       WG Chair: Reshad Rahman
       Editor:   Lianshu Zheng and Reshad Rahman";

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

    revision 2016-07-04 {
      description "Initial revision.";
      reference "RFC XXXX: A YANG data model for BFD IP single-hop";
    }

    augment "/bfd:bfd" {
      description "BFD augmentation for IP single-hop";
      container ip-sh {
        description "BFD IP single-hop top level container";

        container config {
          description "BFD IP single-hop configuration container";
        }
        container oper {
          config "false";
          description "BFD IP single-hop operational container";
          uses bfd:bfd-session-statistics;
        }
      }
    }

    augment "/ni:network-instances/ni:network-instance/bfd:bfd" {
      description "BFD augmentation for IP single-hop.";
```

```
       container ip-sh {
         description "BFD IP single-hop container";
         container config {
           description "BFD configuration";
           container session-cfg {
             description "BFD IP single-hop session configuration";

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

               uses bfd:bfd-grouping-common-cfg-parms;

               uses bfd:bfd-grouping-echo-cfg-parms;
             }
           }
         }

         container oper {
           config "false";
           description "BFD operational container";

           uses bfd:bfd-session-statistics;

           list sessions {
             key "interface dest-addr";
             description "BFD 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 "BFD peer address";
```

```
            }
            leaf source-addr {
              type inet:ip-address;
              description "BFD source address";
            }

            uses bfd:bfd-all-session;
          }
        }
      }
    }

    notification bfd-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:bfd-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>
```

2.15. BFD IP multi-hop Yang Module

```
  <CODE BEGINS> file "ietf-bfd-ip-mh@2016-07-04.yang"
  module ietf-bfd-ip-mh {
    namespace "urn:ietf:params:xml:ns:yang:ietf-bfd-ip-mh";
    // replace with IANA namespace when assigned
    prefix "bfd-ip-mh";

    import ietf-bfd {
      prefix "bfd";
    }

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

```
    }

    import ietf-network-instance {
      prefix "ni";
    }

    organization "IETF BFD Working Group";

    contact
      "WG Web:   <http://tools.ietf.org/wg/bfd>
       WG List:  <rtg-bfd@ietf.org>
       WG Chair: Jeff Haas
       WG Chair: Reshad Rahman
       Editor:   Lianshu Zheng and Reshad Rahman";

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

    revision 2016-07-04 {
      description "Initial revision.";
      reference "RFC XXXX: A YANG data model for BFD IP multi-hop";
    }

    augment "/bfd:bfd" {
      description "BFD augmentation for IP multi-hop";
      container ip-mh {
        description "BFD IP multi-hop top level container";

        container config {
          description "BFD IP multi-hop configuration container";
        }
        container oper {
          config "false";
          description "BFD IP multi-hop operational container";
          uses bfd:bfd-session-statistics;
        }
      }
    }

    augment "/ni:network-instances/ni:network-instance/bfd:bfd" {
      description "BFD augmentation for IP multi-hop.";

      container ip-mh {
        description "BFD IP multi-hop container";
        container config {
          description "BFD configuration";
          container session-cfg {
```

```
            description "BFD IP multi-hop session configuration";

            list sessions {
              key "source-addr dest-addr";
              description "List of IP multi-hop sessions";

              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:bfd-grouping-common-cfg-parms;

              leaf tx-ttl {
                type bfd:ttl;
                default 255;
                description "TTL of outgoing BFD control packets";
              }
              leaf rx-ttl {
                type bfd:ttl;
                mandatory true;
                description
                  "Minimum allowed TTL value for incoming BFD control
                   packets";
              }
            }
          }
        }

        container oper {
          config "false";
          description "BFD operational container";

          container bfd-session-statistics {
            description "BFD session counters";
            uses bfd:bfd-session-statistics;
          }

          list session-group {
            key "source-addr dest-addr";
            description
              "BFD IP multi-hop group of sessions. 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 "BFD source address";
            }
            leaf dest-addr {
              type inet:ip-address;
              description "BFD peer address";
            }
            list sessions {
              key "local-discriminator";
              description
                "The BFD sessions between a source and a. " +
                "destination. Local discriminator is unique for " +
                "each session in the group.";
              leaf ttl {
                type bfd:ttl;
                description "TTL of outgoing packets";
              }
              uses bfd:bfd-all-session;
            }
          }
        }
      }

      notification bfd-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:bfd-notification-parms;

        leaf network-instance {
          type string;
          description "Network instance";
        }
      }
    }
    <CODE ENDS>
```

2.16. BFD over LAG Yang Module

```
    <CODE BEGINS> file "ietf-bfd-lag@2017-01-07.yang"
    module ietf-bfd-lag {
      namespace "urn:ietf:params:xml:ns:yang:ietf-bfd-lag";
```

```
    // replace with IANA namespace when assigned
    prefix "bfd-lag";

    import ietf-bfd {
      prefix "bfd";
    }

    import ietf-interfaces {
      prefix "if";
    }

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

    organization "IETF BFD Working Group";

    contact
      "WG Web:   <http://tools.ietf.org/wg/bfd>
       WG List:  <rtg-bfd@ietf.org>
       WG Chair: Jeff Haas
       WG Chair: Reshad Rahman
       Editor:   Lianshu Zheng and Reshad Rahman";

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

    revision 2017-01-07 {
      description
        "* Update author information
         * Fixed error with when use-ipv4/use-ipv6 statements
        ";
      reference "RFC XXXX: A YANG data model for BFD over LAG";
    }

    revision 2016-02-17 {
      description "Initial revision.";
      reference "RFC XXXX: A YANG data model for BFD over LAG";
    }

    augment "/bfd:bfd" {
      description "BFD augmentation for LAG";
      container lag {
        description "BFD over LAG top level container";

        container config {
          description "BFD over LAG configuration container";
```

```
            container session-cfg {
              description "BFD over LAG session configuration";
              list sessions {
                key "lag-name";
                description "A LAG interface on which BFD is running";
                leaf lag-name {
                  type if:interface-ref ;
                  description "Name of the LAG";
                }
                leaf ipv4-dest-addr {
                  type inet:ipv4-address;
                  description
                    "IPv4 address of the peer, for IPv4 micro-BFD.";
                }
                leaf ipv6-dest-addr {
                  type inet:ipv6-address;
                  description
                    "IPv6 address of the peer, for IPv6 micro-BFD.";
                }
                uses bfd:bfd-grouping-common-cfg-parms;
              }
            }
          }

          container oper {
            config "false";
            description "BFD over LAG operational container.";

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

            list session-lag {
              key "lag-name";
              description "A LAG interface on which BFD is running";
              leaf lag-name {
                type if:interface-ref ;
                description "Name of the LAG";
              }

              leaf 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";
              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:bfd-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:bfd-all-session;
              }
            }
          }
        }
      }

      notification bfd-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:bfd-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>
```

2.17. BFD over MPLS Yang Module

```
  <CODE BEGINS> file "ietf-bfd-mpls@2016-07-04.yang"
  module ietf-bfd-mpls {
    namespace "urn:ietf:params:xml:ns:yang:ietf-bfd-mpls";
    // replace with IANA namespace when assigned
    prefix "bfd-mpls";

    import ietf-bfd {
      prefix "bfd";
    }

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

    import ietf-network-instance {
      prefix "ni";
    }

    organization "IETF BFD Working Group";

    contact
      "WG Web:   <http://tools.ietf.org/wg/bfd>
       WG List:  <rtg-bfd@ietf.org>
       WG Chair: Jeff Haas
       WG Chair: Reshad Rahman
       Editor:   Lianshu Zheng and Reshad Rahman";

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

    revision 2016-07-04 {
      description "Initial revision.";
      reference "RFC XXXX: A YANG data model for BFD over MPLS LSPs";
    }
```

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

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

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

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

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

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


    augment "/bfd:bfd" {
      description "BFD augmentation for MPLS";
      container mpls {
        description "BFD MPLS top level container";

        container config {
          description "BFD MPLS configuration container";

          container egress {
            description "Egress configuration";
```

```
            uses bfd:bfd-grouping-base-cfg-parms;

            uses bfd:bfd-auth-parms;
          }
        }

        container oper {
          config "false";
          description "BFD MPLS operational container";
          uses bfd:bfd-session-statistics;
        }
      }
    }

    augment "/ni:network-instances/ni:network-instance/bfd:bfd" {
      description "BFD augmentation for MPLS.";

      container mpls {
        description "BFD MPLS container";

        container config {
          description "BFD MPLS configuration container";

          container session-cfg {
            description "BFD MPLS session configuration";
            list sessions {
              key "mpls-fec";
              description "List of BFD MPLS sessions";
              leaf mpls-fec {
                type inet:ip-address;
                description "MPLS FEC";
              }

              uses bfd:bfd-grouping-common-cfg-parms;
            }
          }
        }

        container oper {
          config "false";
          description "BFD MPLS operational container";
          uses bfd:bfd-session-statistics;

          list session-group {
            key "mpls-fec";
            description
              "BFD MPLS group of sessions. 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-address;
              description "MPLS-FEC";
            }
            list sessions {
              key "local-discriminator";
              description
                "The BFD sessions for an MPLS FEC. Local " +
                "discriminator is unique for each session in the " +
                "group.";
              uses bfd:bfd-all-session;

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

      notification bfd-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:bfd-notification-parms;

        leaf network-instance {
          type string;
          description "Network instance";
        }

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

2.18. BFD over MPLS-TE Yang Module

```
    <CODE BEGINS> file "ietf-bfd-mpls-te@2016-02-04.yang"
    module ietf-bfd-mpls-te {
      namespace "urn:ietf:params:xml:ns:yang:ietf-bfd-mpls-te";
```

```
    // replace with IANA namespace when assigned
    prefix "bfd-mpls-te";

    import ietf-bfd {
      prefix "bfd";
    }

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

    import ietf-te {
      prefix "te";
    }

    organization "IETF BFD Working Group";

    contact
      "WG Web:   <http://tools.ietf.org/wg/bfd>
       WG List:  <rtg-bfd@ietf.org>
       WG Chair: Jeff Haas
       WG Chair: Reshad Rahman
       Editor:   Lianshu Zheng and Reshad Rahman";

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

    revision 2016-02-04 {
      description "Initial revision.";
      reference "RFC XXXX: A YANG data model for BFD over MPLS-TE";
    }

    augment "/bfd:bfd" {
      description "BFD augmentation for MPLS-TE";
      container mpls-te {
        description "BFD MPLS-TE top level container";

        container config {
          description "BFD MPLS-TE configuration container";

          container egress {
            description "Egress configuration";

            uses bfd:bfd-grouping-base-cfg-parms;

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

```
        }

        container oper {
          config "false";
          description "BFD operational container";
          uses bfd:bfd-session-statistics;
        }
      }
    }

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

      uses bfd:bfd-grouping-common-cfg-parms;

      uses bfd-mpls:bfd-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:bfd-all-session;

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


    notification bfd-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:bfd-notification-parms;

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

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

2.19. Security Considerations

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

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

2.20. IANA Considerations

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

URI:TBD

2.21. 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.

3. References

3.1. Normative References

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

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

[RFC5880] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD)", RFC 5880, DOI 10.17487/RFC5880, June 2010, <http://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, <http://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, <http://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, <http://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, <http://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, <http://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, <http://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, <http://www.rfc-editor.org/info/rfc6241>.

[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, <http://www.rfc-editor.org/info/rfc7130>.

3.2. Informative References

[I-D.ietf-netconf-restconf]
Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", draft-ietf-netconf-restconf-18 (work in progress), October 2016.

[I-D.ietf-netmod-routing-cfg]
Lhotka, L. and A. Lindem, "A YANG Data Model for Routing Management", draft-ietf-netmod-routing-cfg-25 (work in progress), November 2016.

[I-D.ietf-rtgwg-ni-model]
Berger, L., Hopps, C., Lindem, A., and D. Bogdanovic, "YANG Network Instances", draft-ietf-rtgwg-ni-model-01 (work in progress), October 2016.

[I-D.ietf-rtgwg-yang-key-chain]
Lindem, A., Qu, Y., Yeung, D., Chen, I., Zhang, Z., and Y. Yang, "Routing Key Chain YANG Data Model", draft-ietf-rtgwg-yang-key-chain-11 (work in progress), November 2016.

[I-D.ietf-teas-yang-te]
Saad, T., Gandhi, R., Liu, X., Beeram, V., Shah, H., Bryskin, I., Chen, X., Jones, R., and B. Wen, "A YANG Data Model for Traffic Engineering Tunnels and Interfaces", draft-ietf-teas-yang-te-05 (work in progress), October 2016.

Appendix A. Change log

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

A.1. 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.

A.2. Changes between versions -02 and -03

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

A.3. Changes between versions -01 and -02

- Replace routing-instance, which has been removed from [I-D.ietf-netmod-routing-cfg], with network-instance from [I-D.ietf-rtgwg-ni-model]

A.4. 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 key-chain in [I-D.ietf-rtgwg-yang-key-chain]

Authors' Addresses

Lianshu Zheng (editor)
Huawei Technologies
China

Email: vero.zheng@huawei.com

Reshad Rahman (editor)
Cisco Systems
Canada

Email: rrahman@cisco.com

Santosh Pallagatti
India

Email: santosh.pallagatti@gmail.com

Mahesh Jethanandani
Cisco Systems

Email: mjethanandani@gmail.com

Greg Mirsky
Ericsson

Email: gregimirsky@gmail.com