ï»¿

Internet Engineering Task Force (IETF)
Request for Comments: 9314
Updates: 9127
Category: Standards Track
ISSN: 2070-1721

M. Jethanandani, Ed.
Xoriant Corporation
R. Rahman, Ed.

L. Zheng, Ed.
Huawei Technologies
S. Pallagatti
VMware
G. Mirsky
Ericsson

September 2022

YANG Data Model for Bidirectional Forwarding Detection (BFD)

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) (RFC 8342). This document updates "YANG Data Model for Bidirectional Forwarding Detection (BFD)" (RFC 9127).

Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 7841.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at https://www.rfc-editor.org/info/rfc9314.

Table of Contents

1. Introduction

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

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

2. Two systems connected via multiple hops as described in "Bidirectional Forwarding Detection (BFD) for Multihop Paths" [RFC5883].

3. Two systems connected via MPLS Label Switched Paths (LSPs) as described in "Bidirectional Forwarding Detection (BFD) for MPLS Label Switched Paths (LSPs)" [RFC5884].

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

5. Two systems connected via pseudowires (PWs). This is known as Virtual Circuit Connectivity Verification (VCCV) as described in "Bidirectional Forwarding Detection (BFD) for the Pseudowire Virtual Circuit Connectivity Verification (VCCV)" [RFC5885]. This scenario 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 Bidirectional Forwarding Detection (BFD)" [RFC5882]. The obvious candidates that use BFD are those that 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 data and operational state data.

1.1. Tree Diagrams

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

2. Design of the Data Model

Since BFD is used for liveness detection of various forwarding paths, there is no uniform key to identify a BFD session. Therefore, the BFD data model is split into 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 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 that 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 the following YANG modules for their respective specific information:

1. The "ietf-bfd-ip-sh" module (Section 2.13) augments "/routing/control-plane-protocols/control-plane-protocol/bfd/" with the "ip-sh" container for BFD sessions over IP single-hop.

2. The "ietf-bfd-ip-mh" module (Section 2.14) augments "/routing/control-plane-protocols/control-plane-protocol/bfd/" with the "ip-mh" container for BFD sessions over IP multihop.

3. The "ietf-bfd-lag" module (Section 2.15) augments "/routing/control-plane-protocols/control-plane-protocol/bfd/" with the "lag" container for BFD sessions over a LAG.

4. The "ietf-bfd-mpls" module (Section 2.16) augments "/routing/control-plane-protocols/control-plane-protocol/bfd/" with the "mpls" container for BFD-over-MPLS LSPs.

BFD can operate in the following contexts:

1. At the network-device level.

2. In logical network elements (LNEs) as described in "YANG Model for Logical Network Elements" [RFC8530].

3. In network instances as described in "YANG Data Model for Network Instances" [RFC8529].

When used at the network device level, the BFD YANG data model is used "as is". When the BFD YANG data model is used in an LNE or network instance, the BFD YANG data model augments the mounted routing model for the LNE or network instance.

2.1. Design of the Configuration Model

The configuration model consists mainly of the parameters specified in BFD [RFC5880] -- for example, desired minimum transmit interval, required minimum receive interval, and detection multiplier.

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, or BGP. Other implementations have BFD session configuration centralized under BFD, i.e., outside the multiple BFD clients.

The main BFD parameters of interest to a BFD client are those related to 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. Configuration of BFD for all clients should be centralized. However, this is a problem for BFD clients that 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 the grouping "client-cfg-parms" in Section 2.11 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; similarly, when an IGP peer goes away, the IGP would remove the BFD session to that peer. The mechanism for how the BFD sessions are created and removed by the BFD clients is outside the scope of this document, but this would typically be done by using an API implemented by the BFD module on the system. In the case of BFD clients that 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 as follows:

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 that is used for both transmit and receive intervals, or separate values for transmit and receive intervals. The BFD YANG data 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 the "base-cfg-parms" grouping (Section 2.11), which is used by the YANG modules for the various forwarding paths.

For BFD authentication, we have the following:

key-chain
: This is a reference to "key-chain" as 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 a 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, as described 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. We use the configuration parameters defined in Section 2.1.1. The "ip-sh" node also contains a list of interfaces and is used to specify authentication parameters for BFD sessions that are created by BFD clients. See Section 2.1.

[RFC5880] and [RFC5881] do not specify whether the Echo function operates continuously or on demand. Therefore, the mechanism used to

start and stop the Echo function is implementation specific and should be done by augmentation:

1. Configuration. This is suitable for an Echo function that operates continuously. An example is provided in Appendix A.

2. RPC. This is suitable for an Echo function that operates on demand.

2.1.3. Multihop IP

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

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

Source address
: Address belonging to the local system as per "Bidirectional Forwarding Detection (BFD) for Multihop Paths" [RFC5883].

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

We use the configuration parameters defined in Section 2.1.1.

This document also provides the following parameters:

tx-ttl
: TTL of outgoing BFD control packets.

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

2.1.4. MPLS Label Switched Paths

Here, we address MPLS LSPs whose Forwarding Equivalence Class (FEC) [RFC3031] is an IP address. The "bfd" node (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 set of sessions as a "session-group".

Since these LSPs are unidirectional, there is no LSP configuration on the egress node.

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

2.1.5. Link Aggregation Groups

Per "Bidirectional Forwarding Detection (BFD) on Link Aggregation Group (LAG) Interfaces" [RFC7130], configuring BFD on a 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 data model that we can augment. So, a "lag" data node is added to the "bfd" node; see 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 the Operational State Model

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

The overall statistics for the BFD sessions consist of the number of BFD sessions, the number of BFD sessions that are up, etc. This information is available globally (i.e., for all BFD sessions) under

the "bfd" node (Section 2) and also per type of forwarding path.

For each BFD session, three main categories of operational state data are shown.

1. The first category includes fundamental information regarding a BFD session, such as the local discriminator, the remote discriminator, and the ability to support Demand mode.

2. The second category includes 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 configured desired minimum transmit interval. Similar examples include the actual receive interval between the control packets and the actual transmit interval between the Echo packets.

3. The third category contains the detailed statistics for 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 one 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 data model defines notifications to inform end users of important events detected during the protocol operation. The local discriminator identifies the corresponding BFD session on the local system, and the remote discriminator identifies the BFD session on the remote 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 consists 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 the "bfd" node in "control-plane-protocol". The configuration data and operational state data for each BFD IP single-hop session are 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?    key-chain: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?    key-chain: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 "control-plane-protocol". The configuration data and operational state data for each BFD IP multihop session are 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 the 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?    key-chain: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 data and operational state data for each BFD LAG session are 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?    key-chain: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 the key. The "mpls" node can be used in a network device (top level) or can be mounted in an LNE or 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 enabled?                          boolean
       |  +--rw local-multiplier?                 multiplier
       |  +--rw (interval-config-type)?
       |  |  +--:(tx-rx-intervals)
```

```
|  |  |  +--rw desired-min-tx-interval?    uint32
|  |  |  +--rw required-min-rx-interval?   uint32
|  |  +--:(single-interval) {single-minimum-interval}?
|  |     +--rw min-interval?               uint32
|  +--rw authentication! {authentication}?
|     +--rw key-chain?    key-chain: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?    key-chain: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. Interaction with Other YANG Modules

"Generic YANG Data Model for the Management of Operations, Administration, and Maintenance (OAM) Protocols That Use Connectionless Communications" [RFC8532] describes how the Layer-Independent OAM Management in the Multi-Layer Environment (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.

2.10.1. "ietf-interfaces" Module

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.

2.10.2. "ietf-ip" Module

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.

2.10.3. "ietf-mpls" Module

The following boolean configuration is defined in "A YANG Data Model for MPLS Base" [RFC8960]:

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

2.11. BFD Types YANG Module

This YANG module imports typedefs from [RFC6991] and [RFC8177]. It also imports definitions from [RFC5880], [RFC5881], [RFC5883], [RFC5884], and [RFC7130], as well as the "control-plane-protocol" identity from [RFC8349], and references [RFC9127].

<CODE BEGINS> file "ietf-bfd-types@2022-09-22.yang"

```
module ietf-bfd-types {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-bfd-types";
  prefix bfd-types;

  import iana-bfd-types {
    prefix iana-bfd-types;
    reference
      "RFC 9127: YANG Data Model for Bidirectional Forwarding
       Detection (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 key-chain;
    reference
      "RFC 8177: YANG Data Model for Key Chains";
  }

  organization
    "IETF BFD Working Group";
  contact
    "WG Web:   <https://datatracker.ietf.org/wg/bfd/>
     WG List:  <mailto:rtg-bfd@ietf.org>

     Editor:   Reshad Rahman
               <mailto:reshad@yahoo.com>

     Editor:   Lianshu Zheng
               <mailto:veronique_cheng@hotmail.com>

     Editor:   Mahesh Jethanandani
               <mailto:mjethanandani@gmail.com>";
  description
    "This module contains a collection of BFD-specific YANG data type
     definitions, as per RFC 5880, and also groupings that are common
     to other BFD YANG modules.

     Copyright (c) 2022 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 Revised BSD License
     set forth in Section 4.c of the IETF Trust's Legal Provisions
     Relating to IETF Documents
     (https://trustee.ietf.org/license-info).

     This version of this YANG module is part of RFC 9314; see the
     RFC itself for full legal notices.";
  reference
    "RFC 5880: Bidirectional Forwarding Detection (BFD)
     RFC 9314: YANG Data Model for Bidirectional Forwarding
     Detection (BFD)";

  revision 2022-09-22 {
    description
```

```
    "This revision is not backwards compatible with the
     previous version of this model.

     This revision adds an 'if-feature' statement called
     'client-base-cfg-parms' for client configuration parameters.
     Clients expecting to use those parameters now need to
     verify that the server declares support of the feature
     before depending on the presence of the parameters.

     The change was introduced for clients that do not need
     them and have to deviate to prevent them from being
     included.";
  reference
    "RFC 9314: YANG Data Model for Bidirectional Forwarding
     Detection (BFD).";
}
revision 2021-10-21 {
  description
    "Initial revision.";
  reference
    "RFC 9127: YANG Data Model for Bidirectional Forwarding
     Detection (BFD)";
}

/*
 * Feature definitions
 */

feature single-minimum-interval {
  description
    "This feature indicates that the server supports configuration
     of one minimum interval value that 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";
}

feature client-base-cfg-parms {
  description
    "This feature allows protocol models to configure BFD client
     session parameters.";
  reference
    "RFC 9314: YANG Data Model for Bidirectional Forwarding
                Detection (BFD).";
}
```

```
/*
 * 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 the 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 an 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.";
  reference
    "RFC 5880: Bidirectional Forwarding Detection (BFD)";
}

typedef state {
  type enumeration {
    enum adminDown {
      value 0;
      description
        "'adminDown' state.";
    }
    enum down {
      value 1;
      description
        "'Down' state.";
    }
    enum init {
      value 2;
      description
        "'Init' state.";
    }
    enum up {
      value 3;
      description
        "'Up' state.";
    }
  }
  description
    "BFD states 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
     the 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.";
    reference
      "RFC 5880: Bidirectional Forwarding Detection (BFD),
       Section 6.7";
    leaf key-chain {
      type key-chain:key-chain-ref;
      description
        "Name of the 'key-chain' as per RFC 8177.";
    }
    leaf meticulous {
      type boolean;
      description
        "Enables a meticulous mode as per Section 6.7 of
         RFC 5880.";
    }
  }
}

grouping base-cfg-parms {
  description
    "BFD grouping for base configuration parameters.";
  leaf local-multiplier {
    type multiplier;
    default "3";
    description
      "Multiplier transmitted by the local system.";
  }
  choice interval-config-type {
    default "tx-rx-intervals";
    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 BFD clients, e.g., IGP or MPLS.";
  leaf enabled {
    type boolean;
    default "false";
    description
```

```
        "Indicates whether BFD is enabled.";
    }
    uses base-cfg-parms {
      if-feature "client-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
        "Indicates whether the BFD session is 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
        "Indicates 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 the last time this session went down.";
  }
  leaf last-up-time {
    type yang:date-and-time;
    description
      "Time and date of the last time this session went up.";
  }
```

```
    leaf down-count {
      type yang:counter32;
      description
        "The number of times this session has transitioned to the
         'down' state.";
    }
    leaf admin-down-count {
      type yang:counter32;
      description
        "The number of times this session has transitioned to 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 that 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 the 'Up' state
         (as defined in RFC 5880).";
    }
    leaf number-of-sessions-down {
      type yang:gauge32;
      description
        "Number of BFD sessions currently in the 'Down' or 'Init'
         state but not 'adminDown' (as defined in RFC 5880).";
    }
    leaf number-of-sessions-admin-down {
      type yang:gauge32;
      description
        "Number of BFD sessions currently in the 'adminDown' state
         (as defined in RFC 5880).";
    }
  }
}

grouping notification-parms {
  description
```

```
        "This group describes common parameters that will be sent
         as part of BFD notifications.";
      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
          "Reason for the BFD state change.";
      }
      leaf time-of-last-state-change {
        type yang:date-and-time;
        description
          "Calendar time of the most recent 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>
```

2.12. BFD Top-Level YANG Module

This YANG module imports and augments "/routing/control-plane-protocols/control-plane-protocol" from [RFC8349]. It also references [RFC5880].

```
  <CODE BEGINS> file "ietf-bfd@2022-09-22.yang"
  module ietf-bfd {
    yang-version 1.1;
    namespace "urn:ietf:params:xml:ns:yang:ietf-bfd";
    prefix bfd;

    import ietf-bfd-types {
      prefix bfd-types;
      reference
        "RFC 9314: YANG Data Model for Bidirectional Forwarding
         Detection (BFD)";
    }
```

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

organization
  "IETF BFD Working Group";
contact
  "WG Web:   <https://datatracker.ietf.org/wg/bfd/>
   WG List:  <mailto:rtg-bfd@ietf.org>

   Editor:   Reshad Rahman
             <mailto:reshad@yahoo.com>

   Editor:   Lianshu Zheng
             <mailto:veronique_cheng@hotmail.com>

   Editor:   Mahesh Jethanandani
             <mailto:mjethanandani@gmail.com>";
description
  "This module contains the YANG definition for BFD parameters as
   per RFC 5880.

   Copyright (c) 2022 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 Revised BSD License set
   forth in Section 4.c of the IETF Trust's Legal Provisions
   Relating to IETF Documents
   (https://trustee.ietf.org/license-info).

   This version of this YANG module is part of RFC 9314; see the
   RFC itself for full legal notices.";
reference
  "RFC 5880: Bidirectional Forwarding Detection (BFD)
   RFC 9314: YANG Data Model for Bidirectional Forwarding
   Detection (BFD)";

revision 2022-09-22 {
  description
    "Updating reference to RFC 9314.";
  reference
    "RFC 9314: YANG Data Model for Bidirectional Forwarding
     Detection (BFD).";
}
revision 2021-10-21 {
  description
    "Initial revision.";
  reference
    "RFC 9127: YANG Data Model for Bidirectional Forwarding
     Detection (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>
```

2.13. BFD IP Single-Hop YANG Module

This YANG module imports "interface-ref" from [RFC8343] and typedefs from [RFC6991]. It also imports and augments "/routing/control-plane-protocols/control-plane-protocol" from [RFC8349], and it references [RFC5881].

```
  <CODE BEGINS> file "ietf-bfd-ip-sh@2022-09-22.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;

    import ietf-bfd-types {
      prefix bfd-types;
      reference
        "RFC 9314: YANG Data Model for Bidirectional Forwarding
         Detection (BFD)";
    }
    import ietf-bfd {
      prefix bfd;
      reference
        "RFC 9314: YANG Data Model for Bidirectional Forwarding
         Detection (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:   <https://datatracker.ietf.org/wg/bfd/>
       WG List:  <mailto:rtg-bfd@ietf.org>

       Editor:   Reshad Rahman
                 <mailto:reshad@yahoo.com>

       Editor:   Lianshu Zheng
                 <mailto:veronique_cheng@hotmail.com>

       Editor:   Mahesh Jethanandani
                 <mailto:mjethanandani@gmail.com>";
    description
      "This module contains the YANG definition for BFD IP single-hop
       as per RFC 5881.

       Copyright (c) 2022 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 Revised BSD License
       set forth in Section 4.c of the IETF Trust's Legal Provisions
```

```
   Relating to IETF Documents
   (https://trustee.ietf.org/license-info).

   This version of this YANG module is part of RFC 9314; see the
   RFC itself for full legal notices.";
reference
  "RFC 5881: Bidirectional Forwarding Detection (BFD)
   for IPv4 and IPv6 (Single Hop)
   RFC 9314: YANG Data Model for Bidirectional Forwarding
   Detection (BFD)";

revision 2022-09-22 {
  description
    "Updating reference to RFC 9314.";
  reference
    "RFC 9314: YANG Data Model for Bidirectional Forwarding
     Detection (BFD).";
}
revision 2021-10-21 {
  description
    "Initial revision.";
  reference
    "RFC 9127: YANG Data Model for Bidirectional Forwarding
     Detection (BFD)";
}

/*
 * 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.";
      }
      leaf echo-enabled {
        type boolean;
        description
          "Indicates whether Echo was enabled for BFD.";
      }
    }
  }
  <CODE ENDS>
```

2.14. BFD IP Multihop YANG Module

This YANG module imports typedefs from [RFC6991]. It also imports and augments "/routing/control-plane-protocols/control-plane-protocol" from [RFC8349], and it references [RFC5883].

```
  <CODE BEGINS> file "ietf-bfd-ip-mh@2022-09-22.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;

    import ietf-bfd-types {
      prefix bfd-types;
      reference
        "RFC 9314: YANG Data Model for Bidirectional Forwarding
         Detection (BFD)";
    }
    import ietf-bfd {
      prefix bfd;
      reference
        "RFC 9314: YANG Data Model for Bidirectional Forwarding
         Detection (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:   <https://datatracker.ietf.org/wg/bfd/>
```

```
   WG List:  <mailto:rtg-bfd@ietf.org>

   Editor:   Reshad Rahman
             <mailto:reshad@yahoo.com>

   Editor:   Lianshu Zheng
             <mailto:veronique_cheng@hotmail.com>

   Editor:   Mahesh Jethanandani
             <mailto:mjethanandani@gmail.com>";
  description
    "This module contains the YANG definition for BFD IP multihop
     as per RFC 5883.

     Copyright (c) 2022 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 Revised BSD License set
     forth in Section 4.c of the IETF Trust's Legal Provisions
     Relating to IETF Documents
     (https://trustee.ietf.org/license-info).

     This version of this YANG module is part of RFC 9314; see the
     RFC itself for full legal notices.";
  reference
    "RFC 5883: Bidirectional Forwarding Detection (BFD) for
     Multihop Paths
     RFC 9314: YANG Data Model for Bidirectional Forwarding
     Detection (BFD)";

  revision 2022-09-22 {
    description
      "Updating reference to RFC 9314.";
    reference
      "RFC 9314: YANG Data Model for Bidirectional Forwarding
       Detection (BFD).";
  }
  revision 2021-10-21 {
    description
      "Initial revision.";
    reference
      "RFC 9127: YANG Data Model for Bidirectional Forwarding
       Detection (BFD)";
  }

  /*
   * 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 multihop session groups.";
        list session-group {
          key "source-addr dest-addr";
          description
            "Group of BFD IP multihop sessions (for ECMP).  A
             group of sessions is between one source and one
             destination.  Each session has a different field
             in the UDP/IP header 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 multihop 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>
```

2.15. BFD-over-LAG YANG Module

This YANG module imports "interface-ref" from [RFC8343] and typedefs from [RFC6991]. It also imports and augments "/routing/control-plane-protocols/control-plane-protocol" from [RFC8349]. Additionally, it references [RFC7130].

```
  <CODE BEGINS> file "ietf-bfd-lag@2022-09-22.yang"
  module ietf-bfd-lag {
    yang-version 1.1;
    namespace "urn:ietf:params:xml:ns:yang:ietf-bfd-lag";
    prefix bfd-lag;

    import ietf-bfd-types {
      prefix bfd-types;
      reference
        "RFC 9314: YANG Data Model for Bidirectional Forwarding
         Detection (BFD)";
    }
    import ietf-bfd {
      prefix bfd;
      reference
```

```
    "RFC 9314: YANG Data Model for Bidirectional Forwarding
     Detection (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:   <https://datatracker.ietf.org/wg/bfd/>
   WG List:  <mailto:rtg-bfd@ietf.org>

   Editor:   Reshad Rahman
             <mailto:reshad@yahoo.com>

   Editor:   Lianshu Zheng
             <mailto:veronique_cheng@hotmail.com>

   Editor:   Mahesh Jethanandani
             <mailto:mjethanandani@gmail.com>";
description
  "This module contains the YANG definition for BFD-over-LAG
   interfaces as per RFC 7130.

   Copyright (c) 2022 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 Revised BSD License set
   forth in Section 4.c of the IETF Trust's Legal Provisions
   Relating to IETF Documents
   (https://trustee.ietf.org/license-info).

   This version of this YANG module is part of RFC 9314; see the
   RFC itself for full legal notices.";
reference
  "RFC 7130: Bidirectional Forwarding Detection (BFD) on
   Link Aggregation Group (LAG) Interfaces
   RFC 9314: YANG Data Model for Bidirectional Forwarding
   Detection (BFD)";

revision 2022-09-22 {
  description
    "Updating reference to RFC 9314.";
  reference
    "RFC 9314: YANG Data Model for Bidirectional Forwarding
     Detection (BFD).";
}
revision 2021-10-21 {
  description
    "Initial revision.";
  reference
    "RFC 9127: YANG Data Model for Bidirectional Forwarding
     Detection (BFD)";
}
```

```
/*
 * Augments
 */

augment "/rt:routing/rt:control-plane-protocols/"
      + "rt:control-plane-protocol/bfd:bfd" {
  description
    "BFD augmentation for a 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 a 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 a 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 a 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>
```

2.16. BFD-over-MPLS YANG Module

This YANG module imports typedefs from [RFC6991]. It also imports and augments "/routing/control-plane-protocols/control-plane-protocol" from [RFC8349]. Additionally, it references [RFC5586] and [RFC5884].

```
  <CODE BEGINS> file "ietf-bfd-mpls@2022-09-22.yang"
  module ietf-bfd-mpls {
    yang-version 1.1;
    namespace "urn:ietf:params:xml:ns:yang:ietf-bfd-mpls";
    prefix bfd-mpls;

    import ietf-bfd-types {
      prefix bfd-types;
      reference
        "RFC 9314: YANG Data Model for Bidirectional Forwarding
         Detection (BFD)";
    }
    import ietf-bfd {
      prefix bfd;
      reference
        "RFC 9314: YANG Data Model for Bidirectional Forwarding
         Detection (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:   <https://datatracker.ietf.org/wg/bfd/>
   WG List:  <mailto:rtg-bfd@ietf.org>

   Editor:   Reshad Rahman
             <mailto:reshad@yahoo.com>

   Editor:   Lianshu Zheng
             <mailto:veronique_cheng@hotmail.com>

   Editor:   Mahesh Jethanandani
             <mailto:mjethanandani@gmail.com>";
description
  "This module contains the YANG definition for BFD parameters for
   MPLS LSPs as per RFC 5884.

   Copyright (c) 2022 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 Revised BSD License set
   forth in Section 4.c of the IETF Trust's Legal Provisions
   Relating to IETF Documents
   (https://trustee.ietf.org/license-info).

   This version of this YANG module is part of RFC 9314; see the
   RFC itself for full legal notices.";
reference
  "RFC 5884: Bidirectional Forwarding Detection (BFD)
   for MPLS Label Switched Paths (LSPs)
   RFC 9314: YANG Data Model for Bidirectional Forwarding
   Detection (BFD)";

revision 2022-09-22 {
  description
    "Updates to use base-cfg-parms instead of client-cfg-parms,
     and add the enabled flag.";
  reference
    "RFC 9314: YANG Data Model for Bidirectional Forwarding
     Detection (BFD).";
}
revision 2021-10-21 {
  description
    "Initial revision.";
  reference
    "RFC 9127: YANG Data Model for Bidirectional Forwarding
     Detection (BFD)";
}

/*
 * Identity definitions
 */

identity encap-gach {
  base bfd-types:encap-type;
  description
    "BFD with Generic Associated Channel (G-ACh) encapsulation
     as per RFC 5586.";
  reference
    "RFC 5586: MPLS Generic Associated Channel";
```

```
}

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.";
  reference
    "RFC 5884: Bidirectional Forwarding Detection (BFD)
     for MPLS Label Switched Paths (LSPs)";
  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.";
      leaf enabled {
        type boolean;
        default "false";
        description
          "Indicates whether BFD over MPLS is enabled.";
      }
      uses bfd-types:base-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 one FEC.  Each session has a different
               field in the UDP/IP header 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.  The 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>
```

3. Data Model Examples

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

The examples are represented in XML [W3C.REC-xml-20081126].

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 10 ms.

```
<?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 150 ms.

```
  <?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 "ieee8023adLag" has a desired transmit interval and required receive interval set to 10 ms.

```
  <?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>
                    10000
                  </desired-min-tx-interval>
                  <required-min-rx-interval>
                    10000
                  </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 interval and required receive interval are both set to 250 ms.

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

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

There are a number of data nodes defined in these YANG modules 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 from a write access perspective:

/routing/control-plane-protocols/control-plane-protocol/bfd/ip-sh/sessions:

This list specifies the IP single-hop BFD 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 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:

This list specifies the IP multihop BFD session groups.

Data nodes "local-multiplier", "desired-min-tx-interval", "required-min-rx-interval", and "min-interval" all impact the BFD IP multihop session. The "source-addr" and "dest-addr" data nodes can be used to send BFD packets to unwitting recipients. [RFC5880] describes how BFD mitigates such threats. Authentication data nodes "key-chain" and "meticulous" impact the security of the BFD IP multihop session.

/routing/control-plane-protocols/control-plane-protocol/bfd/lag/sessions:

This list specifies the BFD sessions over a LAG.

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 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:

This list specifies the session groups for BFD over MPLS.

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.

The YANG modules have writable data nodes that can be used for the creation of BFD sessions and the modification of BFD session parameters. The system should "police" the creation of BFD sessions to prevent new sessions from causing existing BFD sessions to fail. In the case of BFD session modification, the BFD protocol has mechanisms in place that 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 system that uses BFD (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 acting as a BFD client, attacking the IGP could cause more broad-scale disruption than would (de)configuring a BFD session.

Some of the readable data nodes in these YANG modules 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 from a read access perspective:

/routing/control-plane-protocols/control-plane-protocol/bfd/ip-sh/
summary:
   Access to this information discloses the number of BFD IP single-
   hop sessions that are in the "up", "down", or "admin-down" state.
   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 multihop
   sessions that are in the "up", "down", or "admin-down" state. 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 multihop 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 that are in the "up", "down", or "admin-down" state.
   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 that are in the "up", "down", or "admin-down" state. 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 that are in the "up", "down", or "admin-down" state. 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.

This document does not define any RPC operations.

5. IANA Considerations

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

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-ip-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.

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

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

Reference: RFC 9314

Name: ietf-bfd
Namespace: urn:ietf:params:xml:ns:yang:ietf-bfd
Prefix: bfd
Reference: RFC 9314

Name: ietf-bfd-ip-sh
Namespace: urn:ietf:params:xml:ns:yang:ietf-bfd-ip-sh
Prefix: bfd-ip-sh
Reference: RFC 9314

Name: ietf-bfd-ip-mh
Namespace: urn:ietf:params:xml:ns:yang:ietf-bfd-ip-mh
Prefix: bfd-ip-mh
Reference: RFC 9314

Name: ietf-bfd-lag
Namespace: urn:ietf:params:xml:ns:yang:ietf-bfd-lag
Prefix: bfd-lag
Reference: RFC 9314

Name: ietf-bfd-mpls
Namespace: urn:ietf:params:xml:ns:yang:ietf-bfd-mpls
Prefix: bfd-mpls
Reference: RFC 9314

6. References

6.1. Normative References

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

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

[RFC5880] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD)", RFC 5880, DOI 10.17487/RFC5880, June 2010, <https://www.rfc-editor.org/info/rfc5880>.

[RFC5881] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD) for IPv4 and IPv6 (Single Hop)", RFC 5881, DOI 10.17487/RFC5881, June 2010, <https://www.rfc-editor.org/info/rfc5881>.

[RFC5882] Katz, D. and D. Ward, "Generic Application of Bidirectional Forwarding Detection (BFD)", RFC 5882, DOI 10.17487/RFC5882, June 2010, <https://www.rfc-editor.org/info/rfc5882>.

[RFC5883] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD) for Multihop Paths", RFC 5883, DOI 10.17487/RFC5883, June 2010, <https://www.rfc-editor.org/info/rfc5883>.

[RFC5884] Aggarwal, R., Kompella, K., Nadeau, T., and G. Swallow, "Bidirectional Forwarding Detection (BFD) for MPLS Label Switched Paths (LSPs)", RFC 5884, DOI 10.17487/RFC5884, June 2010, <https://www.rfc-editor.org/info/rfc5884>.

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

[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>.

[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>.

[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>.

[RFC8341] Bierman, A. and M. Bjorklund, "Network Configuration Access Control Model", STD 91, RFC 8341, DOI 10.17487/RFC8341, March 2018, <https://www.rfc-editor.org/info/rfc8341>.

[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>.

[RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <https://www.rfc-editor.org/info/rfc8446>.

[RFC8960] Saad, T., Raza, K., Gandhi, R., Liu, X., and V. Beeram, "A YANG Data Model for MPLS Base", RFC 8960, DOI 10.17487/RFC8960, December 2020, <https://www.rfc-editor.org/info/rfc8960>.

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

6.2. Informative References

[RFC3031] Rosen, E., Viswanathan, A., and R. Callon, "Multiprotocol

Label Switching Architecture", RFC 3031,
DOI 10.17487/RFC3031, January 2001,
<https://www.rfc-editor.org/info/rfc3031>.

[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>.

[RFC8529] Berger, L., Hopps, C., Lindem, A., Bogdanovic, D., and X. Liu, "YANG Data Model for Network Instances", RFC 8529, DOI 10.17487/RFC8529, March 2019, <https://www.rfc-editor.org/info/rfc8529>.

[RFC8530] Berger, L., Hopps, C., Lindem, A., Bogdanovic, D., and X. Liu, "YANG Model for Logical Network Elements", RFC 8530, DOI 10.17487/RFC8530, March 2019, <https://www.rfc-editor.org/info/rfc8530>.

[RFC8532] Kumar, D., Wang, Z., Wu, Q., Ed., Rahman, R., and S. Raghavan, "Generic YANG Data Model for the Management of Operations, Administration, and Maintenance (OAM) Protocols That Use Connectionless Communications", RFC 8532, DOI 10.17487/RFC8532, April 2019, <https://www.rfc-editor.org/info/rfc8532>.

[W3C.REC-xml-20081126]
Bray, T., Paoli, J., Sperberg-McQueen, M., Maler, E., and F. Yergeau, "Extensible Markup Language (XML) 1.0 (Fifth Edition)", World Wide Web Consortium Recommendation REC-xml-20081126, November 2008, <https://www.w3.org/TR/2008/REC-xml-20081126>.

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 discussed in [RFC5881], is implementation specific. In this appendix, 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

This appendix provides an example YANG module for configuration of the BFD Echo function. It imports and augments "/routing/control-plane-protocols/control-plane-protocol" from [RFC8349], and it references [RFC5880].

```
module example-bfd-echo {
  namespace "tag:example.com,2021: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:   <https://datatracker.ietf.org/wg/bfd/>
   WG List:  <mailto:rtg-bfd@ietf.org>

   Editor:   Reshad Rahman
             <mailto:reshad@yahoo.com>

   Editor:   Lianshu Zheng
             <mailto:veronique_cheng@hotmail.com>

   Editor:   Mahesh Jethanandani
             <mailto:mjethanandani@gmail.com>";
description
  "This module contains an example YANG augmentation for
   configuration of the BFD Echo function.

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

   Redistribution and use in source and binary forms, with or
   without modification, is permitted pursuant to, and subject
   to the license terms contained in, the Revised BSD License
   set forth in Section 4.c of the IETF Trust's Legal Provisions
   Relating to IETF Documents
   (https://trustee.ietf.org/license-info).

   This version of this YANG module is part of RFC 9127; see the
   RFC itself for full legal notices.";

revision 2021-10-21 {
  description
    "Initial revision.";
  reference
    "RFC 9127: YANG Data Model for Bidirectional Forwarding
     Detection (BFD)";
}

/*
 * Groupings
 */

grouping echo-cfg-parms {
  description
    "BFD grouping for Echo configuration 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 (RFC 5880) 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
       (RFC 5880).";
  }
}

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 the BFD Echo function.";
      container echo {
        if-feature "bfd-types:echo-mode";
        description
          "BFD Echo function container.";
        uses echo-cfg-parms;
      }
    }
  }
```

Appendix B. Updates since RFC 9127

This document updates the 'ietf-bfd-types' module to define a new feature called 'client-base-cfg-parms and an 'if-feature' statement that conditionally includes definitions of parameters, such as 'multiplier' or 'desired-min-tx-interval'. The feature statement allows YANG implementations of protocols, such as OSPF, IS-IS, PIM, and BGP, to support both a model where such parameters are not needed, such as when multiple BFD sessions are supported over a given interface, as well as when they need to be defined per session. As a result, the BFD MPLS module has to use the base-cfg-parms instead of client-cfg-parms to be able to include all the parameters unconditionally.

The iana-bfd-types module, created in RFC 9127, was delegated to IANA for maintenance. No changes are requested from IANA as part of this update.

Acknowledgments

We would like to thank Nobo Akiya and Jeff Haas for their encouragement on this work. We would also like to thank Tom Petch for his comments on the document. We would also like to thank Acee Lindem for his guidance. Thanks also to JÃ¼rgen SchÃ¶nwÃ¤lder, who was instrumental in improving the YANG modules.

Authors' Addresses

Mahesh Jethanandani (editor)
Xoriant Corporation
1248 Reamwood Ave
Sunnyvale, CA 94089
United States of America
Email: mjethanandani@gmail.com

Reshad Rahman (editor)
Canada
Email: reshad@yahoo.com

Lianshu Zheng (editor)
Huawei Technologies
China
Email: veronique_cheng@hotmail.com

Santosh Pallagatti
VMware
India
Email: santosh.pallagatti@gmail.com

Greg Mirsky
Ericsson
Email: gregimirsky@gmail.com