A Layer Independent Operations, Administration, and Maintenance (OAM) Management YANG Data Model Extension for BFD Support
draft-wang-yang-bfd-oam-01

Abstract

This document presents a YANG Data model for BFD support. The YANG Model presented in this document extends the YANG model for Layer independent OAM Management with BFD technology specifics.

Status of This Memo

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

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

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

This Internet-Draft will expire on September 5, 2015.

Copyright Notice

Copyright (c) 2015 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must
1. Introduction

This document presents a YANG Data model for BFD [RFC5880] support. The YANG Model presented in this document extends the YANG model for Layer Independent OAM Management defined in [I-D.tissa-lime-yang-oam-model] and [I-D.wang-lime-rpc-yang-oam-management].

2. Conventions and Terminology

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

The following terms are defined in [RFC6241] and are not redefined here:

- client
- configuration data
- server
- state data
The following terms are defined in [RFC6020] and are not redefined here:

- augment
- data model
- data node

The terminology for describing YANG data models is found in [RFC6020].

2.1.  Tree Diagrams

A simplified graphical representation of the data model is used in this document. The meaning of the symbols in these diagrams is as follows:

Each node is printed as:

<status> <flags> <name> <opts> <type>

<status> is one of:
  + for current
  x for deprecated
  o for obsolete

<flags> is one of:

  rw for configuration data
  ro for non-configuration data
  -x for rpcs
  -n for notifications

<name> is the name of the node

If the node is augmented into the tree from another module, its name is printed as <prefix>:<name>.   
<opts> is one of:
  
  ?  for an optional leaf or choice
  !  for a presence container
  *  for a leaf-list or list
  [<keys>] for a list’s keys

<type> is the name of the type for leaves and leaf-lists

3. Architecture of OAM YANG Model and Relationship to BFD OAM

Layer independent OAM YANG model[I-D.tissa-lime-yang-oam-model] and
[I-D.wang-lime-rpc-yang-oam-management] is used as the basis for all
the other OAM YANG models. This allows users to span across OAM
tools of different technologies through a uniform API. The following
Figure depicts the relationship of BFD OAM YANG model to the Layer
Independent OAM YANG Model.

```
   +-------------+   +-------------+   +-------------+   +-------------+
   |             |   |             |   |             |   |             |
   | Layer       |   | independent |   | OAM YANG    |
   |-------------|   |-------------|   |-------------|
     O
   |-------------|   |-------------|   |-------------|
   |     +-------+   |     +-------+   |     +-------+   |     +-------+   |
   |     | TRILL   |   |     | NVO3    |   |     | BFD     |   |     | foo    |   |     | OAM YANG |
   |     | OAM YANG |   |     | OAM YANG |   |     | OAM YANG |   |     | OAM YANG |
   |     +-------+   |     +-------+   |     +-------+   |     +-------+   |
   +-------------+   +-------------+   +-------------+   +-------------+
     | Uniform API
     +-------------+
```

Relationship of BFD OAM YANG model to Layer independent OAM YANG
model

4. Generic YANG Model extension for BFD

The Technology parameter is defined in the [I-D.tissa-lime-yang-oam-
model] and [I-D.wang-lime-rpc-yang-oam-management] as an identity.
This allows easy extension of the YANG model by other technologies. Technology
specific extensions are applied only when the technology
is set to the specific type. "bfd" is defined as an identity that augments the base technology-types identity.

```yang
identity bfd {
    base goam:technology-types;
    description "bfd type";
}
```

### 4.1. BFD-session grouping

In BFD session, there are some useful parameters be used to indicate some specific type or operation mode. This document define a bfd-session grouping which include all of these parameters. Using this bfd-session grouping, we can augment the basic gen-oam model as follows:

```yang
grouping bfd-session{
    leaf bfdsess-version-num{
        type uint32;
        default "1";
        description "The version number of the BFD protocol that this session is running in. Write access is available for this object to provide the ability to set the desired version for this BFD session ";
    }
    leaf bfdsess-type{
        type enumeration{
            enum singleHop{value 1;}
            enum multiHopTotallyArbitraryPaths{value 2;}
            enum multiHopOutOfBandSignaling{value 3;}
            enum multiHopUnidirectionalLinks{value 4;}
        }
        description "This leaf specifies the type of this BFD session ";
    }
    leaf bfdsess-oper-mode {
        type enumeration{
            enum asynchModeWEchoFunction{value 1;}
            enum asynchModeWOEchoFunction{value 2;}
            enum demandModeWEchoFunction{value 3;}
            enum demandModeWOEchoFunction{value 4;}
        }
        description "This object specifies the operational mode of this BFD session ";
    }
}
```
leaf bfd sess-demandmode-desired {
    type boolean;
    description
        " This leaf indicates the local system's desire to use Demand mode. Specifically, it is set to true if the local system wishes to use Demand mode or false if not ";
}

leaf bfd sess-controlplanel-indep {
    type boolean;
    description
        " This leaf indicates the local system's ability to continue to function through a disruption of the control plane. Specifically, it is set to true if the local system BFD implementation is independent of the control plane. Otherwise, the value is set to false ";
}

leaf bfd sess-multi-point {
    type boolean;
    description
        " This leaf indicates the Multipoint (M) bit for this session. It is set to true if the Multipoint (M) bit is set to 1. Otherwise, the value is set to false ";
}

leaf bfd sess-detect-mult {
    type uint32;
    description
        " This leaf specifies the Detect time multiplier ";
}

leaf bfd sess-authpres {
    type boolean;
    description
        " This leaf indicates the local system's desire to use authentication. Specifically, it is set to true if the local system wishes the session to be authenticated or false if not";
}

leaf bfd sess-authentication-type {
    type enumeration{
        enum noAuthentication;
        enum reserved;
    }
}
enum simplePassword;
enum keyedMD5;
enum meticulousKeyedMD5;
enum keyedSHA1;
enum meticulousKeyedSHA1;
}
description
"The authentication type used for this BFD session. This field is valid only when the Authentication Present bit is set."

leaf bfdSess-authentication-key {
type uint32;
description
"This leaf specifies the authentication key in use for this session."
}

leaf bfdSess-authentication-keyID{
type uint32;
description
"The authentication key ID in use for this session. This leaf permits multiple keys to be active simultaneously."
}

leaf bfdSess-DesiredMinTx-Interval{
description
"This leaf specifies the minimum interval, in microseconds, that the local system would like to use when transmitting BFD Control packets. The value of zero(0) is reserved in this case and should not be used."

type uint32;
}

leaf bfdSess-ReqMinRx-Interval {
description
"This leaf specifies the minimum interval, in microseconds, between received BFD Control packets that the local system is capable of supporting. The value of zero(0) can be specified when the transmitting system does not want the remote system to send any periodic BFD Control packets."

type uint32;
}

leaf bfdSess-ReqMinEchoRx-Interval {
description
"This leaf specifies the minimum interval, in microseconds, between received BFD Echo packets that this system is capable of supporting. The value must be zero(0) if
this is a multihop BFD session ";
  type uint32;
}
leaf bfdSess-Negotiated-Interval {
  description
  " This leaf specifies the negotiated interval, in
  microseconds, that the local system is transmitting
  BFD Control packets ";
  type uint32;
}
leaf bfdSess-NegotiatedEcho-Interval {
  description
  " This leaf specifies the negotiated interval, in
  microseconds, that the local system is transmitting
  BFD Echo packets. The value is expected to be zero if
  the sessions are not running in Echo mode ";
  type uint32;
}
leaf bfdSess-Source-UdpPort {
  description
  " This leaf specifies the source UDP port number used
  for this BFD session’s Control packets. The value may be
  0 if the session is in adminDown state. Upon
  creation of a new BFD session via this, the value of
  0 specified would permit the implementation to
  choose its own source port number ";
}
leaf bfdSess-echoSrc-Udp {
  type uint32;
  description
  " This leaf specifies the source UDP port number used for
  this BFD session’s Echo packets. The value may be 0
  if the session is not running in the Echo mode, or the
  session is in adminDown state. Upon creation of a new
  BFD session via this, the value of 0 would
  permit the implementation to choose its own source port
  number ";
}
leaf bfdSess-Destination-UdpPort {
  type uint32;
  description
  " This leaf specifies the destination UDP port number
  used for this BFD session’s Control packets.
  3784 represents single-hop BFD session.
  4784 represents multi-hop BFD session.
  6784 represents BFD on Link Aggregation Group (LAG) session.
  The value may be zero(0) if the session is in adminDown(1) state";
{uses BFD-session;}

### 4.2. BFD-session-stats grouping

In BFD session, there are some useful parameters be used to indicate bfd session stats. This document define a bfd-session-status grouping which include these parameters. Using this bfd-session-status grouping, we can augment the basic gen-oam model as follows:

grouping bfd-session-stats{
  leaf bfdsess-status {
    type uint32;
    description
      " Configured BFD session state ";
  }
  leaf bfd-Sess-Diag{
    type enumeration{
      enum noDiagnostic{value 0;}
      enum controlDetectionTimeExpired{value 1;}
      enum echoFunctionFailed{value 2;}
      enum neighborSignaledSessionDown{value 3;}
      enum forwardingPlaneReset{value 4;}
      enum pathDown{value 5;}
      enum concatenatedPathDown{value 6;}
      enum administrativelyDown{value 7;}
      enum reverseConcatenatedPathDown{value 8;}
      enum misConnectivityDefect{value 9;}
    }
    description
      "A diagnostic code specifying the local system’s reason
      for the last transition of the session from up
      to some other state."
  }
  leaf bfdsess-admin-stats {
    type enumeration{
      enum enabled{value 1;}
      enum disabled{value 2;}
      enum adminDown{value 3;}
      enum down{value 4;}
    }
    description
      " Denotes the desired operational status of the BFD session.
      A transition to enabled will start the BFD state machine
      for the session. The state machine will have an initial
A transition to disabled will stop the BFD state machine for the session. The state machine may first transition to adminDown prior to stopping.
A transition to adminDown will cause the BFD state machine to transition to adminDown and will cause the session to remain in this state.
A transition to down will cause the BFD state machine to transition to down and will cause the session to remain in this state;
}
leaf bfdsess-oper-status {
  type enumeration{
    enum up{value 1;}
    enum down{value 2;}
    enum adminDown{value 3;}
  }
}
description
" Denotes the actual operational status of the BFD session. If the value of bfdOperStatus is down, this value MUST eventually be down as well. If the value of bfdOperStatus is adminDown, this value MUST eventually be adminDown as well ";
}
leaf bfdsess-remote-heard {
  type boolean;

description
" This leaf specifies the status of BFD packet reception from the remote system. Specifically, it is set to true if the local system is actively receiving BFD packets from the remote system and is set to false if the local system has not received BFD packets recently (within the detection time) or if the local system is attempting to tear down the BFD session ";
}
}else BFD session stats;

4.3. Identity technology-sub-type

In BFD, different encapsulation types such as MPLS [RFC5884] OSPFv2 [RFC5882], OSPFv3 [RFC5882], IS-IS [RFC5882], BGP [RFC5882], RIP [RFC5882] can be employed. "technology-sub-type" identifies the encapsulation type within the BFD. Technology sub-type is defined as an identity type. This allows different encapsulation types to augment BFD OAM YANG model to include encapsulation type specific extensions without redefining common BFD definitions.
  identity technology-sub-type{
    description
    "this is the base identity of sub-technology. It can be used to identify the encapsulation type such as MPLS, OSPFv2, OSPFv3, IS-IS, BGP, RIP, etc";
  }
}

4.4. connectivity-context

In BFD, connectivity-context is a pair of 32 bit Discriminators (SessDiscriminator, remotSessDiscriminator). This allows other technologies to easily augment that to include technology specific extensions. The snippet below depicts an example of augmenting connectivity-context to include Discriminator.
augment
/goam:session/goam:connectivity-context"
{
  case context-bfdsess {
    leaf bfdsess-discriminator {
      type uint32;
      description
      "This leaf specifies the local discriminator for this BFD
      session, which is used to uniquely identify it ";
    }
    leaf bfdsess-remotediscr {
      type uint32;
      description
      "This leaf specifies the session discriminator chosen
      by the remote system for this BFD session. The value may
      be zero(0) if the remote discriminator is not yet known
      or if the session is in the down or adminDown(1) state ";
    }
    leaf Interface{
      type leafref{
        path
        +"/goam:Interface";
      }
    }
    leaf source-mep{
      type leafref{
        path
        +"/goam:session/goam:source-mep";
      }
    }
    leaf destination-mep{
      type leafref{
        path
        +"/goam:session/goam:destination-mep"
        +"/goam:destination-mep/goam:MEP-ID/goam:MEP-ID-int"
        +"/goam:MEP-ID-int";
      }
    }
  }
}
4.5.  rpc definitions

The rpc model facilitates issuing commands to a NETCONF server (in this case to the device that need to execute the OAM command) and obtaining a response.

CC extension:

augment "/goam:cc/goam:output/goam: maintenance-stats" {
  case session-ccperform-bfd {
    leaf bfdSess-PerfCtrlPkt-In {
      type uint32;
      description
      "The total number of BFD control messages received for this BFD session";
    }
    leaf bfdSess-PerfCtrlPkt-Out {
      type uint32;
      description
      "The total number of BFD control messages sent for this BFD session";
    }
    leaf bfdSess-PerfCtrlPkt-Drop {
      type uint32;
      description
      "The total number of BFD control messages received for this session yet dropped for being invalid";
    }
    leaf bfdSess-PerfCtrlPktDrop-LastTime {
      type uint32;
      description
      "The value of sysUpTime on the most recent occasion at which received the BFD control message for this session was dropped. If no such up event exists, this object contains a zero value";
    }
    leaf bfdSess-PerfEchoPkt-In {
      type uint32;
      description
      "The total number of BFD Echo messages received for this BFD session";
    }
    leaf bfdSess-PerfEchoPkt-Out {
      type uint32;
      description
      "The total number of BFD Echo messages sent for this BFD session";
    }
  }
}
leaf bfdSess-PerfEchoPkt-Drop {
    type uint32;
    description "The total number of BFD Echo messages received for this session yet dropped for being invalid";
}

leaf bfdSess-PerfEchoPktDrop-LastTime {
    type uint32;
    description "The value of sysUpTime on the most recent occasion at which received the BFD Echo message for this session was dropped. If no such up event has been issued, this object contains a zero value";
}

leaf bfdSess-UpTime {
    type uint32;
    description "The value of sysUpTime on the most recent occasion at which the session came up. If no such event has been issued, this object contains a zero value";
}

leaf bfdSess-PerfLastSessDownTime {
    type uint32;
    description "The value of sysUpTime on the most recent occasion at which the last time communication was lost with the neighbor. If no down event has been issued, this object contains a zero value";
}

leaf bfdSess-PerfLastCommLostDiag {
    type uint32;
    description "The BFD diag code for the last time communication was lost with the neighbor. If such an event has not been issued, this object contains a zero value";
}

leaf bfdSess-PerfSess-UpCount {
    type uint32;
    description "The number of times this session has gone into the Up state since the system last rebooted";
}

leaf bfdSess-Perf-DiscTime {
    type uint32;
    description "The value of sysUpTime on the most recent occasion at which any one or more of the session counters suffered a discontinuity";
CV extension:

augment "/goam:cv/goam:output/goam: maintenance-stats" {  
  case session-cvperform-bfdctr {  
    leaf bfdSess-PerfCtrlPkt-In {  
      type uint32;  
      description  
        "The total number of BFD control messages received for this 
        BFD session";  
    }  
    leaf bfdSess-PerfCtrlPkt-Out {  
      type uint32;  
      description  
        "The total number of BFD control messages sent for this BFD 
        session";  
    }  
    leaf bfdSess-PerfCtrlPkt-Drop {  
      type uint32;  
      description  
        "The total number of BFD control messages received for this 
        session yet dropped for being invalid";  
    }  
    leaf bfdSess-PerfCtrlPktDrop-LastTime {  
      type uint32;  
      description  
        "The value of sysUpTime on the most recent occasion at 
        which received the BFD control message for this session was 
        dropped. If no such up event exists, this object contains 
        a zero value";  
    }  
    leaf bfdSess-UpTime {  
      type uint32;  
      description  
        "The value of sysUpTime on the most recent occasion at which 
        the session came up. If no such event has been issued, 
        this object contains a zero value";  
    }  
    leaf bfdSess-PerfLastSessDownTime {  
      type uint32;  
      description  
        "The value of sysUpTime on the most recent occasion at 
        which the last time communication was lost with the 
        neighbor. If no down event has been issued, this object 
        contains a zero value";  
  }  
}
4.6. Notification definitions

In BFD, the bfdSess-admin-status, bfdSess-oper-status, bfdSess-status and the bfdSess-remote-heard are used to indicate the session status. i.e. the bfdSess-admin-status denotes the desired operational status of the BFD session. And the bfdSess-oper-status denotes the actual operational status of the BFD session. The bfdSess-status is used to indicate the BFD session state. And the bfdSess-oper-status specifies the status of BFD packet reception from the remote system.

5. OAM Data Hierarchy

The complete data hierarchy related to the OAM YANG model is presented below.


    +--rw technology-sub-type           identityref
    +--rw bfdSess-version-num?           uint32
++--rw bfdsess-type?          uint32
++--rw bfdsess-oper-mode?     uint32
++--rw bfdsess-demandmode-desired? boolean
++--rw bfdsess-controlplane-indep? boolean
++--rw bfdsess-multi-point?    boolean
++--rw bfdsess-detect-mult?   uint32
++--rw bfdsess-authpres?      boolean
++--rw bfdsess-authentication-type uint32
++--rw bfdsess-authentication-key? uint32
++--rw bfdSess-DesiredMinTx-Interval uint32
++--rw bfdSess-ReqMinRx-Interval? uint32
++--ro bfdSess-Negotiated-Interval? uint32
++--ro bfdSess-NegotiatedEcho-Interval? uint32
++--rw bfdSess-Source-UdpPort? uint32
++--rw bfdSess-echoSrc-UdpPort? uint32
++--rw bfdSess-Destination-UdpPort uint32
++--ro bfdssaess-status?       uint32
++--ro bfdssess-bfd-Sess-Diag? enumeration
++--rw bfdssess-admin-status?   enumeration
++--ro bfdssess-oper-status?   enumeration
++--ro bfdssess-remote-heard?  boolean

augment
++--:(context-bfdsess)
++--rw bfdssess-discriminator? uint32
++--rw bfdssess-remotediscr?    uint32
++--rw Interface?               leafref
++--rw source-mep?              leafref
++--rw destination-mep?         leafref

augment
/goam:cc/goam:output/goam:maintenance-stats:
++--:(session-ccperform-bfd)
++--ro bfdSess-PerfCtrlPkt-In?  uint32
++--ro bfdSess-PerfCtrlPkt-Out? uint32
++--ro bfdSess-PerfCtrlPkt-Drop? uint32
++--ro bfdSess-PerfCtrlPktDrop-LastTime? uint32
++--ro bfdSess-PerfEchoPkt-In?  uint32
++--ro bfdSess-PerfEchoPkt-Out? uint32
++--ro bfdSess-PerfEchoPkt-Drop? uint32
++--ro bfdSess-PerfEchoPktDrop-LastTime? uint32
++--ro bfdSess-UpTime?          uint32
++--ro bfdSess-PerfLastSessDownTime? uint32
++--ro bfdSess-PerfLastCommLostDiag? uint32
++--ro bfdSess-PerfSess-UpCount? uint32
++--ro bfdSess-Perf-DiscTime?   uint32
6. OAM YANG Module

<CODE BEGINS> file "xxx.yang"
<CODE BEGINS> file "xxx.yang"

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

    import gen-oam {
        prefix goam;
    }
    import ietf-inet-types {
        prefix inet;
    }
    import ietf-interfaces {
        prefix if;
    }
    import ietf-yang-types {
        prefix yang;
    }

    revision 2014-08-30 {
        description
        }
"Initial revision."
}

identity bfd {
    base bfd:technology-types;
    description
        "bfd type";
}

identity bfd-continuity-check {
    base goam:command-sub-type;
}

identity bfd-connectivity-verification {
    base goam:command-sub-type;
}

grouping bfd-session{
    leaf bfdsess-version-num{
        type uint32;
        default "1";
        description
            " The version number of the BFD protocol that this session
            is running in. Write access is available for this object
            to provide the ability to set the desired version for this
            BFD session ";
    }

    leaf bfdsess-type{
        type enumeration{
            enum singleHop{value 1;}
            enum multiHopTotallyArbitraryPaths{value 2;}
            enum multiHopOutOfBandSignaling{value 3;}
            enum multiHopUnidirectionalLinks{value 4;}
        }
        description
            " This leaf specifies the type of this BFD session ";
    }

    leaf bfdsess-oper-mode {
        type enumeration{
            enum asynchModeWEchoFunction{value 1;}
            enum asynchModeWOEchoFunction{value 2;}
            enum demandModeWEchoFunction{value 3;}
            enum demandModeWOEchoFunction{value 4;}
        }
        description
            " This object specifies the operational mode of this
            BFD session ";
    }
}

leaf bfdsess-demandmode-desired {
  type boolean;
  description
    "This leaf indicates the local system’s desire to use Demand mode. Specifically, it is set to true if the local system wishes to use Demand mode or false if not";
}

leaf bfdsess-controlplanel-indep {
  type boolean;
  description
    "This leaf indicates the local system’s ability to continue to function through a disruption of the control plane. Specifically, it is set to true if the local system BFD implementation is independent of the control plane. Otherwise, the value is set to false";
}

leaf bfdsess-multi-point {
  type boolean;
  description
    "This leaf indicates the Multipoint (M) bit for this session. It is set to true if the Multipoint (M) bit is set to 1. Otherwise, the value is set to false";
}

leaf bfdsess-detect-mult {
  type uint32;
  description
    "This leaf specifies the Detect time multiplier";
}

leaf bfdsess-authpres {
  type boolean;
  description
    "This leaf indicates the local system’s desire to use authentication. Specifically, it is set to true if the local system wishes the session to be authenticated or false if not";
}

leaf bfdsess-authentication-type {
  type enumeration{
    enum noAuthentication;
    enum reserved;
  }
enum simplePassword;
enum keyedMD5;
enum meticulousKeyedMD5;
enum keyedSHA1;
enum meticulousKeyedSHA1;
}
description
"The authentication type used for this BFD session. This field is valid only when the Authentication Present bit is set."

leaf bfdSess-authentication-key {
  type uint32;
  description
  "This leaf specifies the authentication key in use for this session."
}

leaf bfdSess-authentication-keyID{
  type uint32;
  description
  "The authentication key ID in use for this session. This leaf permits multiple keys to be active simultaneously."
}

leaf bfdSess-DesiredMinTx-Interval{
  description
  "This leaf specifies the minimum interval, in microseconds, that the local system would like to use when transmitting BFD Control packets. The value of zero(0) is reserved in this case and should not be used."
  type uint32;
}

leaf bfdSess-ReqMinRx-Interval {
  description
  "This leaf specifies the minimum interval, in microseconds, between received BFD Control packets the local system is capable of supporting. The value of zero(0) can be specified when the transmitting system does not want the remote system to send any periodic BFD Control packets."
  type uint32;
}

leaf bfdSess-ReqMinEchoRx-Interval {
  description
  "This leaf specifies the minimum interval, in microseconds, between received BFD Echo packets that this system is capable of supporting. The value must be zero(0) if
this is a multihop BFD session
  type uint32;
}
leaf bfdSess-Negotiated-Interval {
  description
  " This leaf specifies the negotiated interval, in
  microseconds, that the local system is transmitting
  BFD Control packets ";
  type uint32;
}
leaf bfdSess-NegotiatedEcho-Interval {
  description
  " This leaf specifies the negotiated interval, in
  microseconds, that the local system is transmitting
  BFD Echo packets. The value is expected to be zero if
  the sessions are not running in Echo mode ";
  type uint32;
}
leaf bfdSess-Source-UdpPort {
  description
  " This leaf specifies the source UDP port number used
  for this BFD session’s Control packets. The value may be
  0 if the session is in adminDown state. Upon
  creation of a new BFD session via this, the value of
  0 specified would permit the implementation to
  choose its own source port number ";
}
leaf bfdSess-echoSrc-Udp{
  type uint32;
  description
  " This leaf specifies the source UDP port number used for
  this BFD session’s Echo packets. The value may be 0
  if the session is not running in the Echo mode, or the
  session is in adminDown state. Upon creation of a new
  BFD session via this, the value of 0 would
  permit the implementation to choose its own source port
  number ";
}
leaf bfdSess-Destination-UdpPort {
  type uint32;
  description
  " This leaf specifies the destination UDP port number
  used for this BFD session’s Control packets.
  3784 represents single-hop BFD session.
  4784 represents multi-hop BFD session.
  6784 represents BFD on Link Aggregation Group (LAG) session.
  The value may be zero(0) if the session is in adminDown(1) state";
grouping bfd-session-stats{
    leaf bfdsess-status {
        type uint32;
        description
            " Configured BFD session state ";
    }
    leaf bfd-Sess-Diag{
        type enumeration{
            enum noDiagnostic{value 0;}
            enum controlDetectionTimeExpired{value 1;}
            enum echoFunctionFailed{value 2;}
            enum neighborSignaledSessionDown{value 3;}
            enum forwardingPlaneReset{value 4;}
            enum pathDown{value 5;}
            enum concatenatedPathDown{value 6;}
            enum administrativelyDown{value 7;}
            enum reverseConcatenatedPathDown{value 8;}
            enum misConnectivityDefect{value 9;}
        }
        description
            "A diagnostic code specifying the local system’s reason
            for the last transition of the session from up
to some other state."
    }
    leaf bfdsess-admin-stats {
        type enumeration{
            enum enabled{value 1;}
            enum disabled{value 2;}
            enum adminDown{value 3;}
            enum down{value 4;}
        }
        description
            " Denotes the desired operational status of the BFD session.
            A transition to enabled will start the BFD state machine
            for the session. The state machine will have an initial
            state of down.
            A transition to disabled will stop the BFD state machine
            for the session. The state machine may first transition to
            adminDown prior to stopping.
            A transition to adminDown will cause the BFD state
            machine to transition to adminDown and will cause the
            session to remain in this state.
            A transition to down will cause the BFD state machine
            to transition to down and will cause the session to
remain in this state; }
leaf bfdsess-oper-status {
  type enumeration{
    enum up{value 1;}
    enum down{value 2;}
    enum adminDown{value 3;}
  };
  description
  " Denotes the actual operational status of the BFD session.
  If the value of bfdOperStatus is down, this value MUST eventually be down as well. If the value of bfdOperStatus is adminDown, this value MUST eventually be adminDown as well ";
}
leaf bfdsess-remote-heard {
  type boolean;
  description
  " This leaf specifies the status of BFD packet reception from the remote system. Specifically, it is set to true if the local system is actively receiving BFD packets from the remote system and is set to false if the local system has not received BFD packets recently (within the detection time) or if the local system is attempting to tear down the BFD session ";
}

augment "/goam:domains/goam:domain/goam:MAsgoam:MA/goam:MEP/goam:session" { identity technology-sub-type{
  description
  "this is the base identity of sub-technology. It can be used to identify the encapsulation type such as MPLS, OSPFv2, OSPFv3, IS-IS, BGP, RIP, etc";
}
}


  leaf bfdsess-discriminator {
}
type uint32;
description
  "This leaf specifies the local discriminator for this BFD
  session, which is used to uniquely identify it ";
}
leaf bfdsess-remotediscr {
  type uint32;
description
  "This leaf specifies the session discriminator chosen
  by the remote system for this BFD session. The value may
  be zero(0) if the remote discriminator is not yet known
  or if the session is in the down or adminDown(1) state ";
}
leaf Interface{
  type leafref{
    path
    +"/goam:Interface";
  }
}
leaf source-mep{
  type leafref{
    path
    +"/goam:session/goam:source-mep";
  }
}
leaf destination-mep{
  type leafref{
    path
    +"/goam:session/goam:destination-mep"
    +"/goam:destination-mep/goam:MEP-ID/goam:MEP-ID-int"
    +"/goam:MEP-ID-int";
  }
}

augment "/moam:cc/moam:output/moam: maintenance-stats " {
case session-ccperform-bfd {
  leaf bfdSess-PerfCtrlPkt-In {
    type uint32;
description
    " The total number of BFD control messages received for this
    BFD session";
  }
  leaf bfdSess-PerfCtrlPkt-Out {

leaf bfdSess-PerfCtrlPkt-Drop {
  type uint32;
  description
  " The total number of BFD control messages received for this BFD
  session Yet dropped for being invalid ";
}

leaf bfdSess-PerfCtrlPktDrop-LastTime {
  type uint32;
  description
  " The value of sysUpTime on the most recent occasion at
  which received the BFD control message for this session was
  dropped. If no such up event exists, this object contains
  a zero value ";
}

leaf bfdSess-PerfEchoPkt-In {
  type uint32;
  description
  " The total number of BFD Echo messages received for this
  BFD session ";
}

leaf bfdSess-PerfEchoPkt-Out {
  type uint32;
  description
  " The total number of BFD Echo messages sent for this BFD
  session ";
}

leaf bfdSess-PerfEchoPkt-Drop {
  type uint32;
  description
  " The total number of BFD Echo messages received for this
  session yet dropped for being invalid ";
}

leaf bfdSess-PerfEchoPktDrop-LastTime {
  type uint32;
  description
  " The value of sysUpTime on the most recent occasion at
  which received the BFD Echo message for this session was
  dropped. If no such up event has been issued, this
  object contains a zero value ";
}

leaf bfdSess-UpTime {
  type uint32;
  description
  " The total number of BFD control messages sent for this BFD
  session";
}
"The value of sysUpTime on the most recent occasion at which
the session came up. If no such event has been issued,
this object contains a zero value";
}
leaf bfdSess-PerfLastSessDownTime {
    type uint32;
    description
    " The value of sysUpTime on the most recent occasion at
which the last time communication was lost with the
neighbor. If no down event has been issued, this object
contains a zero value ";
}
leaf bfdSess-PerfLastCommLostDiag {
    type uint32;
    description
    " The BFD diag code for the last time communication was lost
with the neighbor. If such an event has not been issued,
this object contains a zero value";
}
leaf bfdSess-PerfSess-UpCount {
    type uint32;
    description
    " The number of times this session has gone into the Up
state since the system last rebooted ";
}
leaf bfdSess-Perf-DiscTime {
    type uint32;
    description
    " The value of sysUpTime on the most recent occasion at
which any one or more of the session counters suffered
a discontinuity ";
}
}

augment "/moam: cv /moam:output/moam: maintenance-stats " {
    case session-cvperform-bfdctr {
        leaf bfdSess-PerfCtrlPkt-In {
            type uint32;
            description
            " The total number of BFD control messages received for this
BFD session";
        }
        leaf bfdSess-PerfCtrlPkt-Out {
            type uint32;
            description
            " The total number of BFD control messages sent for this BFD
leaf bfdSess-PerfCtrlPkt-Drop {
    type uint32;
    description "The total number of BFD control messages received for this session yet dropped for being invalid ";
}
leaf bfdSess-PerfCtrlPktDrop-LastTime {
    type uint32;
    description "The value of sysUpTime on the most recent occasion at which received the BFD control message for this session was dropped. If no such up event exists, this object contains a zero value ";
}
leaf bfdSess-UpTime {
    type uint32;
    description "The value of sysUpTime on the most recent occasion at which the session came up. If no such event has been issued, this object contains a zero value";
}
leaf bfdSess-PerfLastSessDownTime {
    type uint32;
    description "The value of sysUpTime on the most recent occasion at which the last time communication was lost with the neighbor. If no down event has been issued, this object contains a zero value ";
}
leaf bfdSess-PerfLastCommLostDiag {
    type uint32;
    description "The BFD diag code for the last time communication was lost with the neighbor. If such an event has not been issued, this object contains a zero value";
}
leaf bfdSess-PerfSess-UpCount {
    type uint32;
    description "The number of times this session has gone into the Up state since the system last rebooted ";
}
leaf bfdSess-Perf-DiscTime {
    type uint32;
    description
" The value of sysUpTime on the most recent occasion at
which any one or more of the session counters suffered
a discontinuity ";
} }
}

augment "/moam:notifications/moam:defect-condition-notification
" { uses bfd-session-stats; }

7. Default Mode for BFD OAM

This document defines default configuration that MUST be present in
the devices that comply with this document. Base Mode allows users
to have zero-touch experience.

8. Security Considerations

TBD.

9. IANA Considerations

TBD.

10. Normative References

[I-D.tissa-lime-yang-oam-model]
Senevirathne, T., Finn, N., Kumar, D., Salam, S., and Q.
Wu, "Generic YANG Data Model for Operations,
Administration, and Maintenance (OAM)", draft-tissa-lime-
yang-oam-model-03 (work in progress), November 2014.

[I-D.wang-lime-rpc-yang-oam-management]
Wang, Z. and Q. Wu, "Additional RPC definitions to Generic
YANG Data Model for layer Independent OAM Management",
draft-wang-lime-rpc-yang-oam-management-00 (work in
progress), October 2014.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate

(BFD)", RFC 5880, June 2010.


Authors’ Addresses

Zitao Wang
Huawei Technologies,Co.,Ltd
101 Software Avenue, Yuhua District
Nanjing  210012
China
Email: wangzitao@huawei.com

Liang Xia
Huawei Technologies,Co.,Ltd
101 Software Avenue, Yuhua District
Nanjing  210012
China
Email: frank.xialiang@huawei.com

Deepak Kumar
Cisco Systems
510 McCarthy Blvd Milpitas,
CA   95035
USA
Email: dekumar@cisco.com

Qin Wu
Huawei
101 Software Avenue, Yuhua District
Nanjing, Jiangsu  210012
China
Email: bill.wu@huawei.com