Abstract

This document proposes a mechanism to run BFD on Link Aggregation Group (LAG) interfaces. It does so by running an independent Asynchronous mode BFD session on every LAG member link.

This mechanism allows the verification of member link continuity, either in combination with, or in absence of, LACP. It provides a shorter detection time than what LACP offers. The continuity check can also cover elements of layer 3 bidirectional forwarding.

This mechanism utilizes a well-known UDP port distinct from that of single-hop BFD over IP. This new UDP port removes the ambiguity of BFD over LAG packets from BFD over single-hop IP.

Requirements Language

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

Status of this Memo

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

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

Internet-Drafts are draft documents valid for a maximum of six months.
Table of Contents

1. Introduction .................................................. 4
2. BFD on LAG member links ....................................... 4
   2.1. Micro BFD session address family ....................... 5
   2.2. Micro BFD session negotiation .......................... 5
   2.3. Micro BFD session Ethernet details ..................... 6
3. LAG Management Module ......................................... 6
   3.1. Interaction between LAG and BFD ....................... 6
   3.2. Handling Exceptions ..................................... 7
4. BFD on LAG members and layer-3 applications .................. 8
5. Detecting a member link failure ................................. 8
6. Security Consideration .......................................  8
7. IANA Considerations .........................................  8
8. Acknowledgements ...........................................  9
9. Contributing authors .........................................  9
10. References .................................................. 10
   10.1. Normative References .................................. 10
   10.2. Informative References ................................. 10
Authors’ Addresses ............................................... 10
1. Introduction

The Bidirectional Forwarding Detection (BFD) protocol [RFC5880] provides a mechanism to detect faults in the bidirectional path between two forwarding engines, including interfaces, data link(s), and to the extent possible the forwarding engines themselves, with potentially very low latency. The BFD protocol also provides a fast mechanism for detecting communication failures on any data links and the protocol can run over any media and at any protocol layer.

Link aggregation (LAG) as defined in [IEEE802.1AX] provides mechanisms to combine multiple physical links into a single logical link. This logical link provides higher bandwidth and better resiliency since if one of the physical member links fails the aggregate logical link can continue to forward traffic over the remaining operational physical member links.

Currently, the Link Aggregation Control Protocol (LACP) is used to detect failures on a per physical member link. However, the use of BFD for failure detection would (1) provide a faster detection (2) provide detection in the absence of LACP (3) and would be able to verify L3 Continuity per member link.

Running a single BFD session over the aggregation without internal knowledge of the member links would make it impossible for BFD to guarantee detection of the physical member link failures.

The goal is to verify link Continuity for every member link. This corresponds to [RFC5882], section 7.3.

The approach taken in this document is to run a Asynchronous mode BFD session over each member link and make BFD control whether the member link should be part of the L2 Loadbalance table of the LAG virtual port in the presence or the absence of LACP.

This document describes how to establish an Asynchronous mode BFD session per physical member link of the LAG virtual port.

While there are native Ethernet mechanisms to detect failures (802.1ax, .3ah) that could be used for LAG, the solution proposed in this document enables operators who have already deployed BFD over different technologies (e.g. IP, MPLS) to use a common failure detection mechanism.

2. BFD on LAG member links

The mechanism proposed for a fast detection of LAG member link
failure is to run Asynchronous mode BFD sessions on every LAG member
link. We call these per LAG member link BFD sessions "micro BFD
sessions" in the remainder of this document.

2.1. Micro BFD session address family

Member link micro BFD sessions, when using IP/UDP encapsulation, can
use IPv4 or IPv6 addresses. Two micro sessions MAY exist per member
link, one IPv4, another IPv6. When an address family is used on one
member link then it MUST be used on all member links of the
particular LAG.

2.2. Micro BFD session negotiation

A single micro BFD session for every enabled address family runs on
each member link of the LAG. The micro BFD session’s negotiation
MUST follow the same procedures defined in [RFC5880] and [RFC5881].

Only Asynchronous mode BFD is considered in this document; the use of
the BFD echo function is outside the scope of this document. At
least one system MUST take the Active role (possibly both). The
micro BFD sessions on the member links are independent BFD sessions:
They use their own unique local discriminator values, maintain their
own set of state variables and have their own independent state
machines. Timer values MAY be different, even among the micro BFD
sessions belonging to the same aggregation, although it is expected
that micro BFD sessions belonging to the same aggregation will use
the same timer values.

The demultiplexing of a received BFD packet is solely based on the
Your Discriminator field, if this field is nonzero. For the initial
Down BFD packets of a BFD session this value MAY be zero. In this
case demultiplexing MUST be based on some combination of other fields
which MUST include the interface information of the member link.

The procedure for the Reception of BFD Control Packets in Section
6.8.6 of [RFC5880] is amended as follows for per member link micro
BFD over LAG sessions: "If the Your Discriminator field is non-zero
and a micro BFD over LAG session is found, the interface on which the
micro BFD control packet arrived on MUST correspond to the interface
associated with that session."

This document defines the BFD Control packets for each micro BFD
session to be IP/UDP encapsulated as defined in [RFC5881], but with a
new UDP destination port 6784.

Control packets use a destination IP address that is configured on
the peer system and can be reached via the LAG interface. The
details of how this destination IP address is learned are outside the scope of this document.

2.3. Micro BFD session Ethernet details

On Ethernet-based LAG member links the destination MAC is the dedicated multicast MAC address 01-00-5E-90-00-01 to be the immediate next hop. This dedicated MAC address MUST be used for the initial BFD packets of a micro BFD session when in the Down/AdminDown and Init state. When a micro BFD session is changing into Up state then the first bfd.DetectMult packets with Up state MUST be sent with the dedicated MAC. For the following BFD packets with Up state the MAC address from the received BFD packets for the session MAY be used instead of the dedicated MAC.

All implementations MUST be able to send and receive BFD packets in Up state using the dedicated MAC address. Implementations supporting both, sending BFD Up packets with the dedicated and the received MAC need to offer means to control the behaviour.

On Ethernet-based LAG member links the source MAC SHOULD be the MAC address of the port transmitting the packet.

This mechanism helps to reduce the use of additional MAC addresses, which reduces the required resources on the Ethernet hardware on the receiving port.

Micro BFD packets SHOULD always be sent untagged. However, when the LAG is operating in the context of IEEE 802.1q or IEEE 802.qinq, the micro BFD packets may either be untagged or sent with a vlan tag of Zero (802.1p priority tagged). Implementations compliant to this standard MUST be able to receive both untagged and 802.1p priority tagged micro BFD packets.

3. LAG Management Module

3.1. Interaction between LAG and BFD

The LAG Management Module (LMM) could be envisaged as a client of BFD; i.e. the LMM requests the micro BFD sessions per member link. The LMM then uses the micro BFD session state, in addition to LACP state, to monitor the health of the individual members links of the LAG.

The micro BFD sessions for a particular port MUST be requested when a member port state is either Distributing or Standby. The sessions MUST be deleted when the member port is neither in Distributing nor
in Standby state anymore.

BFD is used to control if the load balance algorithm is able to select a particular port. In other words, even when LACP is used and considers the member link to be ready to forward traffic, the member link is only used by the load balancer when all the micro BFD sessions of the member link are Up.

In case an implementation has separate load balance tables for IPv4 and IPv6 then if both an IPv4 and IPv6 micro session exist for a member link an implementation MAY enable the member link in the distribution algorithm only when the BFD session with a matching address family is changing into Up state.

An exception are the BFD packets itself. Implementations MAY receive and transmit BFD packets via the Aggregator’s MAC service interface independent of the session state.

3.2. Handling Exceptions

If the BFD over LAG feature were provisioned on an aggregated link member after the link was already active within a LAG, BFD session state SHOULDN'T influence the load balance algorithm until the BFD session state transitions to Up. If the BFD session never transitions to Up but the LAG becomes inactive, the previously documented procedures would then normally apply.

If the BFD over LAG feature were deprovisioned on an aggregate link member after the BFD session had transitioned to Up, BFD MAY indicate to the remote port that it should not take the port down or remove it from the aggregation by setting its BFD session state to AdminDown.

When a micro BFD session receives AdminDown from the peer, it is RECOMMENDED to have a configurable timeout value. If the BFD session has not been removed within the timeout period the link is taken out of forwarding.

When traffic is forwarded across a link before the corresponding micro BFD session is Up it is RECOMMENDED to have a configurable timeout value after which the BFD session must have reached Up state or otherwise the link is taken out of forwarding.

Note that if one device is not operating a micro BFD session on a link, while the other device is and perceives the session to be Down, this will result in the two devices having a different view of the status of the link. This would likely lead to traffic loss across the LAG.
The use of another protocol to bootstrap BFD can detect such mismatched config, since the side that’s not configured can send a rejection error. Such bootstrapping mechanisms are outside the scope of this document.

4. BFD on LAG members and layer-3 applications

The mechanism described in this document is likely to be used by modules like LMM or some Interface management module. Typical layer 3 protocols like OSPF do not have an insight into the LAG and treat it as one bigger interface. The signalling from micro sessions to layer 3 protocols is effectively done by the impact of BFD micro sessions on the load balance table and the LMM’s potential decision to shut down the LAG. An active method to test the impact of micro sessions is for layer 3 protocols to request a single BFD session per LAG.

5. Detecting a member link failure

When a micro BFD session goes down then this member link MUST be taken out of the LAG L2 load balance table(s).

In case an implementation has separate load balance tables for IPv4 and IPv6 then if both an IPv4 and IPv6 micro session exist for a member link an implementation MAY remove the member link from the load balance table only that matches the address family of the failing BFD session. If for example the IPv4 micro session fails but the IPv6 micro session stays up then the member link MAY be removed from the IPv4 load balance table only but remains forwarding in the IPv6 load balance table.

6. Security Consideration

This document does not introduce any additional security issues and the security mechanisms defined in [RFC5880] apply in this document.

7. IANA Considerations

IANA assigned a dedicated MAC address 01-00-5E-90-00-01 as well as UDP port 6784 for UDP encapsulated micro BFD sessions.
8. Acknowledgements

We would like to thank Dave Katz, Alexander Vainshtein, Greg Mirsky and Jeff Tantsura for their comments.

The initial event to start the current discussion was the distribution of draft-chen-bfd-interface-00.

9. Contributing authors

Paul Hitchen
BT
Email: paul.hitchen@bt.com

George Swallow
Cisco Systems
Email: swallow@cisco.com

Wim Henderickx
Alcatel-Lucent
Email: wim.henderickx@alcatel-lucent.com

Nobo Akiya
Cisco Systems
Email: nobo@cisco.com

Neil Ketley
Cisco Systems
Email: nketley@cisco.com

Carlos Pignataro
Cisco Systems
Email: cpignata@cisco.com

Nitin Bahadur
Juniper Networks
Email: nitinb@juniper.net

Zuliang Wang
Huawei Technologies
Email: liang_tsing@huawei.com

Liang Guo
China Telecom
Email: guoliang@gsta.com

Jeff Tantsura
10. References

10.1. Normative References


10.2. Informative References


Authors’ Addresses

Manav Bhatia (editor)
Alcatel-Lucent
Bangalore 560045
India

Email: manav.bhatia@alcatel-lucent.com
Mach(Guoyi) Chen (editor)
Huawei Technologies
Q14 Huawei Campus, No. 156 Beiqing Road, Hai-dian District
Beijing  100095
China

Email: mach@huawei.com

Sami Boutros (editor)
Cisco Systems

Email: sboutros@cisco.com

Marc Binderberger (editor)
Cisco Systems

Email: mbinderb@cisco.com

Jeffrey Haas (editor)
Juniper Networks

Email: jhaas@juniper.net
BFD Support DS-Lite

draft-tsou-bfd-ds-lite-02

Abstract

In DS-Lite, the tunnel is not associated with any state information, which makes it difficult to manage and diagnose. Some tools may be used to resolve this problem.

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 24, 2012.

Copyright Notice

Copyright (c) 2012 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 include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.
1. Problem statement

In DS-Lite [RFC6333], the IPv4-in-IPv6 tunnel is stateless, no status information about tunnel is available, and no keep-alive mechanism is available. It is difficult to know whether the tunnel is up or down, which creates a problem for operation and maintenance.

If a B4 can detect a failure in the link to AFTR, it can switch to another AFTR, setup new tunnel to that AFTR, so as to continue the network service. Anycast could be used for the same purpose -- failover, but there is an ICMP error message problem, that is, when a packet is sent from AFTR to B4, one of the routers along the path may generate an error ICMP message, e.g., packet too big, and the error message is not sent back to the source AFTR, but sent to another AFTR.

In some cases, the operators may want to have some more diagnostic functions besides connectivity test, e.g. delay and throughput test, this may be useful if the operator is providing services like IPTV and video conference. ETHOAM and BFD can provide these functionalities. But ETHOAM[802.1ag - 2007] is for ethernet layer2; and BFD OAM functions now is only available for MPLS-TP[RFC6374]. This is currently out of the scope of this spec.

1.1. Requirements Language

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

2. Terminology

AFTR:  Address Family Transition Router.
B4:    Basic Bridging BroadBand.
BFD:   Bidirectional Forwarding Detection.
CPE:   Customer Premise Equipment (i.e., the DS-Lite B4).
FQDN  Fully Qualified Domain Name.
PCP    Port Control Protocol.
3. Solutions

3.1. BFD

BFD is a mechanism intended to detect faults in the bidirectional path. It is usually used in conjunction with applications like OSPF, IS-IS, etc, for fast fault recovery/fast re-route. [RFC5882]

BFD [RFC5880] can be used in DS-Lite, by creating a BFD session between the B4 and the AFTR to provide tunnel status information. If a fault is detected the B4 can try to create a DS-Lite tunnel with another AFTR and terminate the existing one, so as to continue network service.

[I-D.vinokour-bfd-dhcp] proposes using a DHCP option to distribute BFD parameters to the B4. But in case of DS-Lite, some of the key BFD parameters are already available (e.g., peer IP address is already available), and other parameters can be negotiated by BFD signaling or statically configured, so that no extra DHCP option(s) need to be defined.

3.1.1. DS-Lite Scenario

In DS-Lite [RFC6333], the BFD packet SHOULD be sent through an IPv4-in-IPv6 tunnel, as shown in Figure 1. The IPv4 addresses of the B4 and AFTR SHOULD be the endpoints of a BFD session.

```
+--------------+                  +--------------+
|      |-----+--------------+-----|      |     |              |
| CPE  |     IPv6 Tunnel        | AFTR |-----| IPv4 Network |
| (B4) |     +--------------+-----|      |     |              |
+------+     | IPv6 Network |     +------+     |              |
192.0.0.2     +--------------+    192.0.0.1     +--------------+
```

Figure 1: DS-Lite Scenario

3.1.2. Parameters for BFD

In order to set up a BFD session, the following parameters are needed, as shown in Section 4.1 of RFC5880:

- Peer IP address
- My Discriminator
- Your Discriminator

In DS-Lite [RFC6334], the B4 WAN-side IPv4 address is a well-known
address 192.0.0.2, and the AFTR’s IPv4 address is 192.0.0.1, as
defined in section 5.7 of [RFC6333]. Because all the B4s and AFTRs
use the same well-known IP addresses, IPv4 addresses are not
sufficient for setting up a BFD session. From the B4’s point of
view, the B4 needs to create an IPv6 tunnel to an AFTR so as to get
network connectivity to the AFTR, and send IPv4 BFD packets through
the tunnel to manage it.

The other parameters listed above can be negotiated by BFD signaling,
and initial values can be configured on the B4 and AFTR.

3.1.3. Procedures

In DS-Lite [RFC6333], when a B4 gets online, it will be assigned an
IPv6 prefix/address, and also the FQDN of the AFTR, as defined in
[RFC6334]. The B4 will create an IPv6 tunnel to the AFTR with which,
along with the well known B4 IPv4 address 192.0.0.2 and AFTR IPv4
address 192.0.0.1, the B4 can initiate a BFD session to the AFTR.
BFD packets will be sent through DS-Lite tunnel. As defined in
section 4 of [RFC5881], BFD control packet MUST be sent in UDP packet
with destination port 3784, and BFD echo packet MUST be sent in UDP
packet with destination port 3785.

When sending out the first BFD packet, the B4 can generate a unique
local discriminator, and set the remote discriminator to zero. When
the AFTR receive the first BFD packet from a B4, the AFTR will also
generate a corresponding local discriminator, and put it in the
response packet to the B4. This will finish the discriminator
negotiation in the B4 to AFTR direction, without any manual
configuration.

When the AFTR receives the first packet from a B4, AFTR will get the
IPv6 address and discriminator of the B4, so that the AFTR can
initiate the BFD session in the other direction and a similar
discriminator negotiation can be carried out.

3.1.4. Implementation Considerations

BFD is usually used for quick fault detection, at a very small time
scale, e.g. milliseconds. But in DS-Lite, it may not be necessary to
detect faults in such a short time. On the other hand, an AFTR may
need to support tens of thousands of B4s, which means the AFTR will need to support the same number of BFD sessions. In order to meet performance requirements on the AFTR, it may be necessary to extend the time period between BFD packet transmissions to a longer time, e.g., 10s or 30s.

3.2. PCP

PCP [I-D.ietf-pcp-base-23] is a NAT traversal tool, and it can also be used for network connectivity test if PCP is supported in the network. A common use case of PCP is to create pinhole so that external users can visit the servers located behind a NAT, and the lifetime of the pinhole mapping is usually long, e.g., hours, and the lifetime will be refreshed periodically by the client before it is expired. For the purpose of network connectivity test, B4 can create a mapping in the CGN via PCP, with short life time, e.g. 10s of seconds, and keep on refreshing the mapping before it is expired. If any refresh request fail, B4 knows that something is wrong with the link or the PCP server or CGN.

In order to detect the network connectivity of the DS-Lite tunnel, the encapsulation mode MUST be used for PCP -- PCP packets are sent through DS-Lite tunnel. Encapsulation mode and plain mode are two alternatives for PCP, there is no consensus yet which one should be preferred in the PCP spec.

3.3. PING

PING is a common tool used for network node reachability test, most of the network nodes provide this tool. In case of DS-Lite, B4 can send PING packets to AFTR periodically. If B4 does not receive response packets for a certain number of PING request packet, e.g. 3, then B4 decides that a fault is detected.

In order to test the connectivity of DS-Lite tunnel, PING packets MUST be sent using ICMPv4, rather than ICMPv6.

BFD can provide more diagnostic functions than PING, as depicted in section 4.1 of [RFC5880].

4. Failover

The FQDN of the AFTR is sent to B4 via a DHCP option, as defined in [RFC6334]. Multiple IP addresses can be configured for an FQDN on the DNS server. If B4 detect a failure on the link to AFTR, B4 MUST terminate the current DS-Lite tunnel, choose another AFTR address in the list, and create a tunnel to the new AFTR. B4 SHOULD also re-
configure the connectivity test tool accordingly if necessary, and restart the test procedures.

5. IANA Considerations

This memo includes no request to IANA.

6. Security Considerations

In DS-Lite [RFC6333], the B4 may not be directly connected to the AFTR; there may be other routers between them. Then there are potential spoofing problems, as described in [RFC5883]. Hence cryptographic authentication SHOULD be used as described in [RFC5880] if security is concerned.

7. Acknowledgements

The author would like to thank Mohamed Boucadair for his useful comments, more solutions are included in this memo.

8. References

8.1. Normative References

[I-D.ietf-pcp-base-23]


(BFD) for Multihop Paths", RFC 5883, June 2010.


8.2.  Informative References


Author’s Address

Tina Tsou (editor)
Huawei Technologies (USA)
2330 Central Expressway
Santa Clara CA 95050
USA

Phone: +1 408 330 4424
Email: tina.tsou.zouting@huawei.com
BFD Management Information Base (MIB) extensions
for MPLS and MPLS-TP Networks
draft-vkst-bfd-mpls-mib-02

Abstract

This draft defines a portion of the Management Information Base (MIB) for use with network management protocols in the Internet community. In particular, it extends the BFD Management Information Base BFD-STD-MIB and describes the managed objects for modeling Bidirectional Forwarding Detection (BFD) protocol for MPLS and MPLS-TP networks.

Status of this Memo

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

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt.

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

This Internet-Draft will expire on October 25, 2012.
Copyright Notice

Copyright (c) 2012 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
include Simplified BSD License text as described in Section 4.e of
the Trust Legal Provisions and are provided without warranty as
described in the Simplified BSD License.

Table of Contents

1. Introduction .................................................... 3
2. The Internet-Standard Management Framework .................. 3
3. Overview ......................................................... 3
   3.1 Conventions used in this document ....................... 3
   3.2 Terminology ................................................ 3
4. Acronyms ......................................................... 4
5. Brief description of MIB Objects ................................ 4
   5.1. Extensions to the BFD session table (bfdSessionTable) .... 4
   5.2. Example of BFD session configuration ..................... 6
      5.2.1 Example of BFD Session configuration for MPLS TE
tunnel ......................................................... 6
      5.2.2 Example of BFD Session configuration for Maintenance
Entity of MPLS-TP TE tunnel ................................ 7
6. BFD objects for session performance counters ................. 9
7. Security Considerations ........................................ 17
8. IANA Considerations ............................................ 18
9. References ...................................................... 18
   9.1 Normative References ..................................... 18
   9.2 Informative References ..................................... 18
10. Acknowledgments ............................................... 19
11. Authors’ Addresses ........................................... 19
1 Introduction

Current MIB for BFD as defined by BFD-STD-MIB is used for neighbor monitoring in IP networks. The BFD session association to the neighbors being monitored is done using the source and destination IP addresses of the neighbors configured using the respective MIB objects.

To monitor MPLS/MPLS-TP paths like tunnels or Pseudowires, there is a necessity to identify or associate the BFD session to those paths.

This memo defines an portion of the Management Information Base (MIB) for use with network management protocols in the Internet community. In particular, it extends the BFD Management Information Base BFD-STD-MIB and describes the managed objects to configure and/or monitor Bidirectional Forwarding Detection (BFD) protocol for MPLS [BFD-MPLS] and MPLS-TP networks [RFC6428].

2. The Internet-Standard Management Framework

For a detailed overview of the documents that describe the current Internet-Standard Management Framework, please refer to section 7 of RFC3410 [RFC3410].

Managed objects are accessed via a virtual information store, termed the Management Information Base or MIB. MIB objects are generally accessed through the Simple Network Management Protocol (SNMP). Objects in the MIB are defined using the mechanisms defined in the Structure of Management Information (SMI). This memo specifies a MIB module that is compliant to the SMIv2, which is described in STD 58, RFC2578, STD 58, RFC2579 and STD58, RFC2580.

3. Overview

3.1 Conventions used in this document

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

3.2 Terminology

This document adopts the definitions, acronyms and mechanisms described in [BFD], [BFD-1HOP], [BFD-MH], [BFD-MPLS], [RFC6428]. Unless otherwise stated, the mechanisms described therein will not be re-described here.
4. Acronyms

BFD: Bidirectional Forwarding Detection  
IP: Internet Protocol  
LSP: Label Switching Path  
LSR: Label Switching Router  
MIB: Management Information Base  
MPLS: Multi-Protocol Label Switching  
MPLS-TP: MPLS Transport Profile  
ME: Maintenance Entity  
MEG: Maintenance Entity Group  
MEP: Maintenance Entity End-Point  
PW: Pseudowire  
TP: Transport Profile

5. Brief description of MIB Objects

The objects described in this section support the functionality described in documents [BFD-MPLS] and [RFC6428]. The objects are defined as an extension to the BFD base MIB defined by BFD-STD-MIB.

5.1. Extensions to the BFD session table (bfdSessionTable)

The BFD session table used to identify a BFD session between a pair of nodes, as defined in BFD-STD-MIB, is extended with managed objects to achieve the required functionality in MPLS and MPLS-TP networks as described below:

1. SessionRole - Active/Passive role specification for the BFD session configured on the node. Either end of a BFD session can be configured as Active/Passive to determine which end starts transmitting the BFD control packets.

2. SessionMode - Defines the mode in which BFD session is running, defined as below:
   i. CC - Only Continuity Check and RDI functionality is performed.
   ii. CV - Provides for Continuity Check, Connectivity Verification and RDI functionalities to be supported.

3. Timer Negotiation Flag - Provides for timer negotiation to be enabled or disabled. This object can be used to tune the detection of period-misconfiguration.

4. Map Type - Indicates the type of the path being monitored by the BFD session.
This object can take the following values:

For BFD session over MPLS based paths:

- nonTeIpv4 (1) - BFD session configured for Non-TE IPv4 path
- nonTeIpv6 (2) - BFD session configured for Non-TE IPv6 path
- teIpv4 (3) - BFD session configured for a TE IPv4 path
- teIpv6 (4) - BFD session configured for a TE IPv6 path
- pw (5) - BFD session configured for a PW

For MPLS-TP based paths:

- mep (6) - BFD session configured for an MPLS-TP path (Bidirectional tunnel, PW or Sections) will map to the corresponding maintenance entity.

5. Map Pointer

A Row Pointer object which can be used to point to the first accessible object in the respective instance of the table entry identifying the path being monitored (mplsXCEntry/mplsTunnelEntry/pwEntry respectively for LSP/Tunnel/PW).

For NON-TE LSP, the Map pointer points to the corresponding mplsXCEntry.

For TE based tunnel, the Map pointer points to the corresponding instance of the mplsTunnelEntry.

For PW, this object points to the corresponding instance of pwEntry.

For MPLS-TP paths, this object points to the corresponding instance of mplsOamIdMeEntry configured to monitor the MPLS-TP path associated with the BFD session.

6. Usage of existing object bfdSessType:

Additionally existing object "bfdSessType" in the base MIB can be used with the already defined value multiHopOutOfBandSignaling(3) to specify an OOB (Out of band) mechanism [E.g. LSP Ping] for bootstrapping the BFD session.
5.2. Example of BFD session configuration

This section provides an example of BFD session configuration for an MPLS and MPLS-TP TE tunnel. This example is only meant to enable an understanding of the proposed extension and does not illustrate every permutation of the MIB.

5.2.1 Example of BFD Session configuration for MPLS TE tunnel

This section provides an example BFD session configuration for an MPLS TE tunnel. This example is only meant to enable an understanding of the proposed extension and does not illustrate every permutation of the MIB.

The following denotes the configured tunnel "head" entry:

In mplsTunnelTable:

\begin{verbatim}
{
  mplsTunnelIndex              = 100,
  mplsTunnelInstance           = 1,
  mplsTunnelIngressLSRId       = 192.0.2.1,
  mplsTunnelEgressLSRId        = 192.0.2.3,
  mplsTunnelName               = "Tunnel",
  ...                          
  mplsTunnelSignallingProto    = none (1),
  mplsTunnelSetupPrio          = 0,
  mplsTunnelHoldingPrio        = 0,
  mplsTunnelSessionAttributes  = 0,
  mplsTunnelLocalProtectInUse  = false (0),
  mplsTunnelResourcePointer    = mplsTunnelResourceMaxRate.5,
  mplsTunnelInstancePriority   = 1,
  mplsTunnelHopTableIndex      = 1,
  mplsTunnelIncludeAnyAffinity = 0,
  mplsTunnelIncludeAllAffinity = 0,
  mplsTunnelExcludeAnyAffinity = 0,
  mplsTunnelPathInUse          = 1,
  mplsTunnelRole               = head (1),
  ...                          
  mplsTunnelRowStatus          = Active
}
\end{verbatim}

BFD session parameters used to monitor this tunnel should be configured on head-end as follows:

In bfdSessTable:
BfdSessEntry ::= SEQUENCE {
-- BFD session index

bfdSessIndex = 2,
bfdSessVersionNumber = 1,
-- LSP Ping used for OOB bootstrapping
bfdSessType = multiHopOutOfBandSignaling,
...
bfdSessAdminStatus = start,
...
bfdSessDemandModeDesiredFlag = false,
bfdSessControlPlaneIndepFlag = false,
bfdSessMultipointFlag = false,
bfdSessDesiredMinTxInterval = 100000,
bfdSessReqMinRxInterval = 100000,
...
-- Indicates that the BFD session is to monitor
-- an MPLS TE tunnel
bfdMplsSessMapType = teIpv4(3),

-- OID of the first accessible object (mplsTunnelName) of
-- the mplsTunnelEntry identifying the MPLS TE tunnel (being
-- monitored using BFD) in the MPLS tunnel table.
-- A value of zeroDotzero indicates that no association
-- has been made as yet between the BFD session and the path
-- being monitored.
-- In the above OID example:
-- 100 -> Tunnel Index
-- 1 -> Tunnel instance
-- 3221225985 -> Ingress LSR Id 192.0.2.1
-- 3221225987 -> Egress LSR Id 192.0.2.3
bfdMplsSessMapPointer
= mplsTunnelName.100.1.3221225985.3221225987,
bfdSessRowStatus = createAndGo
}

Similarly BFD session would be configured on the tail-end of
the tunnel. Creating the above row will trigger
the bootstrapping of the session using LSP Ping and its
subsequent establishment over the path by de-multiplexing of
the control packets using the BFD session discriminators.

5.2.2 Example of BFD Session configuration for Maintenance Entity of
MPLS-TP TE tunnel

This example considers the OAM identifiers configuration on a
head-end LSR to manage and monitor a co-routed bidirectional MPLS
tunnel.
Only relevant objects which are applicable for IP based OAM
identifiers of co-routed MPLS tunnel are illustrated here.

In mplsOamIdMegTable:
This will create an entry in the mplsOamIdMegTable to manage and monitor the MPLS tunnel.

The following ME table is used to associate the path information to a MEG.

In mplsOamIdMeTable:
{
    -- ME index (Index to the table)
    mplsOamIdMeIndex = 1,
    -- MP index (Index to the table)
    mplsOamIdMeMpIndex = 1,
    mplsOamIdMeName = "ME1",
    mplsOamIdMeMpIfIndex = 0,
    mplsOamIdMeSourceMepIndex = 0,
    mplsOamIdMeSinkMepIndex = 0,
    mplsOamIdMeMpType = mep (1),
    mplsOamIdMeMepDirection = down (2),
    mplsOamIdMeProactiveOamPhbTCValue = 0,
    mplsOamIdMeOnDemandOamPhbTCValue = 0,
    -- RowPointer MUST point to the first accessible column of an MPLS tunnel
    mplsOamIdMeServicePointer = mplsTunnelName.1.1.1.2,
    -- Mandatory parameters needed to activate the row go here
    mplsOamIdMeRowStatus = createAndGo (4)
}

BFD session parameters used to monitor this tunnel should be configured on head-end as follows:

In bfdSessTable:
BfdSessEntry ::= SEQUENCE {
    -- BFD session index
    bfdSessIndex = 2,
    bfdSessVersionNumber = 1,
-- LSP Ping used for OOB bootstrapping
bfdSessType = multiHopOutOfBandSignaling,
...
bfdSessAdminStatus = start,
...
bfdSessDemandModeDesiredFlag = false,
bfdSessControlPlaneIndepFlag = false,
bfdSessMultipointFlag = false,
bfdSessDesiredMinTxInterval = 100000,
bfdSessReqMinRxInterval = 100000,
...
-- Indicates that the BFD session is to monitor
-- a ME of an MPLS-TP TE tunnel
bfdMplsSessMapType = mep(6),

bfdMplsSessMapPointer
  = mplsOamIdMeName.1.1.1,
bfdSessRowStatus = createAndGo
}

Similarly BFD session would be configured on the tail-end of
the tunnel. Creating the above row will trigger the bootstrapping
of the session using LSP Ping and its subsequent establishment
over the path by de-multiplexing of the control packets using
the BFD session discriminators.

5.3. BFD objects for session performance counters

BFD-STD-MIB defines BFD Session Performance Table
(bfdSessionPerfTable), for collecting per-session BFD performance
counters, as an extension to the bfdSessionTable.

The bfdSessionPerfTable is extended with the performance counters
to collect Mis-connectivity Defect, Loss of Continuity Defect
and RDI (Remote Defect Indication) counters.

1. bfdMplsSessPerfMisDefCount - Mis-connectivity defect count
   for this BFD session.
2. bfdMplsSessPerfLocDefCount - Loss of continuity defect count for
   this BFD session.
3. bfdMplsSessPerfRdiInCount - Total number of RDI messages
   received for this BFD session.
4. bfdMplsSessPerfRdiOutCount - Total number of RDI messages sent
   for this BFD session.

5.4. Notification Objects

To be added in the next version of this document.
6. BFD MPLS-MPLS-TP MIB Module Definition

BFD-EXT-STD-MIB DEFINITIONS ::= BEGIN

IMPORTS
  MODULE-IDENTITY, OBJECT-TYPE, mib-2,
  Counter32, zeroDotZero
FROM SNMPv2-SMI             -- [RFC2578]

RowPointer,TruthValue,TEXTUAL-CONVENTION
FROM SNMPv2-TC              -- [RFC2579]

MODULE-COMPLIANCE, OBJECT-GROUP
FROM SNMPv2-CONF            -- [RFC2580]

bfdSessIndex
FROM BFD-STD-MIB;

bfdMplsMib MODULE-IDENTITY
LAST-UPDATED "201204190000Z" -- April 19 2012
ORGANIZATION "IETF Bidirectional Forwarding Detection
            Working Group"
CONTACT-INFO
  "Sam Aldrin
   Huawei Technologies
   2330 Central Express Way,
   Santa Clara, CA 95051, USA
   Email: aldrin.ietf@gmail.com

   Venkatesan Mahalingam
   Dell Inc.
   350 Holger Way,
   San Jose, CA 95134, USA
   Email: venkat.mahalingams@gmail.com

   Kannan KV Sampath
   Aricent
   India
   Email: Kannan.Sampath@aricent.com

   Thomas D. Nadeau
   Juniper Networks
   10 Technology Park Drive, Westford, MA 01886
   Email: tnadeau@juniper.net"

DESCRIPTION
  "Copyright (c) 2012 IETF Trust and the persons identified
   as the document authors. All rights reserved."
This MIB module is an initial version containing objects to provide a proactive mechanism to detect faults using BFD for MPLS and MPLS-TP networks

REVISION "201204190000Z" -- April 19 2012

DESCRIPTION
" Initial version published as RFC xxx "

::= { mib-2 XXX } -- XXX to be replaced with correct value

-- RFC Ed.: assigned by IANA

-- groups in the MIB

bfMplsObjects OBJECT IDENTIFIER ::= { bfdMplsMib 0 }
bfMplsConformance OBJECT IDENTIFIER ::= { bfdMplsMib 1 }

-- Textual Conventions

SessionMapTypeTC ::= TEXTUAL-CONVENTION

STATUS current

DESCRIPTION
"Used to indicate the type of MPLS or MPLS-TP path associated to the session"

SYNTAX INTEGER {
    nonTeIpv4(1), -- mapping into LDP IPv4
    nonTeIpv6(2), -- mapping into LDP IPv6
    teIpv4(3), -- mapping into TE IPv4
    teIpv6(4), -- mapping into TE IPv6
    pw(5), -- mapping into Pseudowires
    mep(6) -- mapping into MEPs in MPLS-TP
}

-- BFD session table extensions for BFD on MPLS and MPLS-TP

-- bfdMplsSessTable - bfdSessTable Extension

bfdMplsSessTable OBJECT-TYPE
SYNTAX SEQUENCE OF BfdMplsSessEntry
MAX-ACCESS not-accessible
This table is an extension to the bfdSessTable for configuring BFD sessions for MPLS or MPLS-TP paths.

::= { bfdMplsObjects 1 }

bfdMplsSessEntry OBJECT-TYPE
SYNTAX BfdMplsSessEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"A row in this table extends a row in bfdSessTable."
INDEX { bfdSessIndex }
::= { bfdMplsSessTable 1 }

BfdMplsSessEntry ::= SEQUENCE {
  bfdMplsSessRole               INTEGER,
  bfdMplsSessMode               INTEGER,
  bfdMplsSessTmrNegotiate       TruthValue,
  bfdMplsSessMapType            SessionMapTypeTC,
  bfdMplsSessMapPointer         RowPointer
}

bfdMplsSessRole  OBJECT-TYPE
SYNTAX INTEGER {
  active(1),
  passive(2)
}
MAX-ACCESS read-create
STATUS current
DESCRIPTION
"This object specifies whether the system is playing the active(1) role or the passive(2) role for this BFD session."
REFERENCE
"RFC 5880, Section 6.1"

DEFVAL { active }
::= { bfdMplsSessEntry 1 }

bfdMplsSessMode  OBJECT-TYPE
SYNTAX INTEGER {
  cc(1),
  cv(2)
}
MAX-ACCESS read-create
STATUS  current
DESCRIPTION  
"This object specifies whether the BFD session is running in Continuity Check(CC) or the Connectivity Verification(CV) mode."
REFERENCE  
"1.RFC6428, Proactive Connectivity Verification, Continuity Check and Remote Defect Indication for MPLS Transport Profile."
DEFVAL { cc }
 ::= { bfdMplsSessEntry 2 }

bfdMplsSessTmrNegotiate  OBJECT-TYPE
SYNTAX  TruthValue
MAX-ACCESS  read-create
STATUS  current
DESCRIPTION  
"This object specifies if timer negotiation is required for the BFD session. When set to false, timer negotiation is disabled"
DEFVAL { true }
 ::= { bfdMplsSessEntry 3 }

bfdMplsSessMapType  OBJECT-TYPE
SYNTAX  SessionMapTypeTC
MAX-ACCESS  read-create
STATUS  current
DESCRIPTION  
"This object indicates the type of path being monitored by this BFD session entry."
DEFVAL { nonTeIpv4 }
 ::= { bfdMplsSessEntry 4 }

bfdMplsSessMapPointer  OBJECT-TYPE
SYNTAX  RowPointer
MAX-ACCESS  read-create
STATUS  current
DESCRIPTION  
"If bfdMplsSessMapType is nonTeIpv4(1) or nonTeIpv6(2), then this object MUST contain zeroDotZero or point to an instance of the mplsXCEntry indicating the LDP-based LSP associated with this BFD session."

If bfdMplsSessMapType is teIpv4(3) or teIpv6(4), then this object MUST contain zeroDotZero or point to an instance of the mplsTunnelEntry indicating the RSVP-based MPLS TE tunnel associated with this BFD session.
If bfdMplsSessMapType is pw(5), then this object MUST contain zeroDotZero or point to an instance of the pwEntry indicating the MPLS Pseudowire associated with this BFD session.

If bfdMplsSessMapTpye is mep(6). then this object MUST contain zeroDotZero or point to an instance identifying the mplsOamldMeEntry configured for monitoring the MPLS-TP path associated with this BFD session.

If this object points to a conceptual row instance in a table consistent with bfdMplsSessMapType but this instance does not currently exist then no valid path is associated with this session entry.

If this object contains zeroDotZero then no valid path is associated with this BFD session entry till it is populated with a valid pointer consistent with the value of bfdMplsSessMapType as explained above." DEFVAL { zeroDotZero } ::= { bfdMplsSessEntry 5 }

-- BFD Objects for Session performance
-- bdMplsSessPerfTable - bfdSessPerfTable Extension

BfdMplsSessPerfTable OBJECT-TYPE
SYNTAX SEQUENCE OF BfdMplsSessPerfEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION "This table is an extension to the bfdSessPerfTable"
 ::= { bfdMplsObjects 2 }

BfdMplsSessPerfEntry OBJECT-TYPE
SYNTAX BfdMplsSessPerfEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION "A row in this table extends the bfdSessPerfTable"
INDEX { bfdSessIndex }
 ::= { bfdMplsSessPerfTable 1 }

BfdMplsSessPerfEntry ::= SEQUENCE {
   bfdMplsSessPerfMisDefCount Counter32,
bfdMplsSessPerfLocDefCount Counter32,
bfdMplsSessPerfRdiInCount Counter32,
bfdMplsSessPerfRdiOutCount Counter32
}

bfdMplsSessPerfMisDefCount OBJECT-TYPE
SYNTAX Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"This object gives a count of the mis-connectivity defects
detected for the BFD session. For instance, this count
will be incremented when the received BFD control packet
carries an incorrect globally unique source
MEP identifier."
::= { bfdMplsSessPerfEntry 1 }

bfdMplsSessPerfLocDefCount OBJECT-TYPE
SYNTAX Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"This object gives a count of the Loss of continuity
defects detected in MPLS and MPLS-TP paths"
::= { bfdMplsSessPerfEntry 2 }

bfdMplsSessPerfRdiInCount OBJECT-TYPE
SYNTAX Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"This object gives a count of the Remote Defect
Indications received for the BFD session."
::= { bfdMplsSessPerfEntry 3 }

bfdMplsSessPerfRdiOutCount OBJECT-TYPE
SYNTAX Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"This object gives a count of the Remote Defect
Indications sent by the BFD session"
::= { bfdMplsSessPerfEntry 4 }

-- Module compliance
bfdMplsGroups
OBJECT IDENTIFIER ::= { bfdMplsConformance 1 }

bfdMplsCompliances
OBJECT IDENTIFIER ::= { bfdMplsConformance 2 }

-- Compliance requirement for fully compliant implementations.

bfdMplsModuleFullCompliance MODULE-COMPLIANCE
STATUS current
DESCRIPTION
"Compliance statement for agents that provide full
support for the BFD-EXT-STD-MIB module."

MODULE -- This module.

MANDATORY-GROUPS {
  bfdSessionExtGroup,
  bfdSessionExtPerfGroup
}
::= { bfdMplsCompliances 1 }

bfdMplsModuleReadOnlyCompliance MODULE-COMPLIANCE
STATUS current
DESCRIPTION
"Compliance requirement for implementations that only
provide read-only support for BFD-EXT-STD-MIB. Such devices
can then be monitored but cannot be configured using
this MIB module."

MODULE -- This module.

MANDATORY-GROUPS {
  bfdSessionExtGroup,
  bfdSessionExtPerfGroup
}

OBJECT bfdMplsSessRole
MIN-ACCESS read-only
DESCRIPTION "Write access is not required."

OBJECT bfdMplsSessMode
MIN-ACCESS read-only
DESCRIPTION "Write access is not required."

OBJECT bfdMplsSessTmrNegotiate
MIN-ACCESS read-only
DESCRIPTION "Write access is not required."
OBJECT bfdMplsSessMapType
MIN-ACCESS read-only
DESCRIPTION "Write access is not required."

OBJECT bfdMplsSessMapPointer
MIN-ACCESS read-only
DESCRIPTION "Write access is not required."

::= { bfdMplsCompliances 2 }

-- Units of conformance.

bfdSessionExtGroup OBJECT-GROUP
OBJECTS {
    bfdMplsSessRole,
    bfdMplsSessMode,
    bfdMplsSessTmrNegotiate,
    bfdMplsSessMapType,
    bfdMplsSessMapPointer
}
STATUS current
DESCRIPTION
"Collection of objects needed for BFD monitoring for
MPLS and MPLS-TP paths"
::= { bfdMplsGroups 1 }

bfdSessionExtPerfGroup OBJECT-GROUP
OBJECTS {
    bfdMplsSessPerfMisDefCount,
    bfdMplsSessPerfLocDefCount,
    bfdMplsSessPerfRdiInCount,
    bfdMplsSessPerfRdiOutCount
}
STATUS current
DESCRIPTION
"Collection of objects needed to monitor the
performance of BFD sessions on MPLS and MPLS-TP
paths"
::= { bfdMplsGroups 2 }

END

7. Security Considerations

To be added in the next version of this document.
8. IANA Considerations

To be added in the next version of this document.

9. References

9.1 Normative References


[BFD-MPLS] Aggarwal, R. et.al., "Bidirectional Forwarding Detection (BFD) for MPLS Label Switched Paths (LSPs)", RFC 5884, June 2010


9.2 Informative References

10. Acknowledgments

The authors would like to thank Jeffrey Haas, Mukund Mani and Lavanya Srivatsa for their valuable comments.

11. Authors’ Addresses

Sam Aldrin
Huawei Technologies
2330 Central Express Way,
Santa Clara, CA 95051, USA
Email: aldrin.ietf@gmail.com

Venkatesan Mahalingam
Dell Inc.
350 Holger Way,
San Jose, CA 95134, USA
Email: venkat.mahalingams@gmail.com

Kannan KV Sampath
Aricent
India
Email: Kannan.Sampath@aricent.com

Thomas D. Nadeau
Juniper Networks
10 Technology Park Drive, Westford, MA 01886
Email: tnadeau@juniper.net