

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: May 25, 2018

A. Mishra
O3b Networks
M. Jethanandani
November 21, 2017

BFD Performance Measurement
draft-am-bfd-performance-00

Abstract

This document describes an extension to the Bidirectional Forwarding Detection (BFD) protocol to determine the optimal BFD transmit interval for links with high one-way delay.

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 <https://datatracker.ietf.org/drafts/current/>.

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

This Internet-Draft will expire on May 25, 2018.

Copyright Notice

Copyright (c) 2017 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 (<https://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	2
2. Use Cases	2
3. BFD Performance TLV	3
4. Theory of Operations	4
5. IANA Requirements	5
6. Security Consideration	5
7. Normative References	5
Authors' Addresses	5

1. Introduction

The Bidirectional Forwarding Detection (BFD) [RFC5880] protocol operates by transmitting and receiving control frames, generally at high frequency, over the datapath being monitored. In order to prevent significant data loss due to a datapath failure, the tolerance for lost or delayed frames in the Detection Time, as defined in BFD [RFC5880] is set to the smallest feasible value.

This document proposes a mechanism to determine the smallest BFD transmit interval that can be supported on the link. This is achieved by actively measuring the one-way delay for each BFD session and setting the BFD session intervals based on the measured delay. This allows the BFD session to adapt to the fastest rate feasible on the current active path.

2. Use Cases

To ensure stability, the BFD interval is typically set to value greater than the one-way delay of the link. This value is currently manually tuned based on the largest one-way delay in the set of links over which the session can be established.

The method described in this proposal is useful in networks where the network latency is high, or varies with time. Trans-oceanic links and connectivity over geo-synchronous satellites are typical examples of links where the latency is high and the difference in latency on primary and backup paths can be significant.

Another use-case is connectivity using satellites in mid-earth orbit (MEO) or low-earth orbit (LEO). In these systems the one-way delay, while it is low (25msec to 150 msec), varies with time. This

variation, based on various factors, can be as high as 30 msec. With mobile receivers, such as ships, the delay when using such connectivity can be non-trivial to predict. This requires an automated method to determine the optimal BFD interval to allow fastest possible recovery in case of failure.

Many networks employ the use of diverse link types for redundancy where each link has significantly different link characteristics. For example, using geo-stationary orbit (GEO) satellite backup for MEO/LEO connectivity, or using fibre backup for MEO connectivity. The end-to-end BFD sessions for services running on top of the diverse transport will benefit from adaptive BFD rate.

3. BFD Performance TLV

The functionality proposed for BFD performance measurement is achieved by proposing a new BFD Performance TLV to the BFD control frame. This TLV leverages the delay measurement method defined in RFC 6374 [RFC6374]. As BFD Version 1 control frame does not have unused flags, the BFD Performance TLV overloads the BFD Authentication Flag and uses a new auth type BFDP-AUTH-TYPE (code-point TBA). The BFD Performance TLV merges the MPLS delay measurement message with the BFD authentication TLV (while removing fields that are not required for this application)

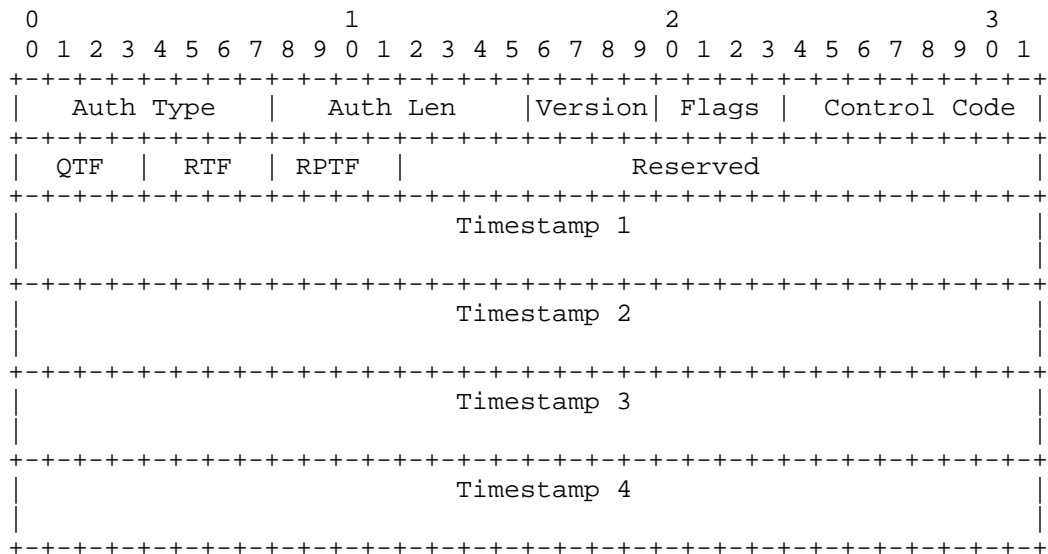


Figure 1: BFD Performance TLV

where:

Auth Type: The Authentication Type, which in this case is BFDP-AUTH-TYPE (value to be assigned).

Auth Len: The length of the Authentication Section, in bytes.

Version: Currently set to 0.

Flags: As specified in Section 3.1 of RFC 6374 [RFC6374]. The T flag is set to 1.

Control Code: As specified in Section 3.1 of RFC 6374 [RFC6374].

QTF: Querier Timestamp Format. The format of the timestamp values written by the querier, as specified in Section 3.4 of RFC 6374 [RFC6374].

RTF: Responder Timestamp Format. The format of the timestamp values written by the responder, as specified in Section 3.4 of RFC 6374 [RFC6374].

RPTF: Responder's Preferred Timestamp Format. The timestamp format preferred by the responder, as specified in Section 3.4 of RFC 6374 [RFC6374].

Timestamp 1-4: Referring to Section 2.4 of RFC 6374 [RFC6374], when a query is sent from A, Timestamp 1 is set to T1 and the other timestamp fields are set to 0. When the query is received at B, Timestamp 2 is set to T2. At this point, B copies Timestamp 1 to Timestamp 3 and Timestamp 2 to Timestamp 4, and re-initializes Timestamp 1 and Timestamp 2 to 0. When B transmits the response, Timestamp 1 is set to T3. When the response is received at A, Timestamp 2 is set to T4. The actual formats of the timestamp fields written by A and B are indicated by the Querier Timestamp Format and Responder Timestamp Format fields respectively.

The mapping of timestamps to the Timestamp 1-4 fields is designed to ensure that transmit timestamps are always written at the same fixed offset in the packet, and likewise for receive timestamps. This property is important for hardware processing.

4. Theory of Operations

This delay measurement follows the method defined in Section 2.4 of RFC 6374 [RFC6374].

The message is classified using the BFD authentication method defined in RFC5880 [RFC5880].

Method for determining the optimal BFD interval for a link with certain delay characteristics is implementation specific and beyond the scope of this document.

5. IANA Requirements

Requesting new BFD Authentication Type for BFD Performance TLV.

6. Security Consideration

Other than concerns raised in BFD [RFC5880], there are no new concerns with this proposal.

7. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [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>>.
- [RFC6374] Frost, D. and S. Bryant, "Packet Loss and Delay Measurement for MPLS Networks", RFC 6374, DOI 10.17487/RFC6374, September 2011, <<https://www.rfc-editor.org/info/rfc6374>>.

Authors' Addresses

Ashesh Mishra
O3b Networks

Email: mishra.ashesh@gmail.com

Mahesh Jethanandani

Email: mjethanandani@gmail.com

Network Working Group
Internet Draft
Intended Status: Informational
Expiration Date: February 4, 2019

E. Chen
N. Shen
Cisco Systems
R. Raszuk
Bloomberg LP
August 3, 2018

Unsolicited BFD for Sessionless Applications
draft-chen-bfd-unsolicited-03.txt

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/lid-abstracts.html>

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

This Internet-Draft will expire on February 4, 2019.

Copyright Notice

Copyright (c) 2018 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.

Abstract

For operational simplification of "sessionless" applications using BFD, in this document we present procedures for "unsolicited BFD" that allow a BFD session to be initiated by only one side, and be established without explicit per-session configuration or registration by the other side (subject to certain per-interface or per-router policies).

1. Introduction

The current implementation and deployment practice for BFD ([RFC5880] and [RFC5881]) usually requires BFD sessions be explicitly configured or registered on both sides. This requirement is not an issue when an application like BGP [RFC4271] has the concept of a "session" that involves both sides for its establishment. However, this requirement can be operationally challenging when the prerequisite "session" does not naturally exist between two endpoints in an application. Simultaneous configuration and coordination may be required on both sides for BFD to take effect. For example:

- o When BFD is used to keep track of the "liveness" of the nexthop of static routes. Although only one side may need the BFD functionality, currently both sides need to be involved in specific configuration and coordination and in some cases static routes are created unnecessarily just for BFD.
- o When BFD is used to keep track of the "liveness" of the third-party nexthop of BGP routes received from the Route Server [RFC7947] at an Internet Exchange Point (IXP). As the third-party nexthop is different from the peering address of the Route Server, for BFD to work, currently two routers peering with the Route Server need to have routes and nexthops from each other (although indirectly via the Router Server), and the nexthop of each router must be present at the same time. These issues are also discussed in [I-D.ietf-idr-rs-bfd].

Clearly it is beneficial and desirable to reduce or eliminate unnecessary configurations and coordination in these "sessionless" applications using BFD.

In this document we present procedures for "unsolicited BFD" that allow a BFD session to be initiated by only one side, and be established without explicit per-session configuration or

registration by the other side (subject to certain per-interface or per-router policies).

With "unsolicited BFD" there is potential risk for excessive resource usage by BFD from "unexpected" remote systems. To mitigate such risks, several mechanisms are recommended in the Security Considerations section.

Compared to the "Seamless BFD" [RFC7880], this proposal involves only minor procedural enhancements to the widely deployed BFD itself. Thus we believe that this proposal is inherently simpler in the protocol itself and deployment. As an example, it does not require the exchange of BFD discriminators over an out-of-band channel before the BFD session bring-up.

When BGP Add-Path [RFC7911] is deployed at an IXP using the Route Server, multiple BGP paths (when exist) can be made available to the clients of the Router Server as described in [RFC7947]. The "unsolicited BFD" can be used in BGP route selection by these clients to eliminate paths with "inaccessible nexthops".

1.1. Specification of Requirements

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].

2. Procedures for Unsolicited BFD

With "unsolicited BFD", one side takes the "Active role" and the other side takes only the "Passive role" as described in [RFC5880].

On the passive side, the "unsolicited BFD" SHOULD be configured explicitly on an interface. The BFD parameters can be either per-interface or per-router based. It MAY also choose to use the parameters that the active side uses in its BFD Control packets. The "Discriminator", however, MUST be chosen to allow multiple unsolicited BFD sessions.

The active side initiates the BFD Control packets as specified in [RFC5880]. The passive side does not initiate the BFD Control packets.

When the passive side receives a BFD Control packet from the active side with 0 as the "remote-discriminator", and it does not find an existing session with the same source address as in the packet and

"unsolicited BFD" is allowed on the interface by local policy, it SHOULD then create a matching BFD session toward the active side (based on the source address and destination address in the BFD Control packet) as if the session were locally registered. It would then start sending the BFD Control packets and perform necessary procedure for bringing up, maintaining and tearing down the BFD session. If the BFD session fails to get established within certain specified time, or if an established BFD session goes down, the passive side would stop sending BFD Control packets and delete the BFD session created until the BFD Control packets is initiated by the active side again.

The "Passive role" may change to the "Active role" when a local client registers for the same BFD session, and from the "Active role" to the "Passive role" when there is no longer any locally registered client for the BFD session.

3. IANA Considerations

This documents makes no IANA requests.

4. Security Considerations

The same security considerations as those described in [RFC5880] and [RFC5881] apply to this document. With "unsolicited BFD" there is potential risk for excessive resource usage by BFD from "unexpected" remote systems. To mitigate such risks, the following measures are RECOMMENDED:

- o Limit the feature to specific interfaces, and to a single-hop BFD with "TTL=255" [RFC5082]. In addition make sure the source address of an incoming BFD packet belongs to the subnet of the interface from which the BFD packet is received.
- o Apply "access control" to allow BFD packets only from certain subnets or hosts.
- o Deploy the feature only in certain "trustworthy" environment, e.g., at an IXP, or between a provider and its customers.
- o Adjust BFD parameters as needed for the particular deployment and scale.
- o Use BFD authentication.

5. Acknowledgments

TBD

6. References

6.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<http://www.rfc-editor.org/info/rfc2119>>.
- [RFC5082] Gill, V., Heasley, J., Meyer, D., Savola, P., Ed., and C. Pignataro, "The Generalized TTL Security Mechanism (GTSM)", RFC 5082, October 2007.
- [RFC5880] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD)", RFC 5880, DOI 10.17487/RFC5880, June 2010, <<http://www.rfc-editor.org/info/rfc5880>>.
- [RFC5881] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD) for IPv4 and IPv6 (Single Hop)", RFC 5881, DOI 10.17487/RFC5881, June 2010, <<http://www.rfc-editor.org/info/rfc5881>>.

6.2. Informative References

- [RFC4271] Rekhter, Y., Ed., Li, T., Ed., and S. Hares, Ed., "A Border Gateway Protocol 4 (BGP-4)", RFC 4271, DOI 10.17487/RFC4271, January 2006, <<http://www.rfc-editor.org/info/rfc4271>>.
- [RFC7880] Pignataro, C., Ward, D., Akiya, N., Bhatia, M., and S. Pallagatti, "Seamless Bidirectional Forwarding Detection (S-BFD)", RFC 7880, DOI 10.17487/RFC7880, July 2016, <<http://www.rfc-editor.org/info/rfc7880>>.
- [RFC7911] Walton, D., Retana, A., Chen, E., and J. Scudder, "Advertisement of Multiple Paths in BGP", RFC 7911, DOI 10.17487/RFC7911, July 2016, <<http://www.rfc-editor.org/info/rfc7911>>.
- [RFC7947] Jasinska, E., Hilliard, N., Raszuk, R., and N. Bakker, "Internet Exchange BGP Route Server", RFC 7947,

DOI 10.17487/RFC7947, September 2016,
<<http://www.rfc-editor.org/info/rfc7947>>.

[I-D.ietf-idr-rs-bfd]

Bush, R., J. Haas, J. Scudder, A. Nipper, and T. King,
"Making Route Servers Aware of Data Link Failures at
IXPs", draft-ietf-idr-rs-bfd-03 (work in progress), July
2017.

7. Authors' Addresses

Enke Chen
Cisco Systems
560 McCarthy Blvd.
Milpitas, CA 95035
USA

Email: enkechen@cisco.com

Naiming Shen
Cisco Systems
560 McCarthy Blvd.
Milpitas, CA 95035
USA

Email: naiming@cisco.com

Robert Raszuk
Bloomberg LP
731 Lexington Ave
New York City, NY 10022
USA

Email: robert@raszuk.net

Internet Engineering Task Force
Internet-Draft
Updates: 5880 (if approved)
Intended status: Standards Track
Expires: June 16, 2019

D. Katz
Juniper Networks
D. Ward
Cisco Systems
S. Pallagatti, Ed.
Rtbrick
G. Mirsky, Ed.
ZTE Corp.
December 13, 2018

BFD for Multipoint Networks
draft-ietf-bfd-multipoint-19

Abstract

This document describes extensions to the Bidirectional Forwarding Detection (BFD) protocol for its use in multipoint and multicast networks.

This document updates RFC 5880.

Status of This Memo

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

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

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

This Internet-Draft will expire on June 16, 2019.

Copyright Notice

Copyright (c) 2018 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 (<https://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	2
2. Keywords	3
3. Goals	4
4. Overview	4
5. Protocol Details	5
5.1. Multipoint BFD Control Packets	5
5.2. Session Model	5
5.3. Session Failure Semantics	5
5.4. State Variables	5
5.4.1. New State Variable Values	6
5.4.2. State Variable Initialization and Maintenance	6
5.5. State Machine	6
5.6. Session Establishment	7
5.7. Discriminators and Packet Demultiplexing	7
5.8. Packet consumption on tails	8
5.9. Bringing Up and Shutting Down Multipoint BFD Service	8
5.10. Timer Manipulation	9
5.11. Detection Times	10
5.12. State Maintenance for Down/AdminDown Sessions	10
5.12.1. MultipointHead Sessions	10
5.12.2. MultipointTail Sessions	10
5.13. Base Specification Text Replacement	10
5.13.1. Reception of BFD Control Packets	11
5.13.2. Demultiplexing BFD Control Packets	13
5.13.3. Transmitting BFD Control Packets	15
6. Congestion Considerations	18
7. IANA Considerations	19
8. Security Considerations	19
9. Contributors	20
10. Acknowledgments	20
11. References	20
11.1. Normative References	20
11.2. Informational References	20
Authors' Addresses	21

1. Introduction

The Bidirectional Forwarding Detection protocol [RFC5880] specifies a method for verifying unicast connectivity between a pair of systems. This document updates [RFC5880] by defining a new method for using

BFD. This new method provides verification of multipoint or multicast connectivity between a multipoint sender (the "head") and a set of one or more multipoint receivers (the "tails").

As multipoint transmissions are inherently unidirectional, this mechanism purports only to verify this unidirectional connectivity. Although this seems in conflict with the "Bidirectional" in BFD, the protocol is capable of supporting this use case. Use of BFD in Demand mode allows a tail to monitor the availability of a multipoint path even without the existence of some kind of a return path to the head. As an option, if a return path from a tail to the head exists, the tail may notify the head of the lack of multipoint connectivity. Details of tail notification to the head are outside the scope of this document and are discussed in [I-D.ietf-bfd-multipoint-active-tail].

This application of BFD allows for the tails to detect a lack of connectivity from the head. For some applications such detection of the failure at the tail is useful. For example, use of multipoint BFD to enable fast failure detection and faster failover in multicast VPN described in [I-D.ietf-bess-mvpn-fast-failover]. Due to unidirectional nature, virtually all options and timing parameters are controlled by the head.

Throughout this document, the term "multipoint" is defined as a mechanism by which one or more systems receive packets sent by a single sender. This specifically includes such things as IP multicast and point-to-multipoint MPLS.

The term "connectivity" in this document is not being used in the context of connectivity verification in transport network but as an alternative to "continuity", i.e., the existence of a forwarding path between the sender and the receiver.

This document effectively updates and extends the base BFD specification [RFC5880].

2. Keywords

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

3. Goals

The primary goal of this mechanism is to allow tails to rapidly detect the fact that multipoint connectivity from the head has failed.

Another goal is for the mechanism to work on any multicast technology.

A further goal is to support multiple, overlapping point-to-multipoint paths, as well as multipoint-to-multipoint paths, and to allow point-to-point BFD sessions to operate simultaneously among the systems participating in Multipoint BFD.

It is not a goal for this protocol to verify point-to-point bi-directional connectivity between the head and any tail. This can be done independently (and with no penalty in protocol overhead) by using point-to-point BFD.

4. Overview

The heart of this protocol is the periodic transmission of BFD Control packets along a multipoint path, from the head to all tails on the path. The contents of the BFD packets provide the means for the tails to calculate the detection time for path failure. If no BFD Control packets are received by a tail for a detection time, the tail declares that the path has failed. For some applications this is the only mechanism necessary; the head can remain ignorant of the status of connectivity to the tails.

The head of a multipoint BFD session may wish to be alerted to the tails' connectivity (or lack thereof). Details of how the head keeps track of tails and how tails alert their connectivity to the head are outside the scope of this document and are discussed in [I-D.ietf-bfd-multipoint-active-tail].

Although this document describes a single head and a set of tails spanned by a single multipoint path, the protocol is capable of supporting (and discriminating between) more than one multipoint path at both heads and tails, as described in Section 5.7 and Section 5.13.2. Furthermore, the same head and tail may share multiple multipoint paths, and a multipoint path may have multiple heads.

5. Protocol Details

This section describes the operation of Multipoint BFD in detail.

5.1. Multipoint BFD Control Packets

Multipoint BFD Control packets (packets sent by the head over a multipoint path) are explicitly marked as such, via the setting of the M bit [RFC5880]. This means that Multipoint BFD does not depend on the recipient of a packet to know whether the packet was received over a multipoint path. This can be useful in scenarios where this information may not be available to the recipient.

5.2. Session Model

Multipoint BFD is modeled as a set of sessions of different types. The elements of procedure differ slightly for each type.

The head has a session of type MultipointHead, as defined in Section 5.4.1, that is bound to a multipoint path. Multipoint BFD Control packets are sent by this session over the multipoint path, and no BFD Control packets are received by it.

Each tail has a session of type MultipointTail, as defined in Section 5.4.1, associated with a multipoint path. These sessions receive BFD Control packets from the head over the multipoint path.

5.3. Session Failure Semantics

The semantics of session failure is subtle enough to warrant further explanation.

MultipointHead sessions cannot fail (since they are controlled administratively).

If a MultipointTail session fails, it means that the tail definitely has lost contact with the head (or the head has been administratively disabled) and the tail may use mechanisms other than BFD, e.g., logging or NETCONF [RFC6241], to send a notification to the user.

5.4. State Variables

Multipoint BFD introduces some new state variables and modifies the usage of a few existing ones.

5.4.1. New State Variable Values

A number of new values of the state variable `bfd.SessionType` are added to the base BFD [RFC5880] and base S-BFD [RFC7880] specifications in support of Multipoint BFD.

`bfd.SessionType`

The type of this session as defined in [RFC7880]. Newly added values are:

`PointToPoint`: Classic point-to-point BFD, as described in [RFC5880].

`MultipointHead`: A session on the head responsible for the periodic transmission of multipoint BFD Control packets along the multipoint path.

`MultipointTail`: A multipoint session on a tail.

This variable **MUST** be initialized to the appropriate type when the session is created.

5.4.2. State Variable Initialization and Maintenance

Some state variables defined in section 6.8.1 of [RFC5880] need to be initialized or manipulated differently depending on the session type.

`bfd.RequiredMinRxInterval`

This variable **MUST** be initialized to 0 for session type `MultipointHead`.

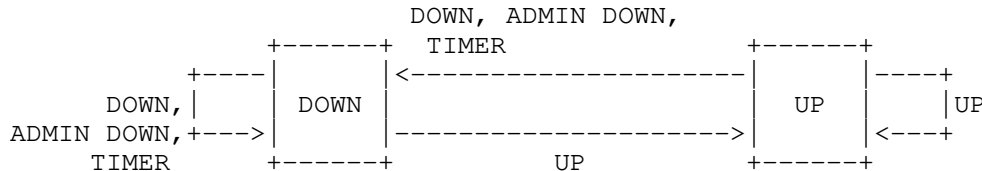
`bfd.DemandMode`

This variable **MUST** be initialized to 1 for session type `MultipointHead` and **MUST** be initialized to 0 for session type `MultipointTail`.

5.5. State Machine

The BFD state machine works slightly differently in the multipoint application. In particular, since there is a many-to-one mapping, three-way handshakes for session establishment and teardown are neither possible nor appropriate. As such, there is no Init state. Sessions of type `MultipointHead` **MUST NOT** send BFD control packets with the State field being set to INIT, and those packets **MUST** be ignored on receipt.

The following diagram provides an overview of the state machine for session type MultipointTail. The notation on each arc represents the state of the remote system (as received in the State field in the BFD Control packet) or indicates the expiration of the Detection Timer.



Sessions of type `MultipointHead` never receive packets and have no `Detection Timer`, and as such all state transitions are administratively driven.

5.6. Session Establishment

Unlike point-to-point BFD, Multipoint BFD provides a form of the discovery mechanism for tails to discover the head. The minimum amount of a priori information required both on the head and tails is the binding to the multipoint path over which BFD is running. The head transmits Multipoint BFD packets on that path, and the tails listen for BFD packets on that path. All other information can be determined dynamically.

A session of type MultipointHead is created for each multipoint path over which the head wishes to run BFD. This session runs in the Active role, per section 6.1 [RFC5880]. Except when administratively terminating BFD service, this session is always in state Up and always operates in Demand mode. No received packets are ever demultiplexed to the MultipointHead session. In this sense, it is a degenerate form of a session.

Sessions on the tail MAY be established dynamically, based on the receipt of a Multipoint BFD Control packet from the head, and are of type MultipointTail. Tail sessions always take the Passive role, per section 6.1 [RFC5880].

5.7. Discriminators and Packet Demultiplexing

The use of Discriminators is somewhat different in Multipoint BFD than in Point-to-point BFD.

The head sends Multipoint BFD Control packets over the multipoint path via the MultipointHead session with My Discriminator set to a

value bound to the multipoint path, and with Your Discriminator set to zero.

IP and MPLS multipoint tails MUST demultiplex BFD packets based on a combination of the source address, My Discriminator and the identity of the multipoint path which the Multipoint BFD Control packet was received from. Together they uniquely identify the head of the multipoint path. Bootstrapping a BFD session to multipoint MPLS LSP may use the control plane, e.g., as described in [I-D.ietf-bess-mvpn-fast-failover], and is outside the scope of this document.

Note that, unlike point-to-point sessions, the My Discriminator value on MultipointHead session MUST NOT be changed during the life of a session. This is a side effect of the more complex demultiplexing scheme.

5.8. Packet consumption on tails

BFD packets received on tails for an IP multicast group MUST be consumed by tails and MUST NOT be forwarded to receivers. Nodes with the BFD session of type MultipointTail MUST identify packets received on an IP multipoint path as BFD control packet if the destination UDP port value equals 3784.

For multipoint LSPs, when IP/UDP encapsulation of BFD control packets is used, MultipointTail MUST expect destination UDP port 3784. Destination IP address of BFD control packet MUST be in 127.0.0.0/8 range for IPv4 or in 0:0:0:0:0:FFFF:7F00:0/104 range for IPv6. The use of these destination addresses is consistent with the explanations and usage in [RFC8029]. Packets identified as BFD packets MUST be consumed by MultipointTail and demultiplexed as described in Section 5.13.2. Use of other types of encapsulation of the BFD control message over multipoint LSP is outside the scope of this document.

5.9. Bringing Up and Shutting Down Multipoint BFD Service

Because there is no three-way handshake in Multipoint BFD, a newly started head (that does not have any previous state information available) SHOULD start with bfd.SessionState set to Down and bfd.RequiredMinRxInterval MUST be set to zero in the MultipointHead session. The session SHOULD remain in this state for a time equal to (bfd.DesiredMinTxInterval * bfd.DetectMult). This will ensure that all MultipointTail sessions are reset (so long as the restarted head is using the same or a larger value of bfd.DesiredMinTxInterval than it did previously).

Multipoint BFD service is brought up by administratively setting `bfd.SessionState` to Up in the MultipointHead session.

The head of a multipoint BFD session may wish to shut down its BFD service in a controlled fashion. This is desirable because the tails need not wait a detection time prior to declaring the multipoint session to be down (and taking whatever action is necessary in that case).

To shut down a multipoint session in a controlled fashion the head MUST administratively set `bfd.SessionState` in the MultipointHead session to either Down or AdminDown and SHOULD set `bfd.RequiredMinRxInterval` to zero. The session SHOULD send BFD Control packets in this state for a period equal to $(\text{bfd.DesiredMinTxInterval} * \text{bfd.DetectMult})$. Alternatively, the head MAY stop transmitting BFD Control packets and not send any more BFD Control packets with the new state (Down or AdminDown). Tails will declare the multipoint session down only after the detection time interval runs out.

5.10. Timer Manipulation

Because of the one-to-many mapping, a session of type MultipointHead SHOULD NOT initiate a Poll Sequence in conjunction with timer value changes. However, to indicate a change in the packets, MultipointHead session MUST send packets with the P bit set. MultipointTail session MUST NOT reply if the packet has M and P bits set and `bfd.RequiredMinRxInterval` set to 0. Because the Poll Sequence is not used, the tail cannot negotiate down MultipointHead's transmit interval. If the value of Desired Min TX Interval in the BFD Control packet received by MultipointTail is too high (that determination may change in time based on the current environment) it must be handled by the implementation and may be controlled by local policy, e.g., close the MultipointTail session.

The MultipointHead, when changing the transmit interval to a higher value, MUST send BFD control packets with P bit set at the old transmit interval before using the higher value in order to avoid false detection timeouts at the tails. MultipointHead session MAY also wait some amount of time before making the changes to the transmit interval (through configuration).

Change in the value of `bfd.RequiredMinRxInterval` is outside the scope of this document and is discussed in [I-D.ietf-bfd-multipoint-active-tail].

5.11. Detection Times

Multipoint BFD is inherently asymmetric. As such, each session type has a different approach to detection times.

Since MultipointHead sessions never receive packets, they do not calculate a detection time.

MultipointTail sessions cannot influence the transmission rate of the MultipointHead session using the Required Min Rx Interval field because of its one-to-many nature. As such, the detection time calculation for a MultipointTail session does not use `bfd.RequiredMinRxInterval`. The detection time is calculated as the product of the last received values of Desired Min TX Interval and Detect Mult.

The value of `bfd.DetectMult` may be changed at any time on any session type.

5.12. State Maintenance for Down/AdminDown Sessions

The length of time session state is kept after the session goes down determines how long the session will continue to send BFD Control packets (since no packets can be sent after the session is destroyed).

5.12.1. MultipointHead Sessions

When a MultipointHead session transitions to states Down or AdminDown, the state SHOULD be maintained for a period equal to $(\text{bfd.DesiredMinTxInterval} * \text{bfd.DetectMult})$ to ensure that the tails more quickly detect the session going down (by continuing to transmit BFD Control packets with the new state).

5.12.2. MultipointTail Sessions

MultipointTail sessions MAY be destroyed immediately upon leaving Up state, since tail will transmit no packets.

Otherwise, MultipointTail sessions SHOULD be maintained as long as BFD Control packets are being received by it (which by definition will indicate that the head is not Up).

5.13. Base Specification Text Replacement

The following sections are meant to replace the corresponding sections in the base specification [RFC5880] in support of BFD for

multipoint networks while not changing processing for point-to-point BFD.

5.13.1. Reception of BFD Control Packets

The following procedure replaces the entire section 6.8.6 of [RFC5880].

When a BFD Control packet is received, the following procedure MUST be followed, in the order specified. If the packet is discarded according to these rules, processing of the packet MUST cease at that point.

If the version number is not correct (1), the packet MUST be discarded.

If the Length field is less than the minimum correct value (24 if the A bit is clear, or 26 if the A bit is set), the packet MUST be discarded.

If the Length field is greater than the payload of the encapsulating protocol, the packet MUST be discarded.

If the Detect Mult field is zero, the packet MUST be discarded.

If the My Discriminator field is zero, the packet MUST be discarded.

Demultiplex the packet to a session according to Section 5.13.2 below. The result is either a session of the proper type, or the packet is discarded (and packet processing MUST cease).

If the A bit is set and no authentication is in use (bfd.AuthType is zero), the packet MUST be discarded.

If the A bit is clear and authentication is in use (bfd.AuthType is nonzero), the packet MUST be discarded.

If the A bit is set, the packet MUST be authenticated under the rules of [RFC5880] section 6.7, based on the authentication type in use (bfd.AuthType). This may cause the packet to be discarded.

Set bfd.RemoteDiscr to the value of My Discriminator.

Set bfd.RemoteState to the value of the State (Sta) field.

Set bfd.RemoteDemandMode to the value of the Demand (D) bit.

Set bfd.RemoteMinRxInterval to the value of Required Min RX Interval.

If the Required Min Echo RX Interval field is zero, the transmission of Echo packets, if any, MUST cease.

If a Poll Sequence is being transmitted by the local system and the Final (F) bit in the received packet is set, the Poll Sequence MUST be terminated.

If bfd.SessionType is PointToPoint, update the transmit interval as described in [RFC5880] section 6.8.2.

If bfd.SessionType is PointToPoint, update the Detection Time as described in section 6.8.4 of [RFC5880].

Else

If bfd.SessionType is MultipointTail, then update the Detection Time as the product of the last received values of Desired Min TX Interval and Detect Mult, as described in Section 5.11 of this specification.

If bfd.SessionState is AdminDown

Discard the packet

If the received state is AdminDown

If bfd.SessionState is not Down

Set bfd.LocalDiag to 3 (Neighbor signaled session down)

Set bfd.SessionState to Down

Else

If bfd.SessionState is Down

If bfd.SessionType is PointToPoint

If received State is Down

Set bfd.SessionState to Init

Else if received State is Init

Set bfd.SessionState to Up

```
    Else (bfd.SessionType is not PointToPoint)

        If received State is Up

            Set bfd.SessionState to Up

    Else if bfd.SessionState is Init

        If received State is Init or Up

            Set bfd.SessionState to Up

    Else (bfd.SessionState is Up)

        If received State is Down

            Set bfd.LocalDiag to 3 (Neighbor signaled session down)

            Set bfd.SessionState to Down

Check to see if Demand mode should become active or not (see
[RFC5880] section 6.6).

If bfd.RemoteDemandMode is 1, bfd.SessionState is Up and
bfd.RemoteSessionState is Up, Demand mode is active on the remote
system and the local system MUST cease the periodic transmission
of BFD Control packets (see Section 5.13.3).

If bfd.RemoteDemandMode is 0, or bfd.SessionState is not Up, or
bfd.RemoteSessionState is not Up, Demand mode is not active on the
remote system and the local system MUST send periodic BFD Control
packets (see Section 5.13.3).

If the Poll (P) bit is set, and bfd.SessionType is PointToPoint,
send a BFD Control packet to the remote system with the Poll (P)
bit clear, and the Final (F) bit set (see Section 5.13.3).

If the packet was not discarded, it has been received for purposes
of the Detection Time expiration rules in [RFC5880] section 6.8.4.
```

5.13.2. Demultiplexing BFD Control Packets

This section is part of the replacement for [RFC5880] section 6.8.6, separated for clarity.

```
    If the Multipoint (M) bit is set
```


If the Your Discriminator field is nonzero, the packet MUST be discarded.

Select a session as based on source address, My Discriminator and the identity of the multipoint path which the Multipoint BFD Control packet was received.

If a session is found, and bfd.SessionType is not MultipointTail, the packet MUST be discarded.

Else

If a session is not found, a new session of type MultipointTail MAY be created, or the packet MAY be discarded. This choice can be controlled by the local policy, e.g., by setting a maximum number of MultipointTail sessions. Use of the local policy and the exact mechanism of it are outside the scope of this specification.

Else (Multipoint bit is clear)

If the Your Discriminator field is nonzero

Select a session based on the value of Your Discriminator.
If no session is found, the packet MUST be discarded.

Else (Your Discriminator is zero)

If the State field is not Down or AdminDown, the packet MUST be discarded.

Otherwise, the session MUST be selected based on some combination of other fields, possibly including source addressing information, the My Discriminator field, and the interface over which the packet was received. The exact method of selection is application-specific and is thus outside the scope of this specification.

If a matching session is found, and bfd.SessionType is not PointToPoint, the packet MUST be discarded.

If a matching session is not found, a new session of type PointToPoint MAY be created, or the packet MAY be discarded. This choice MAY be controlled by a local policy and is outside the scope of this specification.

If the State field is Init and bfd.SessionType is not PointToPoint, the packet MUST be discarded.

5.13.3. Transmitting BFD Control Packets

The following procedure replaces the entire section 6.8.7 of [RFC5880].

With the exceptions listed in the remainder of this section, a system MUST NOT transmit BFD Control packets at an interval less than the larger of `bfd.DesiredMinTxInterval` and `bfd.RemoteMinRxInterval`, less applied jitter (see below). In other words, the system reporting the slower rate determines the transmission rate.

The periodic transmission of BFD Control packets MUST be jittered on a per-packet basis by up to 25%, that is, the interval MUST be reduced by a random value of 0 to 25%, in order to avoid self-synchronization with other systems on the same subnetwork. Thus, the average interval between packets will be roughly 12.5% less than that negotiated.

If `bfd.DetectMult` is equal to 1, the interval between transmitted BFD Control packets MUST be no more than 90% of the negotiated transmission interval, and MUST be no less than 75% of the negotiated transmission interval. This is to ensure that, on the remote system, the calculated Detection Time does not pass prior to the receipt of the next BFD Control packet.

A system MUST NOT transmit any BFD Control packets if `bfd.RemoteDiscr` is zero and the system is taking the Passive role.

A system MUST NOT transmit any BFD Control packets if `bfd.SessionType` is `MultipointTail`.

A system MUST NOT periodically transmit BFD Control packets if Demand mode is active on the remote system (`bfd.RemoteDemandMode` is 1, `bfd.SessionState` is Up, and `bfd.RemoteSessionState` is Up) and a Poll Sequence is not being transmitted.

A system MUST NOT periodically transmit BFD Control packets if `bfd.RemoteMinRxInterval` is zero.

If `bfd.SessionType` is `MultipointHead`, the transmit interval MUST be set to `bfd.DesiredMinTxInterval` (this should happen automatically, as `bfd.RemoteMinRxInterval` will be zero).

If `bfd.SessionType` is not `MultipointHead`, the transmit interval MUST be recalculated whenever `bfd.DesiredMinTxInterval` changes, or whenever `bfd.RemoteMinRxInterval` changes, and is equal to the greater of those two values. See [RFC5880] sections 6.8.2 and 6.8.3 for details on transmit timers.

A system MUST NOT set the Demand (D) bit if `bfd.SessionType` is `MultipointTail`.

A system MUST NOT set the Demand (D) bit if `bfd.SessionType` is `PointToPoint` unless `bfd.DemandMode` is 1, `bfd.SessionState` is Up, and `bfd.RemoteSessionState` is Up.

If `bfd.SessionType` is `PointToPoint` or `MultipointHead`, a BFD Control packet SHOULD be transmitted during the interval between periodic Control packet transmissions when the contents of that packet would differ from that in the previously transmitted packet (other than the Poll and Final bits) in order to more rapidly communicate a change in state.

The contents of transmitted BFD Control packets MUST be set as follows:

Version

Set to the current version number (1).

Diagnostic (Diag)

Set to `bfd.LocalDiag`.

State (Sta)

Set to the value indicated by `bfd.SessionState`.

Poll (P)

Set to 1 if the local system is sending a Poll Sequence or is a session of type `MultipointHead` soliciting the identities of the tails, or 0 if not.

Final (F)

Set to 1 if the local system is responding to a Control packet received with the Poll (P) bit set, or 0 if not.

Control Plane Independent (C)

Set to 1 if the local system's BFD implementation is independent of the control plane (it can continue to function through a disruption of the control plane).

Authentication Present (A)

Set to 1 if authentication is in use in this session (bfd.AuthType is nonzero), or 0 if not.

Demand (D)

Set to bfd.DemandMode if bfd.SessionState is Up and bfd.RemoteSessionState is Up. Set to 1 if bfd.SessionType is MultipointHead. Otherwise it is set to 0.

Multipoint (M)

Set to 1 if bfd.SessionType is MultipointHead. Otherwise, it is set to 0.

Detect Mult

Set to bfd.DetectMult.

Length

Set to the appropriate length, based on the fixed header length (24) plus any Authentication Section.

My Discriminator

Set to bfd.LocalDiscr.

Your Discriminator

Set to bfd.RemoteDiscr.

Desired Min TX Interval

Set to bfd.DesiredMinTxInterval.

Required Min RX Interval

Set to bfd.RequiredMinRxInterval.

Required Min Echo RX Interval

Set to 0 if bfd.SessionType is MultipointHead or MultipointTail. Otherwise, set to the minimum required Echo packet receive interval for this session. If this field is set to zero, the local system is unwilling or unable to loop back BFD Echo packets to the remote system, and the remote system will not send Echo packets.

Authentication Section

Included and set according to the rules in [RFC5880] section 6.7 if authentication is in use (bfd.AuthType is nonzero). Otherwise, this section is not present.

6. Congestion Considerations

As a foreword, although congestion can occur because of a number of factors, it should be noted that high transmission rates are by themselves subject to creating congestion either along the path or at the tail end(s). As such, as stated in [RFC5883]:

"it is required that the operator correctly provision the rates at which BFD is transmitted to avoid congestion (e.g link, I/O, CPU) and false failure detection."

Use of BFD in multipoint networks, as specified in this document, over multiple hops requires consideration of the mechanisms to react to network congestion. Requirements stated in Section 7 of the BFD base specification [RFC5880] equally apply to BFD in multipoint networks and are repeated here:

"When BFD is used across multiple hops, a congestion control mechanism MUST be implemented, and when congestion is detected, the BFD implementation MUST reduce the amount of traffic it generates."

The mechanism to control the load of BFD traffic MAY use BFD's configuration interface to control BFD state variable bfd.DesiredMinTxInterval. However, such a control loop do not form part of the BFD protocol itself and its specification is thus outside the scope of this document.

Additional considerations apply to BFD in multipoint networks, as specified in this document. Indeed, because a tail does not transmit any BFD Control packets to the head of the BFD session, such head node has no BFD based mechanism to be aware of the state of the session at the tail. In the absence of any other mechanism, the head of the session could thus continue to send packets towards the tail(s) even though a link failure has happened. In such a scenario when it is required for the head of the session to be aware of the state of the tail of the session, it is RECOMMENDED to implement [I-D.ietf-bfd-multipoint-active-tail].

7. IANA Considerations

This document has no actions for IANA.

8. Security Considerations

The same security considerations as those described in [RFC5880] apply to this document. Additionally, implementations that create MultipointTail sessions dynamically upon receipt of Multipoint BFD Control packets MUST implement protective measures to prevent an infinite number of MultipointTail sessions being created. Below are listed some points to be considered in such implementations.

If a Multipoint BFD Control packet did not arrive on a multicast path (e.g., on the expected interface, with expected MPLS label, etc), then a MultipointTail session should not be created.

If redundant streams are expected for a given multicast stream, then the implementations should not create more MultipointTail sessions than the number of streams. Additionally, when the number of MultipointTail sessions exceeds the number of expected streams, then the implementation should generate an alarm to users to indicate the anomaly.

The implementation should have a reasonable upper bound on the number of MultipointHead sessions that can be created, with the upper bound potentially being computed based on the load these would generate.

The implementation should have a reasonable upper bound on the number of MultipointTail sessions that can be created, with the upper bound potentially being computed based on the number of multicast streams that the system is expecting.

If authentication is in use, the head and all tails may be configured to have a common authentication key in order for the tails to validate multipoint BFD Control packets.

Shared keys in multipoint scenarios allow any tail to spoof the head from the viewpoint of any other tail. For this reason, using shared keys to authenticate BFD Control packets in multipoint scenarios is a significant security exposure unless all tails can be trusted not to spoof the head. Otherwise, asymmetric message authentication would be needed, e.g., protocols that use Timed Efficient Stream Loss-Tolerant Authentication (TESLA) as described in [RFC4082]. Applicability of the asymmetric message authentication to BFD for multipoint networks is outside the scope of this specification and is for further study.

9. Contributors

Rahul Aggarwal of Juniper Networks and George Swallow of Cisco Systems provided the initial idea for this specification and contributed to its development.

10. Acknowledgments

Authors would also like to thank Nobo Akiya, Vengada Prasad Govindan, Jeff Haas, Wim Henderickx, Gregory Mirsky and Mingui Zhang who have greatly contributed to this document.

11. References

11.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [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>>.
- [RFC7880] Pignataro, C., Ward, D., Akiya, N., Bhatia, M., and S. Pallagatti, "Seamless Bidirectional Forwarding Detection (S-BFD)", RFC 7880, DOI 10.17487/RFC7880, July 2016, <<https://www.rfc-editor.org/info/rfc7880>>.
- [RFC8029] Kompella, K., Swallow, G., Pignataro, C., Ed., Kumar, N., Aldrin, S., and M. Chen, "Detecting Multiprotocol Label Switched (MPLS) Data-Plane Failures", RFC 8029, DOI 10.17487/RFC8029, March 2017, <<https://www.rfc-editor.org/info/rfc8029>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.

11.2. Informational References

- [I-D.ietf-bess-mvpn-fast-failover] Morin, T., Kebler, R., and G. Mirsky, "Multicast VPN fast upstream failover", draft-ietf-bess-mvpn-fast-failover-04 (work in progress), November 2018.

- [I-D.ietf-bfd-multipoint-active-tail]
Katz, D., Ward, D., Networks, J., and G. Mirsky, "BFD Multipoint Active Tails.", draft-ietf-bfd-multipoint-active-tail-10 (work in progress), November 2018.
- [RFC4082] Perrig, A., Song, D., Canetti, R., Tygar, J., and B. Briscoe, "Timed Efficient Stream Loss-Tolerant Authentication (TESLA): Multicast Source Authentication Transform Introduction", RFC 4082, DOI 10.17487/RFC4082, June 2005, <<https://www.rfc-editor.org/info/rfc4082>>.
- [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>>.
- [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>>.

Authors' Addresses

Dave Katz
Juniper Networks
1194 N. Mathilda Ave.
Sunnyvale, California 94089-1206
USA

Email: dkatz@juniper.net

Dave Ward
Cisco Systems
170 West Tasman Dr.
San Jose, California 95134
USA

Email: wardd@cisco.com

Santosh Pallagatti (editor)
Rtbrick

Email: santosh.pallagatti@gmail.com

Greg Mirsky (editor)
ZTE Corp.

Email: gregimirsky@gmail.com

Network Working Group
Internet-Draft
Updates: 5880 (if approved)
Intended status: Standards Track
Expires: 2 February 2022

M. Jethanandani
Kloud Services
A. Mishra
SES Networks
A. Saxena
Ciena Corporation
M. Bhatia
Nokia
1 August 2021

Optimizing BFD Authentication
draft-ietf-bfd-optimizing-authentication-13

Abstract

This document describes an optimization to BFD Authentication as described in Section 6.7 of BFD RFC 5880. This document updates RFC 5880.

Status of This Memo

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

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

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

This Internet-Draft will expire on 2 February 2022.

Copyright Notice

Copyright (c) 2021 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 (<https://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	2
1.1. Requirements Language	3
1.2. Terminology	3
2. Authentication Mode	4
3. NULL Auth Type	6
4. IANA Considerations	7
5. Security Considerations	7
6. References	7
6.1. Normative References	7
6.2. Informative References	7
Authors' Addresses	8

1. Introduction

Authenticating every BFD [RFC5880] control packet with a Simple Password, or with a MD5 Message-Digest Algorithm [RFC1321] , or Secure Hash Algorithm (SHA-1) algorithms is a computationally intensive process. This makes it difficult, if not impossible to authenticate every packet - particularly at faster rates. Also, the recent escalating series of attacks on MD5 and SHA-1 described in Finding Collisions in the Full SHA-1 [SHA-1-attack1] and New Collision Search for SHA-1 [SHA-1-attack2] raise concerns about their remaining useful lifetime as outlined in Updated Security Considerations for the MD5 Message-Digest and the HMAC-MD5 Algorithm [RFC6151] and Security Considerations for the SHA-0 and SHA-1 Message-Digest Algorithm [RFC6194]. If replaced by stronger algorithms, the computational overhead, will make the task of authenticating every packet even more difficult to achieve.

This document proposes that only BFD control packets that signal a state change, a demand mode change (to D bit) or a poll sequence change (P or F bit change) in a BFD control packet be categorized as a significant change. This document also proposes that all BFD control packets which signal a significant change MUST be authenticated if the session's bfd.AuthType is non-zero. Other BFD control packets MAY be transmitted and received without the A bit set.

Most packets that are transmitted and received have no state change associated with them. Limiting authentication to packets that affect a BFD session state allows more sessions to be supported with this optimized method of authentication. Moreover, most BFD control packets that signal a significant change are generally transmitted at a slower interval of 1s, leaving enough time to compute the hash.

To detect a Man In the Middle (MITM) attack, it is also proposed that a BFD control packet without a significant change be authenticated occasionally. The interval of the BFD control packets without a significant change can be configured depending on the detect multiplier and the capability of the system. As an example, this could be equal to the detect multiplier number of packets.

The rest of the document is structured as follows. Section 2 talks about the changes to authentication mode as described in BFD [RFC5880]. Section 3 goes into the details of the new Authentication Type.

1.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

1.2. Terminology

The following terms used in this document have been defined in BFD [RFC5880].

- * Detect Multiplier

- * Detection Time

The following terms are introduced in this document.

Term	Meaning
significant change	State change, a demand model change (to D bit) or a poll sequence change (P or F bit).
configured interval	Interval at which BFD control packets are authenticated in the UP state.

Table 1

2. Authentication Mode

The cryptographic authentication mechanisms specified in BFD [RFC5880] describes enabling and disabling of authentication as a one time operation. As a security precaution, it mentions that authentication state be allowed to change at most once. Once enabled, every packet must have Authentication Bit set and the associated Authentication Type appended. In addition, it states that an implementation SHOULD NOT allow the authentication state to be changed based on the receipt of a BFD control packet.

This document proposes that the authentication mode be modified to be enabled on demand. Instead of authenticating every packet, BFD peers are configured for which packets need to be authenticated, and authenticate only those packets. Rest of the packets can be transmitted and received without authentication. For example, the two ends can be configured such that BFD control packets that indicate a significant change should be authenticated and enable authentication on those packets only. If the two ends have previously been configured as such, but at least one side decides not to authenticate a significant change packet, then the BFD session will fail to come up.

This proposal outlines which BFD control packets need to be authenticated (carry the A-bit), and which packets can be transmitted or received without authentication enabled. A BFD control packet that fails authentication is discarded, or a BFD control packet that was supposed to be authenticated, but was not, e.g. a significant change packet, is discarded. However, there is no change to the state machine for BFD, as the decision of a significant change is still decided by how many valid consecutive packets were received, authenticated or otherwise.

The following table summarizes when the A bit should be set. The table should be read with the column indicating the BFD state the receiver is currently in, and the row indicating the BFD state the receiver might transition to based on the BFD control packet received. The intersection of the two indicates whether the received BFD control packet should have the A bit set (Auth), no authentication is needed (NULL), most packets are NULL AUTH (Select) or the state transition is not applicable. The BFD state refers to the states in BFD state machine described in Section 6.2 of BFD [RFC5880].

Read : On state change from <column> to <row>
 Auth : Authenticate BFD control packet
 NULL : No Authentication. Use NULL AUTH Type.
 n/a : Invalid state transition.
 Select : Most packets NULL AUTH. Selective (periodic) packets authenticated.

	DOWN	INIT	UP
DOWN	NULL	Auth	Auth
INIT	Auth	NULL	n/a
UP	Auth	Auth	Select

Figure 1: Optimized Authentication Map

If P or F bit changes value, the BFD control packet MUST be authenticated. If the D bit changes value, the BFD control packet MUST be authenticated.

All packets already carry the sequence number. The NULL AUTH packets MUST contain the Type specified in Section 3. This enables a monotonically increasing sequence number to be carried in each packet, and prevents man-in-the-middle from capturing and replaying the same packet again. Since all packets still carry a sequence number, the logic for sequence number maintenance remains unchanged from BFD [RFC5880]. If at a later time, a different scheme is adopted for changing sequence number, e.g. Secure BFD Sequence Numbers [I-D.ietf-bfd-secure-sequence-numbers], this method can use the updated scheme without any impact.

Most packets transmitted on a BFD session are BFD UP packets. Authenticating a small subset of these packets, for example, a detect multiplier number of packets per configured interval, significantly reduces the computational demand for the system while maintaining

security of the session across the configured interval. A minimum of Detect Multiplier packets MUST be transmitted per configured interval. This ensures that the BFD session should see at least one authenticated packet during that interval.

3. NULL Auth Type

This section describes a new Authentication Type as:

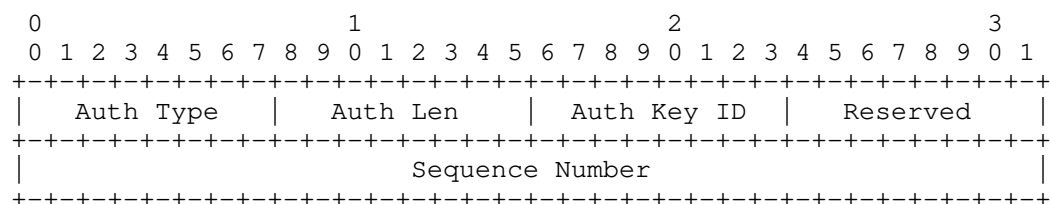


Figure 2: NULL Auth Type

where:

Auth Type: The Authentication Type, which in this case is TBD (NULL, to be assigned by IANA)

Auth Len: The length of the NULL Auth Type, in bytes i.e. 8 bytes

Auth Key ID: The authentication key ID in use for this packet. Must be set to zero.

Reserved: This byte MUST be set to zero on transmit and ignored on receive.

Sequence Number: The sequence number for this packet. Implementation may use sequence numbers (bfd.XmitAuthSeq) as defined in BFD [RFC5880], or secure sequence numbers as defined in Secure BFD Sequence Numbers [I-D.ietf-bfd-secure-sequence-numbers].

The NULL Auth Type must be used for all packets that are not authenticated. This protects against replay-attacks by allowing the session to maintain an incrementing sequence number for all packets (authenticated and un-authenticated).

In the future, if a new scheme is adopted for changing the sequence number, this method can adopt the new scheme without any impact.

4. IANA Considerations

This document requests an update to the registry titled "BFD Authentication Types". IANA is requested to assign a new BFD Auth Type for "NULL" (see Section 3).

Note to RFC Editor: this section may be removed on publication as an RFC.

5. Security Considerations

The approach described in this document enhances the ability to authenticate a BFD session by taking away the onerous requirement that every BFD control packet be authenticated. By authenticating packets that affect the state of the session, the security of the BFD session is maintained. In this mode, packets that are a significant change but are not authenticated, are dropped by the system. Therefore, a malicious user that tries to inject a non-authenticated packet, e.g. with a Down state to take a session down will fail. That combined with the proposal of using sequence number defined in Secure BFD Sequence Numbers [I-D.ietf-bfd-secure-sequence-numbers] further enhances the security of BFD sessions.

6. References

6.1. Normative References

- [I-D.ietf-bfd-secure-sequence-numbers]
Jethanandani, M., Agarwal, S., Mishra, A., Saxena, A., and A. DeKok, "Secure BFD Sequence Numbers", Work in Progress, Internet-Draft, draft-ietf-bfd-secure-sequence-numbers-08, 8 March 2021, <<https://www.ietf.org/archive/id/draft-ietf-bfd-secure-sequence-numbers-08.txt>>.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [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>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.

6.2. Informative References

- [RFC1321] Rivest, R., "The MD5 Message-Digest Algorithm", RFC 1321, DOI 10.17487/RFC1321, April 1992, <<https://www.rfc-editor.org/info/rfc1321>>.
- [RFC6151] Turner, S. and L. Chen, "Updated Security Considerations for the MD5 Message-Digest and the HMAC-MD5 Algorithms", RFC 6151, DOI 10.17487/RFC6151, March 2011, <<https://www.rfc-editor.org/info/rfc6151>>.
- [RFC6194] Polk, T., Chen, L., Turner, S., and P. Hoffman, "Security Considerations for the SHA-0 and SHA-1 Message-Digest Algorithms", RFC 6194, DOI 10.17487/RFC6194, March 2011, <<https://www.rfc-editor.org/info/rfc6194>>.
- [SHA-1-attack1]
Wang, X., Yin, Y., and H. Yu, "Finding Collisions in the Full SHA-1", 2005.
- [SHA-1-attack2]
Wang, X., Yao, A., and F. Yao, "New Collision Search for SHA-1", 2005.

Authors' Addresses

Mahesh Jethanandani
Kloud Services
United States of America

Email: mjethanandani@gmail.com

Ashesh Mishra
SES Networks

Email: mishra.ashesh@gmail.com

Ankur Saxena
Ciena Corporation
3939 N 1st Street
San Jose, CA 95134
United States of America

Email: ankurpsaxena@gmail.com

Manav Bhatia
Nokia
Bangalore
India

Email: manav.bhatia@nokia.com

Network Working Group
Internet-Draft
Updates: 5880 (if approved)
Intended status: Standards Track
Expires: 30 September 2022

M. Jethanandani
Kloud Services
S. Agarwal
Cisco Systems, Inc
A. Mishra
O3b Networks
A. Saxena
Ciena Corporation
A. Dekok
Network RADIUS SARL
29 March 2022

Secure BFD Sequence Numbers
draft-ietf-bfd-secure-sequence-numbers-09

Abstract

This document describes two new BFD Authentication mechanism, Meticulous Keyed ISAAC, and Meticulous Keyed FNV1A. These mechanisms can be used to authenticate BFD packets, and secure the sequence number exchange, with less CPU time cost than using MD5 or SHA1, with the tradeoff of decreased security. This document updates RFC 5880.

Status of This Memo

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

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

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

This Internet-Draft will expire on 30 September 2022.

Copyright Notice

Copyright (c) 2022 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 (<https://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 Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction	2
2. Requirements Language	3
3. Meticulous Keyed ISAAC	3
4. Meticulous Keyed FNV1A	5
5. Operation	6
5.1. Seeding and Operation of ISAAC	7
5.2. Secret Key	8
5.3. Seeding ISAAC	8
6. Meticulous Keyed ISAAC Authentication	9
7. Meticulous Keyed FNV1A Authentication	10
7.1. Calculation of the FNV-1a Digest	12
8. IANA Considerations	12
9. Security Considerations	13
9.1. Spoofing	14
9.2. Re-Use of keys	14
10. Acknowledgements	15
11. References	15
11.1. Normative References	15
11.2. Informative References	15
Authors' Addresses	15

1. Introduction

BFD [RFC5880] defines a number of authentication mechanisms, including Simple Password (Section 6.7.2), and various other methods based on MD5 and SHA1 hashes. The benefit of using cryptographic hashes is that they are secure. The downside to cryptographic hashes is that they are expensive and time consuming on resource-constrained hardware.

When BFD packets are unauthenticated, it is possible for an attacker to forge, modify, and/or replay packets on a link. These attacks have a number of side effects. They can cause parties to believe that a link is down, or they can cause parties to believe that the link is up when it is, in fact, down. The goal of these methods is to prevent spoofing of the BFD session by someone who could guess the next sequence number. We therefore define simple and fast Auth Type methods which allow parties to detect and prevent both spoofed sequence numbers, and spoofed packets.

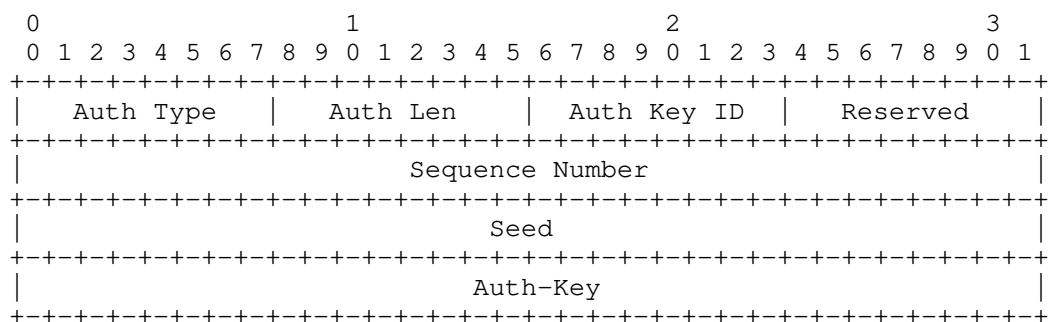
This document proposes the use of Authentication methods which provides meticulous keying, but which have less impact on resource constrained systems. The algorithms chosen are ISAAC [ISAAC], which is a fast cryptographic random number generator, and FNV-1a FNV1A [FNV1A] which is a fast (but non-cryptographic) hash. ISAAC has been subject to significant cryptanalysis in the past thirty years, and has not yet been broken. Similarly, FNV-1a is fast, and while not cryptographically secure, it is has good hashing properties.

2. 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].

3. Meticulous Keyed ISAAC

If the Authentication Present (A) bit is set in the header, and the State (Sta) field equals 3 (Up), and the Authentication Type field contains TB1 (Meticulous Keyed ISAAC), the Authentication Section has the following format:



Auth Type

The Authentication Type, which in this case is TB1 (Meticulous Keyed ISAAC). If the State (Sta) field value is not 3 (Up), then Meticulous Keyed ISAAC MUST NOT be used.

Auth Len

The length of the Authentication Section, in bytes. For Meticulous Keyed ISAAC authentication, the length is 16.

Auth Key ID

The authentication key ID in use for this packet. This allows multiple keys to be active simultaneously.

Reserved

This byte MUST be set to zero on transmit, and ignored on receipt.

Sequence Number

The sequence number for this packet. For Meticulous Keyed ISAAC Authentication, this value is incremented for each successive packet transmitted for a session. This provides protection against replay attacks.

Seed

A 32-bit (4 octet) seed which is used in conjunction with the shared key in order to configure and initialize the ISAAC pseudo-random-number-generator (PRNG). It is used to identify and distinguish "streams" of random numbers which are generated by ISAAC.

Auth-Key

This field carries the 32-bit (4 octet) ISAAC output which is associated with the Sequence Number. The ISAAC PRNG MUST be configured and initialized as given in section TBD.

Note that the Auth-Key here does not include any summary or hash of the packet. The packet itself is completely unauthenticated.

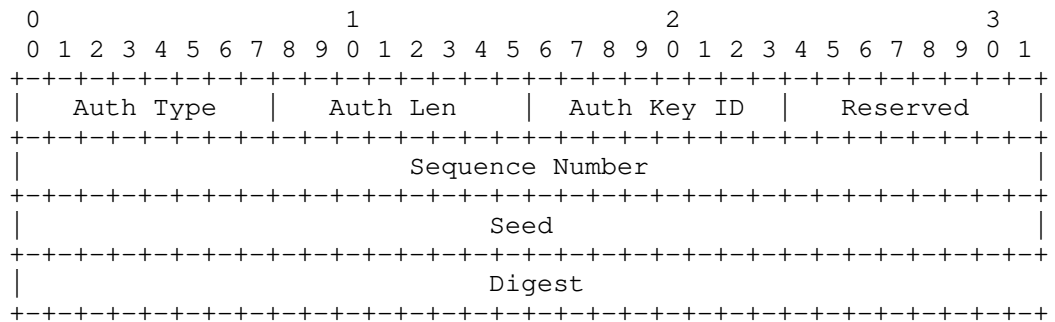
The purpose of this method is to secure the sequence number exchange, and to both detect and prevent spoofing of sequence numbers. In some cases, it is acceptable to not authenticate the entire packet, in which case this method may be used.

When the receiving party receives a BFD packet with an expected sequence number, and the correct corresponding ISAAC output, it knows that only the authentic sending party could have sent that message. The sending party is therefore alive/up, and intended to send the message.

While the rest of the contents of the BFD packet are unauthenticated and may be modified by an attacker, the same is true of stronger Auth Types, such as MD5 or SHA1. The Auth Type methods are not designed to prevent such attacks. Instead, they are designed to prevent an attacker from spoofing identities, and an attacker from artificially keeping a session "Up".

4. Meticulous Keyed FNV1A

If the Authentication Present (A) bit is set in the header, and the State (Sta) field equals 3 (Up), and the Authentication Type field contains TB2 (Meticulous Keyed FNV1A), the Authentication Section has the following format:



Auth Type

The Authentication Type, which in this case is TB2 (Meticulous Keyed FNV1A). If the State (Sta) field value is not 3 (Up), then Meticulous Keyed FNV1A MUST NOT be used.

Auth Len

The length of the Authentication Section, in bytes. For Meticulous Keyed FNV1A authentication, the length is 16.

Auth Key ID

The authentication key ID in use for this packet. This allows multiple keys to be active simultaneously.

Reserved

This byte MUST be set to zero on transmit, and ignored on receipt.

Sequence Number

The sequence number for this packet. For Meticulous Keyed FNV1A Authentication, this value is incremented for each successive packet transmitted for a session. This provides protection against replay attacks.

Seed

A 32-bit (4 octet) seed which is used in conjunction with the shared key in order to configure and initialize the ISAAC PRNG. It is also used to identify and distinguish "streams" of random numbers which are generated by ISAAC.

Digest

This field carries the 32-bit (4 octet) FNV1A digest associated with the Sequence Number. The ISAAC PRNG MUST be configured and initialized as given in section TBD.

Note that the ISAAC PRNG output is still used with this authentication type. The FNV1A hash is fast, but it is not secure. In order to reach an acceptable level of security with FNV1A, we use ISAAC to generate secure per-packet "signing keys". These per-packet keys are then used with FNV1A in order to perform a keyed of hash the packet, and therefore create the Digest.

5. Operation

BFD requires fast and reasonably secure authentication of messages which are exchanged. Methods using MD5 or SHA1 are CPU intensive, and can negatively impact systems with limited CPU power.

We use ISAAC here as a way to generate an infinite stream of pseudo-random numbers. With Meticulous Keyed ISAAC, these numbers are used as a signal that the sending party is authentic. That is, only the sending party can generate the numbers. Therefore if the receiving party sees a correct number, then only the sending party could have generated that number. The sender is therefore authentic, even if the packet contents are not necessarily trusted.

Note that since the packets are not signed with this authentication type, the Meticulous Keyed ISAAC method MUST NOT be used to signal BFD state changes. For BFD state changes, and a more optimized way

to authenticate packets, please refer to BFD Authentication [I-D.ietf-bfd-optimizing-authentication]. Instead, the packets containing Meticulous Keyed ISAAC are only a signal that the sending party is still alive, and that the sending party is authentic. That is, these Auth Type methods must only be used when `bfd.SessionState=Up`, and the State (Sta) field equals 3 (Up).

If slightly more security is desired, the packets can be authenticated via the Meticulous Keyed FNV1A method. This method is similar to the Meticulous Keyed ISAAC authentication type, except that the FNV-1A hash function is used to hash a combination of the packet, and per-packet ISAAC pseudo-random number. If the receiving party is able to validate the hash, then the receiver knows both that the sender is authentic, and that the packet contents have likely not been modified.

As this hash function is not very secure, this method can be used only in situations where the Meticulous Keyed ISAAC method can be used. The Meticulous Keyed FNV1A method MUST NOT be used to signal BFD state changes.

5.1. Seeding and Operation of ISAAC

The ISAAC PRNG state is initialized with the 32-bit Seed, followed by the secret key, and then the rest of the state is filled with zeros. The internal state of ISAAC is 1024 bytes, so the secret key is limited to 1020 bytes in length.

The origin of the Seed field is discussed later in this document. For now, we note that each time a new Seed is used, the `bfd.XmitAuthSeq` value MUST be set to zero.

Once the state has been initialized, the standard ISAAC initial mixing function is run. Once this operation has been performed, ISAAC will be able to produce 256 random numbers at near-zero cost. When all 256 numbers are consumed, the ISAAC mixing function is run, which then results in another set of 256 random numbers

ISAAC can be thought of here as producing an infinite stream of numbers, based on a secret key, where the numbers are produced in "pages" of 256 32-bit values. This property of ISAAC allows for essentially zero-cost "seeking" within a page. The expensive operation of mixing is performed only once per 256 packets, which means that most BFD packet exchanges can be fast and efficient.

The Sequence number is used to "seek" within a the stream of 32-bit numbers produced by ISAAC. The sending party increments the Sequence Number on every packet sent, to indicate to the receiving party where it is in the sequence.

The receiving party can then look at the Sequence Number to determine which particular PRNG value is being used in the packet. The Sequence Number thus permits the two parties to synchronise if/when a packet or packets are lost. Incrementing the Sequence Number for every packet also prevents the re-use of any individual pseudo-random number which was derived from ISAAC.

The Sequence Number can increment without bounds, though it can wrap once it reaches the limit of the 32-bit counter field. ISAAC has a cycle length of 2^{8287} , so there is no issue with using more than 2^{32} values from it.

The result of the above operation is an infinite series of numbers which are unguessable, and which can be used to authenticate the sending party.

5.2. Secret Key

For interoperability, the management interface by which the key is configured MUST accept ASCII strings, and SHOULD also allow for the configuration of any arbitrary binary string in hexadecimal form. Other configuration methods MAY be supported.

The secret Key is mixed with the Seed before being used in ISAAC. If instead ISAAC was initialized without a Seed, then an attacker could pre-compute ISAAC states for many keys, and perform an off-line dictionary attack. The addition of the Seed makes these attacks infeasible.

As a result, it is safe to use the same secret Key for the Auth Types defined here, and also for other Auth Types.

5.3. Seeding ISAAC

The value of the Seed field SHOULD be derived from a secure source. Exactly how this can be done is outside of the scope of this document.

The Seed value SHOULD remain the same for the duration of a BFD session. The Seed value MAY change when the BFD state changes.

If the sending party changes its Seed value, `bfd.XmitAuthSeq` value MUST be set to zero, otherwise the receiving party would be unable to synchronize its sequence of numbers produced by the ISAAC generator. There is no way to signal or negotiate Seed changes. The receiving party MUST remember the current Seed value, and then detect if the Seed changes. Note that the Seed value MUST NOT change unless sending party has signalled a BFD state change with a packet that is authenticated using a more secure Auth Type method.

6. Meticulous Keyed ISAAC Authentication

In this method of authentication, one or more secret keys (with corresponding key IDs) are configured in each system. One of the keys is used to seed the ISAAC PRNG. The output of ISAAC (I) is used to signal that the sender is authentic. To help avoid replay attacks, a sequence number is also carried in each packet. For Meticulous Keyed ISAAC, the sequence number is incremented on every packet.

The receiving system accepts the packet if the key ID matches one of the configured Keys, the Auth-Key derived from the selected Key, Seed, and Sequence Number matches the Auth-Key carried in the packet, and the sequence number is strictly greater than the last sequence number received (modulo wrap at 2^{32})

Transmission Using Meticulous Keyed ISAAC Authentication

The Auth Type field MUST be set to TBD1 (Meticulous Keyed ISAAC). The Auth Len field MUST be set to 16. The Auth Key ID field MUST be set to the ID of the current authentication key. The Sequence Number field MUST be set to `bfd.XmitAuthSeq`.

The Seed field MUST be set to the value of the current seed used for this sequence.

The Auth-Key field MUST be set to the output of ISAAC, which depends on the secret Key, the current Seed, and the Sequence Number.

For Meticulous Keyed ISAAC, `bfd.XmitAuthSeq` MUST be incremented on each packet, in a circular fashion (when treated as an unsigned 32-bit value). The `bfd.XmitAuthSeq` MUST NOT be incremented by more than one for a packet.

Receipt using Meticulous Keyed ISAAC Authentication

If the received BFD Control packet does not contain an Authentication Section, or the Auth Type is not correct (TBD2 for Meticulous Keyed ISAAC), then the received packet MUST be discarded.

If the Auth Key ID field does not match the ID of a configured authentication key, the received packet MUST be discarded.

If the Auth Len field is not equal to 16, the packet MUST be discarded.

If the Seed field does not match the current Seed value, the packet MUST be discarded.

If `bfd.AuthSeqKnown` is 1, examine the Sequence Number field. For Meticulous Keyed FNV1A, if the sequence number lies outside of the range of `bfd.RcvAuthSeq+1` to `bfd.RcvAuthSeq+(3*Detect Mult)` inclusive (when treated as an unsigned 32-bit circular number space) the received packet MUST be discarded.

Calculate the current expected output of ISAAC, which depends on the secret Key, the current Seed, and the Sequence Number. If the value does not matches the Auth-Key field, then the packet MUST be discarded.

Note that in some cases, calculating the expected output of ISAAC will result in the creation of a new "page" of 256 numbers. This process will irreversible, and will destroy the current "page". As a result, if the generation of a new output will create a new "page", the receiving party MUST save a copy of the entire ISAAC state before proceeding with this calculation. If the outputs match, then the saved copy can be discarded, and the new ISAAC state is used. If the outputs do not match, then the saved copy MUST be restored, and the modified copy discarded.

7. Meticulous Keyed FNV1A Authentication

Where slightly more security is needed, the sender can use Meticulous Keyed FNV1A. In this method, each packet is signed with a non-cryptographic hash, FNV-1a [FNV1A]. This hash is reasonably fast, it has good distribution, and collisions are rare. However, it is linear, and potentially reversible. In addition, its output is only 32 bits, and it is not cryptographically strong.

In this methods of authentication, one or more secret keys (with corresponding key IDs) are configured in each system. One of the keys is included in an FNV1A digest calculated over the outgoing BFD Control packet, but the Key itself is not carried in the packet. To

help avoid replay attacks, a sequence number is also carried in each packet. For Meticulous Keyed FNV1A, the sequence number is incremented on every packet.

The receiving system accepts the packet if the key ID matches one of the configured Keys, an FNV-1a digest including the selected key matches the digest carried in the packet, and the sequence number is strictly greater than the last sequence number received (modulo wrap at 2^{32})

Transmission Using Meticulous Keyed FNV1A Authentication

The Auth Type field MUST be set to TBD2 (Meticulous Keyed FNV1A). The Auth Len field MUST be set to 16. The Auth Key ID field MUST be set to the ID of the current authentication key. The Sequence Number field MUST be set to `bfd.XmitAuthSeq`.

The Digest field MUST be set to the value of the FNV-1a digest, as described below.

For Meticulous Keyed FNV1A, `bfd.XmitAuthSeq` MUST be incremented on each packet, in a circular fashion (when treated as an unsigned 32-bit value). The `bfd.XmitAuthSeq` MUST NOT be incremented by more than one for a packet.

Receipt Using Meticulous Keyed FNV1A Authentication

If the received BFD Control packet does not contain an Authentication Section, or the Auth Type is not correct (TBD2 for Meticulous Keyed FNV1A), then the received packet MUST be discarded.

If the Auth Key ID field does not match the ID of a configured authentication key, the received packet MUST be discarded.

If the Auth Len field is not equal to 16, the packet MUST be discarded.

If the Seed field does not match the current Seed value, the packet MUST be discarded.

If `bfd.AuthSeqKnown` is 1, examine the Sequence Number field. For Meticulous Keyed FNV1A, if the sequence number lies outside of the range of `bfd.RcvAuthSeq+1` to `bfd.RcvAuthSeq+(3*Detect Mult)` inclusive (when treated as an unsigned 32-bit circular number space) the received packet MUST be discarded.

Otherwise (bfd.AuthSeqKnown is 0), bfd.AuthSeqKnown MUST be set to 1, and bfd.RcvAuthSeq MUST be set to the value of the received Sequence Number field.

Replace the contents of the Digest field with zeros, and calculate the FNV-1a digest as described below. If the calculated FNV-1a digest is equal to the received value of the Digest field, the received packet MUST be accepted. Otherwise (the digest does not match the Digest field), the received packet MUST be discarded.

7.1. Calculation of the FNV-1a Digest

Unlike other authentication mechanisms, the user-supplied key is not placed into the Auth Key / Digest field, and the packet hashed. As FNV-1a is not a cryptographic hash, such a process would simplify the process for an attacker to "crack" the key.

Instead, for a particular packet "P", and ISAAC pseudo-random number "I", the FNV1A digest "D" is calculated as shown below, where "+" indicates concatenation.

$$D = \text{FNV1A}(I + P + I)$$

Where "+" denotes concatenation. We also note that the Digest field of the packet MUST be initialized to all zeroes before this calculation is performed

The calculated value "D" is then inserted into the packet in the Digest field, and the packet is sent as normal. The receiving party reverses this operation in order to validate the packet.

8. IANA Considerations

This document asks that IANA allocate a new entry in the "BFD Authentication Types" registry.

Address - TBD1

BFD Authentication Type Name - Meticulous Keyed ISAAC

Reference - this document

Address - TBD2

BFD Authentication Type Name - Meticulous Keyed FNV1A

Reference - this document

Note to RFC Editor: this section may be removed on publication as an RFC.

9. Security Considerations

The security of this proposal depends strongly on the length of the secret, and the entropy of the key. It is RECOMMENDED that the key be 16 octets in length or more.

The security of this proposal depends strongly on ISAAC. This generator has been analyzed and has not been broken. Research shows few other CSRNGs which are as simple and as fast as ISAAC. For example, many other generators are based on AES, which is infeasible for resource constrained systems.

The security of this proposal depends on the strength of the FNV-1a hash algorithm. Folding the output of ISAAC into the hash limits the ability of an attacker to reverse the hash, or to perform off-line dictionary attacks. Even if one particular 32-bit per-packet key is found via brute force, that information will be useless, as the next packet will use a different key. And since ISAAC is secure, knowledge of one particular key will give an attacker no ability to predict the next key.

In a keyed algorithm, the key is shared between the two systems. Distribution of this key to all the systems at the same time can be quite a cumbersome task. BFD sessions running a fast rate will require these keys to be refreshed often, which poses a further challenge. Therefore, it is difficult to change the keys during the operation of a BFD session without affecting the stability of the BFD session. Therefore, it is recommended to administratively disable the BFD session before changing the keys.

This method allows the BFD end-points to detect a malicious packet, as the calculated hash value will not match the value found in the packet. The behavior of the session, when such a packet is detected, is based on the implementation. A flood of such malicious packets may cause a BFD session to be operationally down.

As noted earlier with Meticulous Keyed FNV1A, each packet is associated with a unique, per-packet key. This process means that even if an observer sees the Auth-Key, or the FNV-1a hash for one packet, the only information gained will be a key which is never be re-used, and will therefore be useless to an attacker. Further, even if the attacker can "crack" a sequence of packets to obtain a stream of keys, the cryptographic nature of ISAAC makes it impossible for the attacker to derive the input key which is used to "seed" the ISAAC state.

The particular method of hashing was chosen because of the non-cryptographic and reversible nature of the FNV-1a hash. If the digest had been calculated any other way, then an attacker would have significantly less work to do in order to "crack" the hash. In short the per-packet key protects the hash, and the hash protects the per-packet key.

We believe that this construction is reasonably secure, given the constraints. If cryptographic security is desired, then implementors can use MD5 or SHA1 authentication mechanisms

9.1. Spoofing

When Meticulous Keyed ISAAC is used, it is possible for an attacker who can see the packets to observe a particular Auth Key value, and then copy it to a different packet as a "man-in-the-middle" attack. However, the usefulness of such an attack is limited by the requirements that these packets must not signal state changes in the BFD session, and that the key changes on every packet.

Performing such an attack would require an attacker to have the following information and capabilities:

- This is man-in-the-middle active attack.

- The attacker has the contents of a stable packet

- The attacker has managed to deduce the ISAAC key and knows which per-packet key is being used.

The attack is therefore limited to keeping the BFD session up when it would otherwise drop.

However, the usual actual attack which we are protecting BFD from is availability. That is, the attacker is trying to shut down then connection when the attacked parties are trying to keep it up. As a result, the attacks here seem to be irrelevant in practice.

9.2. Re-Use of keys

The strength of the Auth-Type methods is significantly different between the strong one like SHA-1 and ISAAC. While ISAAC has had cryptanalysis, and has not been shown to be broken, that analysis is limited. The question then is whether or not it is safe to use the same key for both Auth Type methods (SHA1 and ISAAC), or should we require different keys for each method?

If we recommend different keys, then it is possible for the two keys to be configured differently on each side of a BFD lin. For example. the strong key can be properly provisioned, which allows to the BFD state machine to advance to Up, Then, when we switch to the weaker Auth Type which uses a different key, that key may not match, and the session will immediatly drop.

We believe that the use of the same key is acceptable, as the Auth Types which use ISAAC also depend on a Seed. The use of the Seed increases the difficulty of breaking the key, and makes off-line dictionary attacks infeasible.

10. Acknowledgements

The authors would like to thank Jeff Haas and Reshad Rahman for their reviews of and suggestions for the document.

11. References

11.1. Normative References

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

11.2. Informative References

- [FNV1A] Noll, L. C., "FNV-1a", <http://www.isthe.com/chongo/tech/comp/fnv/index.html#FNV-1a>, 2013.
- [I-D.ietf-bfd-optimizing-authentication] Jethanandani, M., Mishra, A., Saxena, A., and M. Bhatia, "Optimizing BFD Authentication", Work in Progress, Internet-Draft, draft-ietf-bfd-optimizing-authentication-13, 1 August 2021, <<https://www.ietf.org/archive/id/draft-ietf-bfd-optimizing-authentication-13.txt>>.
- [ISAAC] Jenkins, R. J., "ISAAC", <http://www.burtleburtle.net/bob/rand/isaac.html>, 1996.

Authors' Addresses

Mahesh Jethanandani
Kloud Services
Email: mjethanandani@gmail.com

Sonal Agarwal
Cisco Systems, Inc
170 W. Tasman Drive
San Jose, CA 95070
United States of America
Email: agarwaso@cisco.com
URI: www.cisco.com

Ashesh Mishra
O3b Networks
Email: mishra.ashesh@gmail.com

Ankur Saxena
Ciena Corporation
3939 North First Street
San Jose, CA 95134
United States of America
Email: ankurpsaxena@gmail.com

Alan DeKok
Network RADIUS SARL
100 Centrepointe Drive #200
Ottawa ON K2G 6B1
Canada
Email: aland@freeradius.org

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: October 14, 2021

A. Mishra
SES
M. Jethanandani
Kloud Services
A. Saxena
Ciena Corporation
S. Pallagatti
VMware
M. Chen
Huawei
P. Fan
China Mobile
April 12, 2021

BFD Stability
draft-ietf-bfd-stability-10

Abstract

This document describes extensions to the Bidirectional Forwarding Detection (BFD) protocol to measure BFD stability. Specifically, it describes a mechanism for detection of BFD packet loss.

Status of This Memo

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

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

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

This Internet-Draft will expire on October 14, 2021.

Copyright Notice

Copyright (c) 2021 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

(<https://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	2
2. Terminology	3
3. Use Cases	3
4. BFD NULL-Authentication Type	3
5. Theory of Operation	3
5.1. Loss Measurement	4
6. ietf-bfd-stability YANG Module	4
6.1. Data Model Overview	4
6.2. YANG Module	5
7. IANA Considerations	9
7.1. The "IETF XML" Registry	9
7.2. The "YANG Module Names" Registry	9
8. Security Consideration	9
9. Contributors	10
10. Acknowledgements	10
11. References	10
11.1. Normative References	10
11.2. Informative References	12
Authors' Addresses	12

1. Introduction

The Bidirectional Forwarding Detection (BFD) [RFC5880] protocol operates by transmitting and receiving BFD control packets, generally at high frequency, over the datapath being monitored. In order to prevent significant data loss due to a datapath failure, BFD session detection time as defined in BFD [RFC5880] is set to the smallest feasible value.

This document proposes a mechanism to detect lost packets in a BFD session in addition to the datapath fault detection mechanisms of BFD. Such a mechanism presents significant value to measure the stability of BFD sessions and provides data to the operators for the cause of a BFD failure.

This document does not propose any BFD extension to measure data traffic loss or delay on a link or tunnel and the scope is limited to BFD packets.

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119] and RFC 8174 [RFC8174].

The reader is expected to be familiar with the BFD [RFC5880], Optimizing BFD Authentication [I-D.ietf-bfd-optimizing-authentication] and BFD Secure Sequence Numbers [I-D.ietf-bfd-secure-sequence-numbers].

3. Use Cases

Bidirectional Forwarding Detection as defined in BFD [RFC5880] cannot detect any BFD packet loss if the loss does not last for detection time. This document proposes a method to detect a dropped packet on the receiver. For example, if the receiver receives BFD control packet k at time t but receives packet k+3 at time t+10ms, and never receives packet k+1 and/or k+2, then it has experienced a drop.

This proposal enables BFD implementations to generate diagnostic information on the health of each BFD session that could be used to preempt a failure on a datapath that BFD was monitoring by allowing time for a corrective action to be taken.

In a faulty datapath scenario, an operator can use BFD health information to trigger delay and loss measurement OAM protocol, Connectivity Fault Management (CFM) [IEEE802.1ag] or Loss Measurement (LM)-Delay Measurement (DM)) as defined by A One-way Active Measurement Protocol (OWAMP) [RFC4656] to further isolate the issue.

4. BFD NULL-Authentication Type

The functionality proposed for BFD stability measurement is achieved by appending an authentication section with the NULL Authentication type (as defined in Optimizing BFD Authentication [I-D.ietf-bfd-optimizing-authentication]) to the BFD control packets that do not have authentication enabled.

5. Theory of Operation

This mechanism allows operators to measure the loss of BFD control packets.

When using MD5 or SHA authentication, BFD uses an authentication section that carries the Sequence Number. However, if non-meticulous authentication is being used, or no authentication is in use, then

the non-authenticated BFD control packets MUST include an authentication section with the NULL Authentication type.

5.1. Loss Measurement

Loss measurement counts the number of BFD control packets missed at the receiver during any Detection Time period. The loss is detected by comparing the Sequence Number field in the Auth TLV (NULL or otherwise) in successive BFD control packets. The Sequence Number in each successive control packet generated on a BFD session by the transmitter is incremented by one. This loss count can then be exposed using the YANG module defined in the subsequent section.

The first BFD authentication section with a non-zero sequence number, in a valid BFD control packet, processed by the receiver is used for bootstrapping the logic. When using secure sequence numbers, if the expected values are pre-calculated, the value must be matched to detect lost packets as defined in BFD secure sequence numbers [I-D.ietf-bfd-secure-sequence-numbers].

6. ietf-bfd-stability YANG Module

6.1. Data Model Overview

This YANG module augments the "ietf-bfd" module to add to the per-session set of counters a 'loss-packet-count' for BFD packets that are lost but do not necessarily result in the BFD session going down.

```

module: ietf-bfd-stability
  augment /rt:routing/rt:control-plane-protocols
    /rt:control-plane-protocol/bfd:bfd/bfd-ip-sh:ip-sh
    /bfd-ip-sh:sessions/bfd-ip-sh:session
    /bfd-ip-sh:session-statistics:
    +--ro lost-packet-count?   yang:counter32
  augment /rt:routing/rt:control-plane-protocols
    /rt:control-plane-protocol/bfd:bfd/bfd-ip-mh:ip-mh
    /bfd-ip-mh:session-groups/bfd-ip-mh:session-group
    /bfd-ip-mh:sessions/bfd-ip-mh:session-statistics:
    +--ro lost-packet-count?   yang:counter32
  augment /rt:routing/rt:control-plane-protocols
    /rt:control-plane-protocol/bfd:bfd/bfd-lag:lag
    /bfd-lag:sessions/bfd-lag:session/bfd-lag:member-links
    /bfd-lag:micro-bfd-ipv4/bfd-lag:session-statistics:
    +--ro lost-packet-count?   yang:counter32
  augment /rt:routing/rt:control-plane-protocols
    /rt:control-plane-protocol/bfd:bfd/bfd-lag:lag
    /bfd-lag:sessions/bfd-lag:session/bfd-lag:member-links
    /bfd-lag:micro-bfd-ipv6/bfd-lag:session-statistics:
    +--ro lost-packet-count?   yang:counter32
  augment /rt:routing/rt:control-plane-protocols
    /rt:control-plane-protocol/bfd:bfd/bfd-mpls:mpls
    /bfd-mpls:session-groups/bfd-mpls:session-group
    /bfd-mpls:sessions/bfd-mpls:session-statistics:
    +--ro lost-packet-count?   yang:counter32

```

6.2. YANG Module

This YANG module imports Common YANG Types [RFC6991], A YANG Data Model for Routing [RFC8349], and YANG Data Model for Bidirectional Forwarding Detection (BFD) [I-D.ietf-bfd-yang].

```

<CODE BEGINS> file "ietf-bfd-stability@2021-04-11.yang"
module ietf-bfd-stability {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-bfd-stability";
  prefix "bfds";

  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-bfd {
    prefix bfd;
    reference
      "I-D.ietf-bfd-yang: YANG Data Model for Bidirectional
        Forwarding Detection.";
  }

  import ietf-bfd-ip-sh {
    prefix bfd-ip-sh;
    reference
      "I-D.ietf-bfd-yang: YANG Data Model for Bidirectional
        Forwarding Detection.";
  }

  import ietf-bfd-ip-mh {
    prefix bfd-ip-mh;
    reference
      "I-D.ietf-bfd-yang: YANG Data Model for Bidirectional
        Forwarding Detection.";
  }

  import ietf-bfd-lag {
    prefix bfd-lag;
    reference
      "I-D.ietf-bfd-yang: YANG Data Model for Bidirectional
        Forwarding Detection.";
  }

  import ietf-bfd-mpls {
    prefix bfd-mpls;
    reference
      "I-D.ietf-bfd-yang: YANG Data Model for Bidirectional
        Forwarding Detection.";
  }

  organization
    "IETF BFD Working Group";

  contact
    "WG Web:  <http://tools.ietf.org/wg/bfd>
      WG List: <bfd@ietf.org>

      Authors: Mahesh Jethanandani (mjethanandani@gmail.com)
               Ashesh Mishra (mishra.ashesh@gmail.com)"
```


Ankur Saxena (ankurpsaxena@gmail.com)
Santosh Pallagatti (santosh.pallagatti@gmail.com)
Mach Chen (mach.chen@huawei.com)
Peng Fan (fanp08@gmail.com).";

description

"This YANG module augments the base BFD YANG model to add attributes related to BFD Stability. In particular it adds a per session count for BFD packets that are lost.

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 Simplified 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 XXXX (<https://www.rfc-editor.org/info/rfcXXXX>); see the RFC itself for full legal notices.

The key words 'MUST', 'MUST NOT', 'REQUIRED', 'SHALL', 'SHALL NOT', 'SHOULD', 'SHOULD NOT', 'RECOMMENDED', 'NOT RECOMMENDED', 'MAY', and 'OPTIONAL' in this document are to be interpreted as described in BCP 14 (RFC 2119) (RFC 8174) when, and only when, they appear in all capitals, as shown here.";

```
revision "2021-04-11" {  
  description  
    "Initial Version.";  
  reference  
    "RFC XXXX, BFD Stability.";  
}
```

```
augment "/rt:routing/rt:control-plane-protocols/" +  
  "rt:control-plane-protocol/bfd:bfd/bfd-ip-sh:ip-sh/" +  
  "bfd-ip-sh:sessions/bfd-ip-sh:session/" +  
  "bfd-ip-sh:session-statistics" {  
  leaf lost-packet-count {  
    type yang:counter32;  
    description  
      "Number of BFD packets that were lost without bringing the  
      session down.";  
  }  
}
```

```
    description
      "Augment the 'bfd' container to add attributes related to BFD
      stability.";
  }

  augment "/rt:routing/rt:control-plane-protocols/" +
    "rt:control-plane-protocol/bfd:bfd/bfd-ip-mh:ip-mh/" +
    "bfd-ip-mh:session-groups/bfd-ip-mh:session-group/" +
    "bfd-ip-mh:sessions/bfd-ip-mh:session-statistics" {
    leaf lost-packet-count {
      type yang:counter32;
      description
        "Number of BFD packets that were lost without bringing the
        session down.";
    }
    description
      "Augment the 'bfd' container to add attributes related to BFD
      stability.";
  }

  augment "/rt:routing/rt:control-plane-protocols/" +
    "rt:control-plane-protocol/bfd:bfd/bfd-lag:lag/" +
    "bfd-lag:sessions/bfd-lag:session/bfd-lag:member-links/" +
    "bfd-lag:micro-bfd-ipv4/bfd-lag:session-statistics" {
    leaf lost-packet-count {
      type yang:counter32;
      description
        "Number of BFD packets that were lost without bringing the
        session down.";
    }
    description
      "Augment the 'bfd' container to add attributes related to BFD
      stability.";
  }

  augment "/rt:routing/rt:control-plane-protocols/" +
    "rt:control-plane-protocol/bfd:bfd/bfd-lag:lag/" +
    "bfd-lag:sessions/bfd-lag:session/bfd-lag:member-links/" +
    "bfd-lag:micro-bfd-ipv6/bfd-lag:session-statistics" {
    leaf lost-packet-count {
      type yang:counter32;
      description
        "Number of BFD packets that were lost without bringing the
        session down.";
    }
    description
      "Augment the 'bfd' container to add attributes related to BFD
      stability.";
```

```
    }  
  
    augment "/rt:routing/rt:control-plane-protocols/" +  
        "rt:control-plane-protocol/bfd:bfd/bfd-mpls:mpls/" +  
        "bfd-mpls:session-groups/bfd-mpls:session-group/" +  
        "bfd-mpls:sessions/bfd-mpls:session-statistics" {  
        leaf lost-packet-count {  
            type yang:counter32;  
            description  
                "Number of BFD packets that were lost without bringing the  
                session down.";  
        }  
        description  
            "Augment the 'bfd' container to add attributes related to BFD  
            stability.";  
    }  
}  
<CODE ENDS>
```

7. IANA Considerations

7.1. The "IETF XML" Registry

This document registers one URIs in the "ns" subregistry of the "IETF XML" registry [RFC3688]. Following the format in [RFC3688], the following registration is requested:

URI: urn:ietf:params:xml:ns:yang:ietf-bfd-stability
Registrant Contact: The IESG
XML: N/A, the requested URI is an XML namespace.

7.2. The "YANG Module Names" Registry

This document registers one YANG module in the "YANG Module Names" registry YANG [RFC6020]. Following the format in YANG [RFC6020], the following registrations are requested:

name: ietf-bfd-stability
namespace: urn:ietf:params:xml:ns:yang:ietf-bfd-stability
prefix: bfds
reference: RFC XXXX

8. Security Consideration

The YANG module specified in this document defines a schema for data that is designed to be accessed via network management protocols such as NETCONF [RFC6241] or RESTCONF [RFC8040]. The lowest NETCONF layer is the secure transport layer, and the mandatory-to-implement secure

transport is Secure Shell (SSH) [RFC6242]. The lowest RESTCONF layer is HTTPS, and the mandatory-to-implement secure transport is TLS [RFC8446]. The NETCONF 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.

The YANG module does not define any writeable/creatable/deletable data nodes.

The only readable data nodes in YANG module may be considered sensitive or vulnerable in some network environments. It is thus important to control read access (e.g., via get, get-config, or notification) to these data nodes. The model does not define any readable subtrees and data nodes.

The YANG module does not define any RPC operations.

9. Contributors

Manav Bhatia

10. Acknowledgements

Authors would like to thank Nobo Akiya, Jeffery Haas, Dileep Singh, Basil Saji, Sagar Soni, Albert Fu and Mallik Mudigonda who also contributed to this document.

11. References

11.1. Normative References

[I-D.ietf-bfd-optimizing-authentication]

Jethanandani, M., Mishra, A., Saxena, A., and M. Bhatia, "Optimizing BFD Authentication", draft-ietf-bfd-optimizing-authentication-11 (work in progress), July 2020.

[I-D.ietf-bfd-secure-sequence-numbers]

Jethanandani, M., Agarwal, S., Mishra, A., Saxena, A., and A. DeKok, "Secure BFD Sequence Numbers", draft-ietf-bfd-secure-sequence-numbers-07 (work in progress), December 2020.

- [I-D.ietf-bfd-yang]
Rahman, R., Zheng, L., Jethanandani, M., Pallagatti, S.,
and G. Mirsky, "YANG Data Model for Bidirectional
Forwarding Detection (BFD)", draft-ietf-bfd-yang-17 (work
in progress), August 2018.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate
Requirement Levels", BCP 14, RFC 2119,
DOI 10.17487/RFC2119, March 1997,
<<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688,
DOI 10.17487/RFC3688, January 2004,
<<https://www.rfc-editor.org/info/rfc3688>>.
- [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>>.
- [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>>.
- [RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF
Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017,
<<https://www.rfc-editor.org/info/rfc8040>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC
2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174,
May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.

- [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>>.
- [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>>.

11.2. Informative References

- [IEEE802.1ag] Institute of Electrical and Electronics Engineers, Inc., "802.1ag - Connectivity Fault Management", September 2007, <<https://www.ieee802.org/1/pages/802.1ag.html>>.
- [RFC4656] Shalunov, S., Teitelbaum, B., Karp, A., Boote, J., and M. Zekauskas, "A One-way Active Measurement Protocol (OWAMP)", RFC 4656, DOI 10.17487/RFC4656, September 2006, <<https://www.rfc-editor.org/info/rfc4656>>.

Authors' Addresses

Ashesh Mishra
SES

Email: mishra.ashesh@gmail.com

Mahesh Jethanandani
Kloud Services
CA
USA

Email: mjethanandani@gmail.com

Ankur Saxena
Ciena Corporation
3939 North 1st Street
San Jose, CA 95134
USA

Email: ankurpsaxena@gmail.com
URI: www.ciena.com

Santosh Pallagatti
VMware
Bangalore, Karnataka 560103
India

Email: santosh.pallagatti@gmail.com

Mach Chen
Huawei

Email: mach.chen@huawei.com

Peng Fan
China Mobile
32 Xuanwumen West Street
Beijing, Beijing
China

Email: fanp08@gmail.com

BFD
Internet-Draft
Intended status: Informational
Expires: April 29, 2021

S. Pallagatti, Ed.
VMware
G. Mirsky, Ed.
ZTE Corp.
S. Paragiri
Individual Contributor
V. Govindan
M. Mudigonda
Cisco
October 26, 2020

BFD for VXLAN
draft-ietf-bfd-vxlan-16

Abstract

This document describes the use of the Bidirectional Forwarding Detection (BFD) protocol in point-to-point Virtual eXtensible Local Area Network (VXLAN) tunnels used to form an overlay network.

Status of This Memo

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

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

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

This Internet-Draft will expire on April 29, 2021.

Copyright Notice

Copyright (c) 2020 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 (<https://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	2
2. Conventions Used in this Document	3
2.1. Acronyms	3
2.2. Requirements Language	4
3. Deployment	4
4. Use of the Management VNI	5
5. BFD Packet Transmission over VXLAN Tunnel	6
6. Reception of BFD Packet from VXLAN Tunnel	8
7. Echo BFD	8
8. IANA Considerations	8
9. Security Considerations	9
10. Contributors	9
11. Acknowledgments	9
12. References	10
12.1. Normative References	10
12.2. Informational References	10
Authors' Addresses	11

1. Introduction

"Virtual eXtensible Local Area Network" (VXLAN) [RFC7348] provides an encapsulation scheme that allows building an overlay network by decoupling the address space of the attached virtual hosts from that of the network.

One use of VXLAN is in data centers interconnecting virtual machines (VMs) of a tenant. VXLAN addresses requirements of the Layer 2 and Layer 3 data center network infrastructure in the presence of VMs in a multi-tenant environment by providing a Layer 2 overlay scheme on a Layer 3 network [RFC7348]. Another use is as an encapsulation for Ethernet VPN [RFC8365].

This document is written assuming the use of VXLAN for virtualized hosts and refers to VMs and VXLAN Tunnel End Points (VTEPs) in hypervisors. However, the concepts are equally applicable to non-virtualized hosts attached to VTEPs in switches.

In the absence of a router in the overlay, a VM can communicate with another VM only if they are on the same VXLAN segment. VMs are unaware of VXLAN tunnels as a VXLAN tunnel is terminated on a VTEP.

VTEPs are responsible for encapsulating and decapsulating frames exchanged among VMs.

The ability to monitor path continuity, i.e., perform proactive continuity check (CC) for point-to-point (p2p) VXLAN tunnels, is important. The asynchronous mode of BFD, as defined in [RFC5880], is used to monitor a p2p VXLAN tunnel.

In the case where a Multicast Service Node (MSN) (as described in Section 3.3 of [RFC8293]) participates in VXLAN, the mechanisms described in this document apply and can, therefore, be used to test the continuity of the path between the source NVE and the MSN.

This document describes the use of Bidirectional Forwarding Detection (BFD) protocol to enable monitoring continuity of the path between VXLAN VTEPs that are performing as Network Virtualization Endpoints, and/or between the source NVE and a replicator MSN using a Management VNI (Section 4). All other uses of the specification to test toward other VXLAN endpoints are out of the scope.

2. Conventions Used in this Document

2.1. Acronyms

BFD Bidirectional Forwarding Detection

CC Continuity Check

p2p Point-to-point

MSN Multicast Service Node

NVE Network Virtualization Endpoint

VFI Virtual Forwarding Instance

VM Virtual Machine

VNI VXLAN Network Identifier (or VXLAN Segment ID)

VTEP VXLAN Tunnel End Point

VXLAN Virtual eXtensible Local Area Network

2.2. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

3. Deployment

Figure 1 illustrates the scenario with two servers, each of them hosting two VMs. The servers host VTEPs that terminate two VXLAN tunnels with VXLAN Network Identifier (VNI) number 100 and 200 respectively. Separate BFD sessions can be established between the VTEPs (IP1 and IP2) for monitoring each of the VXLAN tunnels (VNI 100 and 200). Using a BFD session to monitor a set of VXLAN VNIs between the same pair of VTEPs might help to detect and localize problems caused by misconfiguration. An implementation that supports this specification MUST be able to control the number of BFD sessions that can be created between the same pair of VTEPs. This method is applicable whether the VTEP is a virtual or physical device.

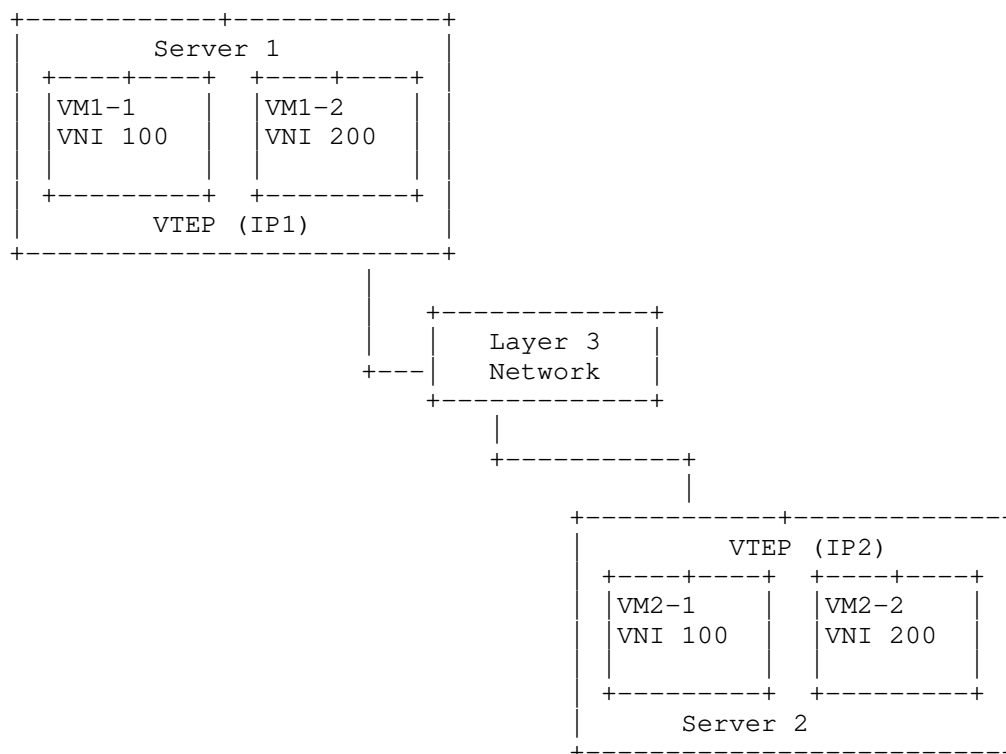


Figure 1: Reference VXLAN Domain

At the same time, a service layer BFD session may be used between the tenants of VTEPs IP1 and IP2 to provide end-to-end fault management (this use case is outside the scope of this document). In such a case, for VTEPs, the BFD Control packets of that session are indistinguishable from data packets.

For BFD Control packets encapsulated in VXLAN (Figure 2), the inner destination IP address SHOULD be set to one of the loopback addresses from 127/8 range for IPv4 or to one of IPv4-mapped IPv6 loopback addresses from `::ffff:127.0.0.0/104` range for IPv6.

4. Use of the Management VNI

In most cases, a single BFD session is sufficient for the given VTEP to monitor the reachability of a remote VTEP, regardless of the number of VNIs. BFD control messages MUST be sent using the Management VNI which acts as the control and management channel between VTEPs. An implementation MAY support operating BFD on

another (non-Management) VNI although the implications of this are outside the scope of this document. The selection of the VNI number of the Management VNI MUST be controlled through a management plane. An implementation MAY use VNI number 1 as the default value for the Management VNI. All VXLAN packets received on the Management VNI MUST be processed locally and MUST NOT be forwarded to a tenant.

5. BFD Packet Transmission over VXLAN Tunnel

BFD packets MUST be encapsulated and sent to a remote VTEP as explained in this section. Implementations SHOULD ensure that the BFD packets follow the same forwarding path as VXLAN data packets within the sender system.

BFD packets are encapsulated in VXLAN as described below. The VXLAN packet format is defined in Section 5 of [RFC7348]. The value in the VNI field of the VXLAN header MUST be set to the value selected as the Management VNI. The Outer IP/UDP and VXLAN headers MUST be encoded by the sender as defined in [RFC7348].

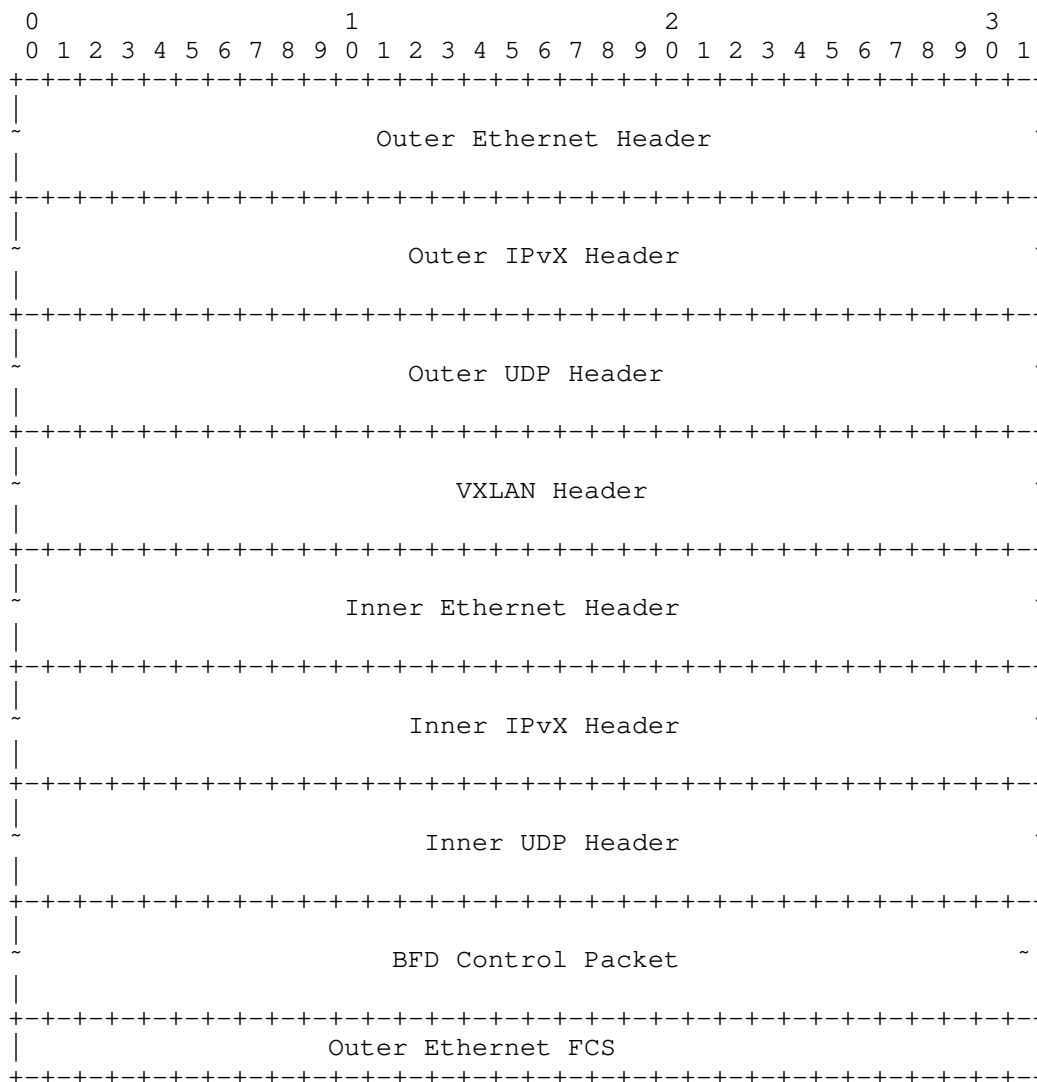


Figure 2: VXLAN Encapsulation of BFD Control Packet

The BFD packet MUST be carried inside the inner Ethernet frame of the VXLAN packet. The choice of Destination MAC and Destination IP addresses for the inner Ethernet frame MUST ensure that the BFD Control packet is not forwarded to a tenant but is processed locally at the remote VTEP. The inner Ethernet frame carrying the BFD Control packet- has the following format:

Ethernet Header:

Destination MAC: A Management VNI, which does not have any tenants, will have no dedicated MAC address for decapsulated traffic. The value (TBD1) SHOULD be used in this field.

Source MAC: MAC address associated with the originating VTEP.

Ethertype: is set to 0x0800 if the inner IP header is IPv4, and is set to 0x86DD if the inner IP header is IPv6.

IP header:

Destination IP: IP address MUST NOT be of one of tenant's IP addresses. The IP address SHOULD be selected from the range 127/8 for IPv4, for IPv6 - from the range ::ffff:127.0.0.0/104. Alternatively, the destination IP address MAY be set to VTEP's IP address.

Source IP: IP address of the originating VTEP.

TTL or Hop Limit: MUST be set to 255 in accordance with [RFC5881].

The fields of the UDP header and the BFD Control packet are encoded as specified in [RFC5881].

6. Reception of BFD Packet from VXLAN Tunnel

Once a packet is received, the VTEP MUST validate the packet. If the packet is received on the management VNI and is identified as BFD control packet addressed to the VTEP, and then the packet can be processed further. Processing of BFD control packets received on non-management VNI is outside the scope of this specification.

The received packet's inner IP payload is then validated according to Sections 4 and 5 in [RFC5881].

7. Echo BFD

Support for echo BFD is outside the scope of this document.

8. IANA Considerations

IANA is requested to assign a single MAC address to the value TBD1 from the "IANA Unicast 48-bit MAC Address" registry from the "Unassigned (small allocations)" block. The Usage field will be "BFD for VXLAN" with a Reference field of this document.

9. Security Considerations

Security issues discussed in [RFC5880], [RFC5881], and [RFC7348] apply to this document.

This document recommends using an address from the Internal host loopback addresses 127/8 range for IPv4 or an IP4-mapped IPv6 loopback address from ::ffff:127.0.0.0/104 range for IPv6 as the destination IP address in the inner IP header. Using such an address prevents the forwarding of the encapsulated BFD control message by a transient node in case the VXLAN tunnel is broken as according to [RFC1812].

A router SHOULD NOT forward, except over a loopback interface, any packet that has a destination address on network 127. A router MAY have a switch that allows the network manager to disable these checks. If such a switch is provided, it MUST default to performing the checks.

The use of IPv4-mapped IPv6 addresses has the same property as using the IPv4 network 127/8, moreover, the IPv4-mapped IPv6 addresses prefix is not advertised in any routing protocol.

If the implementation supports establishing multiple BFD sessions between the same pair of VTEPs, there SHOULD be a mechanism to control the maximum number of such sessions that can be active at the same time.

10. Contributors

Reshad Rahman
rrahman@cisco.com
Cisco

11. Acknowledgments

Authors would like to thank Jeff Haas of Juniper Networks for his reviews and feedback on this material.

Authors would also like to thank Nobo Akiya, Marc Binderberger, Shahram Davari, Donald E. Eastlake 3rd, Anoop Ghanwani, Dinesh Dutt, Joel Halpern, and Carlos Pignataro for the extensive reviews and the most detailed and constructive comments.

12. References

12.1. Normative References

- [RFC1812] Baker, F., Ed., "Requirements for IP Version 4 Routers", RFC 1812, DOI 10.17487/RFC1812, June 1995, <<https://www.rfc-editor.org/info/rfc1812>>.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [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>>.
- [RFC7348] Mahalingam, M., Dutt, D., Duda, K., Agarwal, P., Kreeger, L., Sridhar, T., Bursell, M., and C. Wright, "Virtual eXtensible Local Area Network (VXLAN): A Framework for Overlaying Virtualized Layer 2 Networks over Layer 3 Networks", RFC 7348, DOI 10.17487/RFC7348, August 2014, <<https://www.rfc-editor.org/info/rfc7348>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.

12.2. Informational References

- [RFC8293] Ghanwani, A., Dunbar, L., McBride, M., Bannai, V., and R. Krishnan, "A Framework for Multicast in Network Virtualization over Layer 3", RFC 8293, DOI 10.17487/RFC8293, January 2018, <<https://www.rfc-editor.org/info/rfc8293>>.
- [RFC8365] Sajassi, A., Ed., Drake, J., Ed., Bitar, N., Shekhar, R., Uttaro, J., and W. Henderickx, "A Network Virtualization Overlay Solution Using Ethernet VPN (EVPN)", RFC 8365, DOI 10.17487/RFC8365, March 2018, <<https://www.rfc-editor.org/info/rfc8365>>.

Authors' Addresses

Santosh Pallagatti (editor)
VMware

Email: santosh.pallagatti@gmail.com

Greg Mirsky (editor)
ZTE Corp.

Email: gregimirsky@gmail.com

Sudarsan Paragiri
Individual Contributor

Email: sudarsan.225@gmail.com

Vengada Prasad Govindan
Cisco

Email: venggovi@cisco.com

Mallik Mudigonda
Cisco

Email: mmudigon@cisco.com

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

R. Rahman, Ed.
Cisco Systems
L. Zheng, Ed.
Huawei Technologies
M. Jethanandani, Ed.
Xoriant Corporation
S. Pallagatti
Rtbrick
G. Mirsky
ZTE Corporation
August 1, 2018

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

Abstract

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

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

Status of This Memo

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

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

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

This Internet-Draft will expire on February 2, 2019.

Copyright Notice

Copyright (c) 2018 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

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

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

1. Introduction

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

- 1) Two systems directly connected via IP. This is known as BFD over single-hop IP, a.k.a. BFD for IPv4 and IPv6 [RFC5881]
- 2) Two systems connected via multiple hops as described in BFD for Multiple Hops. [RFC5883]
- 3) Two systems connected via MPLS Label Switched Paths (LSPs) as described in BFD for MPLS LSP [RFC5884]
- 4) Two systems connected via a Link Aggregation Group (LAG) interface as described in BFD on LAG Interfaces [RFC7130]

5) Two systems connected via pseudowires (PWs), this is known as Virtual Circuit Connectivity Verification (VCCV) as described in BFD for PW VCCV [RFC5885]. This is not addressed in this document.

BFD typically does not operate on its own. Various control protocols, also known as BFD clients, use the services provided by BFD for their own operation as described in Generic Application of BFD [RFC5882]. The obvious candidates which use BFD are those which do not have "hellos" to detect failures, e.g. static routes, and routing protocols whose "hellos" do not support sub-second failure detection, e.g. OSPF and IS-IS.

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

1.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

1.2. Tree Diagrams

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

2. Design of the Data Model

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

A new control-plane protocol "bfdv1" is defined and a "bfd" container is created under control-plane-protocol as specified in "A YANG Data Model for Routing Management (NMDA Version)" [RFC8349]. This new "bfd" container is augmented by all the YANG modules for their respective specific information:

1. ietf-bfd-ip-sh.yang augments "/routing/control-plane-protocols/control-plane-protocol/bfd/" with the "ip-sh" container for BFD sessions over IP single-hop.
2. ietf-bfd-ip-mh.yang augments "/routing/control-plane-protocols/control-plane-protocol/bfd/" with the "ip-mh" container for BFD sessions over IP multi-hop.
3. ietf-bfd-lag.yang augments "/routing/control-plane-protocols/control-plane-protocol/bfd/" with the "lag" container for BFD sessions over LAG.
4. ietf-bfd-mpls.yang augments "/routing/control-plane-protocols/control-plane-protocol/bfd/" with the "mpls" container for BFD over MPLS LSPs.
5. ietf-bfd-mpls-te.yang augments "/routing/control-plane-protocols/control-plane-protocol/bfd/" with the "mpls-te" container for BFD over MPLS-TE.

BFD can operate in the following contexts:

1. At the network device level
2. In Logical Network Elements as described in YANG Logical Network Element [I-D.ietf-rtgwg-lne-model]
3. In Network Instances as described in YANG Logical Network Element [I-D.ietf-rtgwg-ni-model]

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

2.1. Design of Configuration Model

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

BFD clients are applications that use BFD for fast detection of failures. Some implementations have BFD session configuration under the BFD clients. For example, BFD session configuration under routing applications such as OSPF, IS-IS, BGP etc. Other

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

The BFD parameters of interest to a BFD client are mainly the multiplier and interval(s) since those parameters impact the convergence time of the BFD clients when a failure occurs. Other parameters such as BFD authentication are not specific to the requirements of the BFD client. Ideally all configuration should be centralized under BFD. However, this is a problem for clients of BFD which auto-discover their peers. For example, IGPs do not have the peer address configured, instead the IGP is enabled on an interface and the IGP peers are auto-discovered. So for an operator to configure BFD to an IGP peer, the operator would first have to determine the peer addresses. And when a new peer is discovered, BFD configuration would need to be added. To avoid this issue, we define grouping client-cfg-parms in Section 2.13 for BFD clients to configure BFD: this allows BFD clients such as the IGPs to have configuration (multiplier and intervals) for the BFD sessions they need. For example, when a new IGP peer is discovered, the IGP would create a BFD session to the newly discovered peer and similarly when an IGP peer goes away, the IGP would remove the BFD session to that peer. The mechanism how the BFD sessions are created and removed by the BFD clients is outside the scope of this document, but typically this would be done by use of an API implemented by the BFD module on the system. For BFD clients which create BFD sessions via their own configuration, authentication parameters (if required) are still specified in BFD.

2.1.1.1. Common BFD configuration parameters

The basic BFD configuration parameters are:

local-multiplier

This is the detection time multiplier as defined in BFD [RFC5880].

desired-min-tx-interval

This is the Desired Min TX Interval as defined in BFD [RFC5880].

required-min-rx-interval

This is the Required Min RX Interval as defined in BFD [RFC5880].

Although BFD [RFC5880] allows for different values for transmit and receive intervals, some implementations allow users to specify just one interval which is used for both transmit and receive intervals or separate values for transmit and receive intervals. The BFD YANG

model supports this: there is a choice between "min-interval", used for both transmit and receive intervals, and "desired-min-tx-interval" and "required-min-rx-interval". This is supported via a grouping which is used by the YANG modules for the various forwarding paths.

For BFD authentication we have:

key-chain

This is a reference to key-chain defined in YANG Data Model for Key Chains [RFC8177]. The keys, cryptographic algorithms, key lifetime etc are all defined in the key-chain model.

meticulous

This enables meticulous mode as per BFD [RFC5880].

2.1.2. Single-hop IP

For single-hop IP, there is an augment of the "bfd" data node in Section 2. The "ip-sh" node contains a list of IP single-hop sessions where each session is uniquely identified by the interface and destination address pair. For the configuration parameters we use what is defined in Section 2.1.1. The "ip-sh" node also contains a list of interfaces, this is used to specify authentication parameters for BFD sessions which are created by BFD clients, see Section 2.1.

[RFC5880] and [RFC5881] do not specify whether echo function is continuous or on demand. Therefore the mechanism used to start and stop echo function is implementation specific and should be done by augmentation:

1) Configuration. This is suitable for continuous echo function. An example is provided in Appendix A.

2) RPC. This is suitable for on-demand echo function.

2.1.3. Multihop IP

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

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

source address

Address belonging to the local system as per BFD for Multiple Hops [RFC5883]

destination address

Address belonging to the remote system as per BFD for Multiple Hops [RFC5883]

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

Here are some extra parameters:

tx-ttl

TTL of outgoing BFD control packets.

rx-ttl

Minimum TTL of incoming BFD control packets.

2.1.4. MPLS Traffic Engineering Tunnels

For MPLS-TE tunnels, BFD is configured under the MPLS-TE tunnel since the desired failure detection parameters are a property of the MPLS-TE tunnel. This is achieved by augmenting the MPLS-TE data model in YANG Data Model for TE Topologies [I-D.ietf-teas-yang-te]. For BFD parameters which are specific to the TE application, e.g. whether to tear down the tunnel in the event of a BFD session failure, these parameters will be defined in the YANG model of the MPLS-TE application.

On top of the usual BFD parameters, we have the following per MPLS-TE tunnel:

encap

Encapsulation for the BFD packets: choice between IP, G-ACh and IP with G-ACh as per MPLS Generic Associated Channel [RFC5586]

For general MPLS-TE data, "mpls-te" data node is added under the "bfd" node in Section 2. Since some MPLS-TE tunnels are uni-directional there is no MPLS-TE configuration for these tunnels on the egress node (note that this does not apply to bi-directional MPLS-TP tunnels). The BFD parameters for the egress node are added under "mpls-te".

2.1.5. MPLS Label Switched Paths

Here we address MPLS LSPs whose FEC is an IP address. The "bfd" node in Section 2 is augmented with "mpls" which contains a list of sessions uniquely identified by an IP prefix. Because of multiple paths, there could be multiple MPLS sessions to an MPLS FEC. We identify this as a "session-group".

Since these LSPs are uni-directional there is no LSP configuration on the egress node.

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

2.1.6. Link Aggregation Groups

Per BFD on LAG Interfaces [RFC7130], configuring BFD on LAG consists of having micro-BFD sessions on each LAG member link. Since the BFD parameters are an attribute of the LAG, they should be under the LAG. However there is no LAG YANG model which we can augment. So a "lag" data node is added to the "bfd" node in Section 2, the configuration is per-LAG: we have a list of LAGs. The destination IP address of the micro-BFD sessions is configured per-LAG and per address-family (IPv4 and IPv6)

2.2. Design of Operational State Model

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

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

For each BFD session, mainly three categories of operational state data are shown. The fundamental information of a BFD session such as the local discriminator, remote discriminator and the capability of supporting demand detect mode are shown in the first category. The second category includes a BFD session running information, e.g. the remote BFD state and the diagnostic code received. Another example is the actual transmit interval between the control packets, which may be different from the desired minimum transmit interval configured, is shown in this category. Similar examples are actual received interval between the control packets and the actual transmit interval between the echo packets. The third category contains the detailed statistics of the session, e.g. when the session transitioned up/down and how long it has been in that state.

For some path types, there may be more than 1 session on the virtual path to the destination. For example, with IP multihop and MPLS LSPs, there could be multiple BFD sessions from the source to the same destination to test the various paths (ECMP) to the destination. This is represented by having multiple "sessions" under each "session-group".

2.3. Notifications

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

2.4. RPC Operations

None.

2.5. BFD top level hierarchy

At the "bfd" node under control-plane-protocol, there is no configuration data, only operational state data. The operational state data consist of overall BFD session statistics, i.e. for BFD on all types of forwarding paths.

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

2.6. BFD IP single-hop hierarchy

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

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

```

augment /rt:routing/rt:control-plane-protocols
  /rt:control-plane-protocol/bfd:bfd:
  +--rw ip-sh
    +--ro summary
      +--ro number-of-sessions?          yang:gauge32
      +--ro number-of-sessions-up?       yang:gauge32
      +--ro number-of-sessions-down?     yang:gauge32
      +--ro number-of-sessions-admin-down? yang:gauge32
    +--rw sessions
      +--rw session* [interface dest-addr]
        +--rw interface                  if:interface-ref
        +--rw dest-addr                  inet:ip-address
        +--rw source-addr?               inet:ip-address
        +--rw local-multiplier?          multiplier
        +--rw (interval-config-type)?
          +--:(tx-rx-intervals)
            +--rw desired-min-tx-interval? uint32
            +--rw required-min-rx-interval? uint32
          +--:(single-interval) {single-minimum-interval}?
            +--rw min-interval?          uint32
        +--rw demand-enabled?            boolean
        | {demand-mode}?
        +--rw admin-down?                boolean
        +--rw authentication! {authentication}?
          +--rw key-chain?               kc:key-chain-ref
          +--rw meticulous?             boolean
        +--ro path-type?                 identityref
        +--ro ip-encapsulation?          boolean
        +--ro local-discriminator?       discriminator
        +--ro remote-discriminator?      discriminator
        +--ro remote-multiplier?         multiplier
        +--ro demand-capability?         boolean
        | {demand-mode}?
        +--ro source-port?               inet:port-number
        +--ro dest-port?                 inet:port-number
        +--ro session-running
          +--ro session-index?           uint32
          +--ro local-state?             state
          +--ro remote-state?            state
          +--ro local-diagnostic?
            | iana-bfd-types:diagnostic
          +--ro remote-diagnostic?
            | iana-bfd-types:diagnostic
          +--ro remote-authenticated?     boolean
          +--ro remote-authentication-type?
            | iana-bfd-types:auth-type {authentication}?
          +--ro detection-mode?           enumeration
          +--ro negotiated-tx-interval?   uint32

```

```

|         |   +--ro negotiated-rx-interval?      uint32
|         |   +--ro detection-time?              uint32
|         |   +--ro echo-tx-interval-in-use?      uint32
|         |       {echo-mode}?
+--ro session-statistics
|   +--ro create-time?
|       |   yang:date-and-time
+--ro last-down-time?
|       |   yang:date-and-time
+--ro last-up-time?
|       |   yang:date-and-time
+--ro down-count?                                yang:counter32
+--ro admin-down-count?                          yang:counter32
+--ro receive-packet-count?                       yang:counter64
+--ro send-packet-count?                          yang:counter64
+--ro receive-invalid-packet-count?               yang:counter64
+--ro send-failed-packet-count?                   yang:counter64
+--rw interfaces* [interface]
|   +--rw interface          if:interface-ref
|   +--rw authentication!    {authentication}?
|       +--rw key-chain?     kc:key-chain-ref
|       +--rw meticulous?    boolean
notifications:
+---n singlehop-notification
|   +--ro local-discr?        discriminator
|   +--ro remote-discr?       discriminator
|   +--ro new-state?          state
|   +--ro state-change-reason? iana-bfd-types:diagnostic
|   +--ro time-of-last-state-change? yang:date-and-time
|   +--ro dest-addr?          inet:ip-address
|   +--ro source-addr?        inet:ip-address
|   +--ro session-index?      uint32
|   +--ro path-type?          identityref
|   +--ro interface?          if:interface-ref
|   +--ro echo-enabled?       boolean

```

2.7. BFD IP multihop hierarchy

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

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

```

augment /rt:routing/rt:control-plane-protocols
  /rt:control-plane-protocol/bfd:bfd:
  +--rw ip-mh
    +--ro summary
      | +--ro number-of-sessions?          yang:gauge32
      | +--ro number-of-sessions-up?      yang:gauge32
      | +--ro number-of-sessions-down?    yang:gauge32
      | +--ro number-of-sessions-admin-down? yang:gauge32
    +--rw session-groups
      +--rw session-group* [source-addr dest-addr]
        +--rw source-addr                inet:ip-address
        +--rw dest-addr                  inet:ip-address
        +--rw local-multiplier?          multiplier
        +--rw (interval-config-type)?
          | +--:(tx-rx-intervals)
          | | +--rw desired-min-tx-interval? uint32
          | | +--rw required-min-rx-interval? uint32
          | +--:(single-interval) {single-minimum-interval}?
          | +--rw min-interval?          uint32
        +--rw demand-enabled?            boolean
          | {demand-mode}?
        +--rw admin-down?                boolean
        +--rw authentication! {authentication}?
          | +--rw key-chain?      kc:key-chain-ref
          | +--rw meticulous?    boolean
        +--rw tx-ttl?              bfd-types:hops
        +--rw rx-ttl              bfd-types:hops
      +--ro sessions* []
        +--ro path-type?          identityref
        +--ro ip-encapsulation?    boolean
        +--ro local-discriminator? discriminator
        +--ro remote-discriminator? discriminator
        +--ro remote-multiplier?   multiplier
        +--ro demand-capability?   boolean {demand-mode}?
        +--ro source-port?         inet:port-number
        +--ro dest-port?          inet:port-number
        +--ro session-running
          | +--ro session-index?      uint32
          | +--ro local-state?        state
          | +--ro remote-state?       state
          | +--ro local-diagnostic?
          | | iana-bfd-types:diagnostic
          | +--ro remote-diagnostic?
          | | iana-bfd-types:diagnostic
          | +--ro remote-authenticated? boolean
          | +--ro remote-authentication-type?
          | | iana-bfd-types:auth-type {authentication}?
          | +--ro detection-mode?    enumeration

```

```

    |   +--ro negotiated-tx-interval?      uint32
    |   +--ro negotiated-rx-interval?      uint32
    |   +--ro detection-time?              uint32
    |   +--ro echo-tx-interval-in-use?      uint32
    |       {echo-mode}?
+--ro session-statistics
    +--ro create-time?
    |       yang:date-and-time
    +--ro last-down-time?
    |       yang:date-and-time
    +--ro last-up-time?
    |       yang:date-and-time
    +--ro down-count?
    |       yang:counter32
    +--ro admin-down-count?
    |       yang:counter32
    +--ro receive-packet-count?
    |       yang:counter64
    +--ro send-packet-count?
    |       yang:counter64
    +--ro receive-invalid-packet-count?
    |       yang:counter64
    +--ro send-failed-packet-count?
    |       yang:counter64

notifications:
+---n multihop-notification
    +--ro local-dscr?                      discriminator
    +--ro remote-dscr?                     discriminator
    +--ro new-state?                       state
    +--ro state-change-reason?             iana-bfd-types:diagnostic
    +--ro time-of-last-state-change?       yang:date-and-time
    +--ro dest-addr?                       inet:ip-address
    +--ro source-addr?                     inet:ip-address
    +--ro session-index?                   uint32
    +--ro path-type?                       identityref

```

2.8. BFD over LAG hierarchy

A "lag" node is added under the "bfd" node in control-plane-protocol. The configuration and operational state data for each BFD LAG session is under this "lag" node.

```

module: ietf-bfd-lag
augment /rt:routing/rt:control-plane-protocols
    /rt:control-plane-protocol/bfd:bfd:
    +--rw lag

```



```

+--rw micro-bfd-ipv4-session-statistics
|   +--ro summary
|   |   +--ro number-of-sessions?          yang:gauge32
|   |   +--ro number-of-sessions-up?        yang:gauge32
|   |   +--ro number-of-sessions-down?      yang:gauge32
|   |   +--ro number-of-sessions-admin-down? yang:gauge32
+--rw micro-bfd-ipv6-session-statistics
|   +--ro summary
|   |   +--ro number-of-sessions?          yang:gauge32
|   |   +--ro number-of-sessions-up?        yang:gauge32
|   |   +--ro number-of-sessions-down?      yang:gauge32
|   |   +--ro number-of-sessions-admin-down? yang:gauge32
+--rw sessions
|   +--rw session* [lag-name]
|   |   +--rw lag-name                      if:interface-ref
|   |   +--rw ipv4-dest-addr?
|   |   |   inet:ipv4-address
|   |   +--rw ipv6-dest-addr?
|   |   |   inet:ipv6-address
|   |   +--rw local-multiplier?             multiplier
|   |   +--rw (interval-config-type)?
|   |   |   +--:(tx-rx-intervals)
|   |   |   |   +--rw desired-min-tx-interval? uint32
|   |   |   |   +--rw required-min-rx-interval? uint32
|   |   |   +--:(single-interval) {single-minimum-interval}?
|   |   |   |   +--rw min-interval?           uint32
|   |   +--rw demand-enabled?               boolean
|   |   |   {demand-mode}?
|   |   +--rw admin-down?                   boolean
|   |   +--rw authentication! {authentication}?
|   |   |   +--rw key-chain?      kc:key-chain-ref
|   |   |   +--rw meticulous?    boolean
|   |   +--rw use-ipv4?             boolean
|   |   +--rw use-ipv6?             boolean
|   |   +--ro member-links* [member-link]
|   |   |   +--ro member-link      if:interface-ref
|   |   |   +--ro micro-bfd-ipv4
|   |   |   |   +--ro path-type?          identityref
|   |   |   |   +--ro ip-encapsulation?    boolean
|   |   |   |   +--ro local-discriminator? discriminator
|   |   |   |   +--ro remote-discriminator? discriminator
|   |   |   |   +--ro remote-multiplier?   multiplier
|   |   |   |   +--ro demand-capability?   boolean
|   |   |   |   |   {demand-mode}?
|   |   |   +--ro source-port?            inet:port-number
|   |   |   +--ro dest-port?              inet:port-number
|   |   |   +--ro session-running
|   |   |   |   +--ro session-index?       uint32

```

```

+---ro local-state?                               state
+---ro remote-state?                             state
+---ro local-diagnostic?
|         iana-bfd-types:diagnostic
+---ro remote-diagnostic?
|         iana-bfd-types:diagnostic
+---ro remote-authenticated?                     boolean
+---ro remote-authentication-type?
|         iana-bfd-types:auth-type
|         {authentication}?
+---ro detection-mode?                           enumeration
+---ro negotiated-tx-interval?                   uint32
+---ro negotiated-rx-interval?                   uint32
+---ro detection-time?                           uint32
+---ro echo-tx-interval-in-use?                  uint32
|         {echo-mode}?
+---ro session-statistics
+---ro create-time?
|         yang:date-and-time
+---ro last-down-time?
|         yang:date-and-time
+---ro last-up-time?
|         yang:date-and-time
+---ro down-count?
|         yang:counter32
+---ro admin-down-count?
|         yang:counter32
+---ro receive-packet-count?
|         yang:counter64
+---ro send-packet-count?
|         yang:counter64
+---ro receive-invalid-packet-count?
|         yang:counter64
+---ro send-failed-packet-count?
|         yang:counter64
+---ro micro-bfd-ipv6
+---ro path-type?                                identityref
+---ro ip-encapsulation?                         boolean
+---ro local-discriminator?                      discriminator
+---ro remote-discriminator?                    discriminator
+---ro remote-multiplier?                        multiplier
+---ro demand-capability?                       boolean
|         {demand-mode}?
+---ro source-port?                              inet:port-number
+---ro dest-port?                                inet:port-number
+---ro session-running
|         +---ro session-index?                  uint32
|         +---ro local-state?                    state

```

```

|   +--ro remote-state?                               state
|   +--ro local-diagnostic?
|   |   iana-bfd-types:diagnostic
|   +--ro remote-diagnostic?
|   |   iana-bfd-types:diagnostic
|   +--ro remote-authenticated?                       boolean
|   +--ro remote-authentication-type?
|   |   iana-bfd-types:auth-type
|   |   {authentication}?
|   +--ro detection-mode?                             enumeration
|   +--ro negotiated-tx-interval?                     uint32
|   +--ro negotiated-rx-interval?                     uint32
|   +--ro detection-time?                             uint32
|   +--ro echo-tx-interval-in-use?                   uint32
|   |   {echo-mode}?
+--ro session-statistics
|   +--ro create-time?
|   |   yang:date-and-time
|   +--ro last-down-time?
|   |   yang:date-and-time
|   +--ro last-up-time?
|   |   yang:date-and-time
|   +--ro down-count?
|   |   yang:counter32
|   +--ro admin-down-count?
|   |   yang:counter32
|   +--ro receive-packet-count?
|   |   yang:counter64
|   +--ro send-packet-count?
|   |   yang:counter64
|   +--ro receive-invalid-packet-count?
|   |   yang:counter64
|   +--ro send-failed-packet-count?
|   |   yang:counter64
notifications:
+---n lag-notification
|   +--ro local-discr?                               discriminator
|   +--ro remote-discr?                             discriminator
|   +--ro new-state?                                 state
|   +--ro state-change-reason?                      iana-bfd-types:diagnostic
|   +--ro time-of-last-state-change?                yang:date-and-time
|   +--ro dest-addr?                                inet:ip-address
|   +--ro source-addr?                              inet:ip-address
|   +--ro session-index?                            uint32
|   +--ro path-type?                                identityref
|   +--ro lag-name?                                  if:interface-ref
|   +--ro member-link?                              if:interface-ref

```

2.9. BFD over MPLS LSPs hierarchy

An "mpls" node is added under the "bfd" node in control-plane-protocol. The configuration is per MPLS FEC under this "mpls" node. In the operational state model we support multiple BFD sessions per MPLS FEC (ECMP), the local discriminator is used as key. The "mpls" node can be used in a network device (top-level), or mounted in an LNE or in a network instance.

```

module: ietf-bfd-mpls
  augment /rt:routing/rt:control-plane-protocols
    /rt:control-plane-protocol/bfd:bfd:
      +--rw mpls
        +--ro summary
          | +--ro number-of-sessions?                yang:gauge32
          | +--ro number-of-sessions-up?            yang:gauge32
          | +--ro number-of-sessions-down?          yang:gauge32
          | +--ro number-of-sessions-admin-down?    yang:gauge32
        +--rw egress
          | +--rw enable?                            boolean
          | +--rw local-multiplier?                  multiplier
          | +--rw (interval-config-type)?
          |   | +---:(tx-rx-intervals)
          |   | | +--rw desired-min-tx-interval?    uint32
          |   | | +--rw required-min-rx-interval?   uint32
          |   | +---:(single-interval) {single-minimum-interval}?
          |   | | +--rw min-interval?               uint32
          | +--rw authentication! {authentication}?
          |   +--rw key-chain?      kc:key-chain-ref
          |   +--rw meticulous?    boolean
        +--rw session-groups
          +--rw session-group* [mpls-fec]
            +--rw mpls-fec                inet:ip-prefix
            +--rw local-multiplier?        multiplier
            +--rw (interval-config-type)?
            | +---:(tx-rx-intervals)
            | | +--rw desired-min-tx-interval?    uint32
            | | +--rw required-min-rx-interval?   uint32
            | +---:(single-interval) {single-minimum-interval}?
            | | +--rw min-interval?               uint32
            +--rw demand-enabled?          boolean
            | {demand-mode}?
            +--rw admin-down?              boolean
            +--rw authentication! {authentication}?
            | +--rw key-chain?      kc:key-chain-ref
            | +--rw meticulous?    boolean
            +--ro sessions* []

```

```

+--ro path-type?                identityref
+--ro ip-encapsulation?         boolean
+--ro local-discriminator?     discriminator
+--ro remote-discriminator?    discriminator
+--ro remote-multiplier?       multiplier
+--ro demand-capability?       boolean {demand-mode}?
+--ro source-port?             inet:port-number
+--ro dest-port?               inet:port-number
+--ro session-running
|   +--ro session-index?        uint32
|   +--ro local-state?         state
|   +--ro remote-state?        state
|   +--ro local-diagnostic?
|   |   iana-bfd-types:diagnostic
|   +--ro remote-diagnostic?
|   |   iana-bfd-types:diagnostic
|   +--ro remote-authenticated? boolean
|   +--ro remote-authentication-type?
|   |   iana-bfd-types:auth-type {authentication}?
|   +--ro detection-mode?      enumeration
|   +--ro negotiated-tx-interval? uint32
|   +--ro negotiated-rx-interval? uint32
|   +--ro detection-time?      uint32
|   +--ro echo-tx-interval-in-use? uint32
|   |   {echo-mode}?
+--ro session-statistics
|   +--ro create-time?
|   |   yang:date-and-time
|   +--ro last-down-time?
|   |   yang:date-and-time
|   +--ro last-up-time?
|   |   yang:date-and-time
|   +--ro down-count?
|   |   yang:counter32
|   +--ro admin-down-count?
|   |   yang:counter32
|   +--ro receive-packet-count?
|   |   yang:counter64
|   +--ro send-packet-count?
|   |   yang:counter64
|   +--ro receive-invalid-packet-count?
|   |   yang:counter64
|   +--ro send-failed-packet-count?
|   |   yang:counter64
+--ro mpls-dest-address?       inet:ip-address

```

notifications:

```

+---n mpls-notification

```

+++ro local-dscr?	discriminator
+++ro remote-dscr?	discriminator
+++ro new-state?	state
+++ro state-change-reason?	iana-bfd-types:diagnostic
+++ro time-of-last-state-change?	yang:date-and-time
+++ro dest-addr?	inet:ip-address
+++ro source-addr?	inet:ip-address
+++ro session-index?	uint32
+++ro path-type?	identityref
+++ro mpls-dest-address?	inet:ip-address

2.10. BFD over MPLS-TE hierarchy

YANG Data Model for TE Topologies [I-D.ietf-teas-yang-te] is augmented. BFD is configured per MPLS-TE tunnel, and BFD session operational state data is provided per MPLS-TE LSP.

```

module: ietf-bfd-mpls-te
  augment /rt:routing/rt:control-plane-protocols
    /rt:control-plane-protocol/bfd:bfd:
      +--rw mpls-te
        +--rw egress
          |   +--rw enable?                               boolean
          |   +--rw local-multiplier?                     multiplier
          |   +--rw (interval-config-type)?
          |     |   +--:(tx-rx-intervals)
          |     |   |   +--rw desired-min-tx-interval?    uint32
          |     |   |   +--rw required-min-rx-interval?    uint32
          |     |   +--:(single-interval) {single-minimum-interval}?
          |     |   |   +--rw min-interval?                uint32
          |     +--rw authentication! {authentication}?
          |     +--rw key-chain?      kc:key-chain-ref
          |     +--rw meticulous?     boolean
        +--ro summary
          +--ro number-of-sessions?          yang:gauge32
          +--ro number-of-sessions-up?       yang:gauge32
          +--ro number-of-sessions-down?     yang:gauge32
          +--ro number-of-sessions-admin-down? yang:gauge32
      augment /te:te/te:tunnels/te:tunnel:
        +--rw local-multiplier?                multiplier
        +--rw (interval-config-type)?
          |   +--:(tx-rx-intervals)
          |   |   +--rw desired-min-tx-interval?    uint32
          |   |   +--rw required-min-rx-interval?    uint32
          |   +--:(single-interval) {single-minimum-interval}?
          |   |   +--rw min-interval?                uint32
        +--rw demand-enabled?                  boolean {demand-mode}?

```

```

    +--rw admin-down?                               boolean
    +--rw authentication! {authentication}?
    |   +--rw key-chain?      kc:key-chain-ref
    |   +--rw meticulous?    boolean
    +--rw encap?                                       identityref
augment /te:te/te:lsps-state/te:lsp:
    +--ro path-type?                                identityref
    +--ro ip-encapsulation?                          boolean
    +--ro local-discriminator?                      discriminator
    +--ro remote-discriminator?                    discriminator
    +--ro remote-multiplier?                        multiplier
    +--ro demand-capability?                        boolean {demand-mode}?
    +--ro source-port?                              inet:port-number
    +--ro dest-port?                                inet:port-number
    +--ro session-running
    |   +--ro session-index?                          uint32
    |   +--ro local-state?                            state
    |   +--ro remote-state?                          state
    |   +--ro local-diagnostic?                      iana-bfd-types:diagnostic
    |   +--ro remote-diagnostic?                    iana-bfd-types:diagnostic
    |   +--ro remote-authenticated?                  boolean
    |   +--ro remote-authentication-type?            iana-bfd-types:auth-type
    |   |   {authentication}?
    |   +--ro detection-mode?                        enumeration
    |   +--ro negotiated-tx-interval?                uint32
    |   +--ro negotiated-rx-interval?                uint32
    |   +--ro detection-time?                        uint32
    |   +--ro echo-tx-interval-in-use?                uint32 {echo-mode}?
    +--ro session-statistics
    |   +--ro create-time?                            yang:date-and-time
    |   +--ro last-down-time?                        yang:date-and-time
    |   +--ro last-up-time?                          yang:date-and-time
    |   +--ro down-count?                            yang:counter32
    |   +--ro admin-down-count?                      yang:counter32
    |   +--ro receive-packet-count?                  yang:counter64
    |   +--ro send-packet-count?                     yang:counter64
    |   +--ro receive-invalid-packet-count?          yang:counter64
    |   +--ro send-failed-packet-count?              yang:counter64
    +--ro mpls-dest-address?                          inet:ip-address

notifications:
    +---n mpls-te-notification
    |   +--ro local-discr?                          discriminator
    |   +--ro remote-discr?                        discriminator
    |   +--ro new-state?                            state
    |   +--ro state-change-reason?                  iana-bfd-types:diagnostic
    |   +--ro time-of-last-state-change?            yang:date-and-time
    |   +--ro dest-addr?                            inet:ip-address

```

+++ro source-addr?	inet:ip-address
+++ro session-index?	uint32
+++ro path-type?	identityref
+++ro mpls-dest-address?	inet:ip-address
+++ro tunnel-name?	string

2.11. Interaction with other YANG modules

Generic YANG Data Model for Connectionless OAM protocols [I-D.ietf-lime-yang-connectionless-oam] describes how the LIME connectionless OAM model could be extended to support BFD.

Also, the operation of the BFD data model depends on configuration parameters that are defined in other YANG modules.

2.11.1. Module ietf-interfaces

The following boolean configuration is defined in A YANG Data Model for Interface Management [RFC8343]:

```
/if:interfaces/if:interface/if:enabled
    If this configuration is set to "false", no BFD packets can
    be transmitted or received on that interface.
```

2.11.2. Module ietf-ip

The following boolean configuration is defined in A YANG Data Model for IP Management [RFC8344]:

```
/if:interfaces/if:interface/ip:ipv4/ip:enabled
    If this configuration is set to "false", no BFD IPv4 packets
    can be transmitted or received on that interface.

/if:interfaces/if:interface/ip:ipv4/ip:forwarding
    If this configuration is set to "false", no BFD IPv4 packets
    can be transmitted or received on that interface.

/if:interfaces/if:interface/ip:ipv6/ip:enabled
    If this configuration is set to "false", no BFD IPv6 packets
    can be transmitted or received on that interface.

/if:interfaces/if:interface/ip:ipv6/ip:forwarding
    If this configuration is set to "false", no BFD IPv6 packets
    can be transmitted or received on that interface.
```


2.11.3. Module ietf-mpls

The following boolean configuration is defined in A YANG Data Model for MPLS Base [I-D.ietf-mpls-base-yang]:

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

2.11.4. Module ietf-te

The following configuration is defined in the "ietf-te" YANG module YANG Data Model for TE Topology [I-D.ietf-teas-yang-te]:

```
/ietf-te:te/ietf-te:tunnels/ietf-te:tunnel/ietf-te:config/ietf-
te:admin-status
    If this configuration is not set to "state-up", no BFD MPLS
    packets can be transmitted or received on that tunnel.
```

2.12. IANA BFD YANG Module

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

module iana-bfd-types {

    yang-version 1.1;

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

    prefix "iana-bfd-types";

    organization "IANA";

    contact
        "
            Internet Assigned Numbers Authority

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

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

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

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

Copyright (c) 2018 IETF Trust and the persons identified as authors of the code. All rights reserved.

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

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

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

```
reference "RFC XXXX";
```

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

```
/*
 * Type Definitions
 */
typedef diagnostic {
  type enumeration {
    enum none {
      value 0;
      description "None";
    }
    enum control-expiry {
      value 1;
      description "Control timer expiry";
    }
    enum echo-failed {
      value 2;
      description "Echo failure";
    }
    enum neighbor-down {
      value 3;
      description "Neighbor down";
    }
    enum forwarding-reset {
```

```
        value 4;
        description "Forwarding reset";
    }
    enum path-down {
        value 5;
        description "Path down";
    }
    enum concatenated-path-down {
        value 6;
        description "Concatenated path down";
    }
    enum admin-down {
        value 7;
        description "Admin down";
    }
    enum reverse-concatenated-path-down {
        value 8;
        description "Reverse concatenated path down";
    }
    enum mis-connectivity-defect {
        value 9;
        description "Mis-connectivity defect as specified in RFC6428";
    }
}
description
    "BFD diagnostic as defined in RFC 5880, values are maintained in
    the 'BFD Diagnostic Codes' IANA registry. Range is 0 to 31.";
}

typedef auth-type {
    type enumeration {
        enum reserved {
            value 0;
            description "Reserved";
        }
        enum simple-password {
            value 1;
            description "Simple password";
        }
        enum keyed-md5 {
            value 2;
            description "Keyed MD5";
        }
        enum meticulous-keyed-md5 {
            value 3;
            description "Meticulous keyed MD5";
        }
        enum keyed-sha1 {
```

```
        value 4;
        description "Keyed SHA1";
    }
    enum meticulous-keyed-sha1 {
        value 5;
        description "Meticulous keyed SHA1";
    }
}
description
    "BFD authentication type as defined in RFC 5880, values are
    maintained in the 'BFD Authentication Types' IANA registry.
    Range is 0 to 255.";
}
}

<CODE ENDS>
```

2.13. BFD types YANG Module

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

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

```
module ietf-bfd-types {

    yang-version 1.1;

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

    prefix "bfd-types";

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

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

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

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

```
}

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

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

organization "IETF BFD Working Group";

contact
  "WG Web:    <http://tools.ietf.org/wg/bfd>
  WG List:    <rtg-bfd@ietf.org>

  Editors:    Reshad Rahman (rrahman@cisco.com),
              Lianshu Zheng (vero.zheng@huawei.com),
              Mahesh Jethanandani (mjethanandani@gmail.com)";

description
  "This module contains a collection of BFD specific YANG data type
  definitions, as per RFC 5880, and also groupings which are common
  to other BFD YANG modules.

  Copyright (c) 2018 IETF Trust and the persons
  identified as authors of the code. All rights reserved.

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

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

reference "RFC XXXX";

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

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

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

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

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

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

identity path-type {
```

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

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

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

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

typedef state {
    type enumeration {
        enum adminDown {
            value 0;
            description "admindown";
        }
        enum down {
            value 1;
            description "down";
        }
        enum init {
            value 2;
            description "init";
        }
        enum up {
            value 3;
            description "up";
        }
    }
    description "BFD state as defined in RFC 5880.";
}

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

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



```
}

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

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

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

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

  choice interval-config-type {
    description
      "Two interval values or one value used for both transmit and
      receive.";
    case tx-rx-intervals {
      leaf desired-min-tx-interval {
        type uint32;
        units microseconds;
        default 1000000;
        description
          "Desired minimum transmit interval of control packets.";
      }
    }
  }
}
```

```
    }

    leaf required-min-rx-interval {
        type uint32;
        units microseconds;
        default 1000000;
        description
            "Required minimum receive interval of control packets.";
    }
}
case single-interval {
    if-feature single-minimum-interval;

    leaf min-interval {
        type uint32;
        units microseconds;
        default 1000000;
        description
            "Desired minimum transmit interval and required " +
            "minimum receive interval of control packets.";
    }
}
}
}

grouping client-cfg-parms {
    description
        "BFD grouping for configuration parameters
        used by clients of BFD, e.g. IGP or MPLS.";

    leaf enable {
        type boolean;
        default false;
        description
            "Indicates whether the BFD is enabled.";
    }
    uses base-cfg-parms;
}

grouping common-cfg-parms {
    description
        "BFD grouping for common configuration parameters.";

    uses base-cfg-parms;

    leaf demand-enabled {
        if-feature demand-mode;
        type boolean;
    }
}
```

```
    default false;
    description
        "To enable demand mode.";
}

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

grouping all-session {
    description "BFD session operational information";
    leaf path-type {
        type identityref {
            base path-type;
        }
        config "false";
        description
            "BFD path type, this indicates the path type that BFD is
            running on.";
    }
    leaf ip-encapsulation {
        type boolean;
        config "false";
        description "Whether BFD encapsulation uses IP.";
    }
    leaf local-discriminator {
        type discriminator;
        config "false";
        description "Local discriminator.";
    }
    leaf remote-discriminator {
        type discriminator;
        config "false";
        description "Remote discriminator.";
    }
    leaf remote-multiplier {
        type multiplier;
        config "false";
        description "Remote multiplier.";
    }
    leaf demand-capability {
        if-feature demand-mode;
        type boolean;
    }
}
```

```
    config "false";
    description "Local demand mode capability.";
  }
  leaf source-port {
    when "../ip-encapsulation = 'true'" {
      description
        "Source port valid only when IP encapsulation is used.";
    }
    type inet:port-number;
    config "false";
    description "Source UDP port";
  }
  leaf dest-port {
    when "../ip-encapsulation = 'true'" {
      description
        "Destination port valid only when IP encapsulation is used.";
    }
    type inet:port-number;
    config "false";
    description "Destination UDP port.";
  }
}

container session-running {
  config "false";
  description "BFD session running information.";
  leaf session-index {
    type uint32;
    description
      "An index used to uniquely identify BFD sessions.";
  }
  leaf local-state {
    type state;
    description "Local state.";
  }
  leaf remote-state {
    type state;
    description "Remote state.";
  }
  leaf local-diagnostic {
    type iana-bfd-types:diagnostic;
    description "Local diagnostic.";
  }
  leaf remote-diagnostic {
    type iana-bfd-types:diagnostic;
    description "Remote diagnostic.";
  }
  leaf remote-authenticated {
    type boolean;
  }
}
```

```
    description
      "Indicates whether incoming BFD control packets are
      authenticated.";
  }
  leaf remote-authentication-type {
    when "../remote-authenticated = 'true'" {
      description
        "Only valid when incoming BFD control packets are
        authenticated.";
    }
    if-feature authentication;
    type iana-bfd-types:auth-type;
    description
      "Authentication type of incoming BFD control packets.";
  }
  leaf detection-mode {
    type enumeration {
      enum async-with-echo {
        value "1";
        description "Async with echo.";
      }
      enum async-without-echo {
        value "2";
        description "Async without echo.";
      }
      enum demand-with-echo {
        value "3";
        description "Demand with echo.";
      }
      enum demand-without-echo {
        value "4";
        description "Demand without echo.";
      }
    }
    description "Detection mode.";
  }
  leaf negotiated-tx-interval {
    type uint32;
    units microseconds;
    description "Negotiated transmit interval.";
  }
  leaf negotiated-rx-interval {
    type uint32;
    units microseconds;
    description "Negotiated receive interval.";
  }
  leaf detection-time {
    type uint32;
```

```
        units microseconds;
        description "Detection time.";
    }
    leaf echo-tx-interval-in-use {
        when "../..//path-type = 'bfd-types:path-ip-sh'" {
            description
                "Echo is supported for IP single-hop only.";
        }
        if-feature echo-mode;
        type uint32;
        units microseconds;
        description "Echo transmit interval in use.";
    }
}

container session-statistics {
    config "false";
    description "BFD per-session statistics.";

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

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

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

```
    }  
  }  
  
  grouping notification-parms {  
    description  
      "This group describes common parameters that will be sent " +  
      "as part of BFD notification.";  
  
    leaf local-discr {  
      type discriminator;  
      description "BFD local discriminator.";  
    }  
  
    leaf remote-discr {  
      type discriminator;  
      description "BFD remote discriminator.";  
    }  
  
    leaf new-state {  
      type state;  
      description "Current BFD state.";  
    }  
  
    leaf state-change-reason {  
      type iana-bfd-types:diagnostic;  
      description "BFD state change reason.";  
    }  
  
    leaf time-of-last-state-change {  
      type yang:date-and-time;  
      description  
        "Calendar time of previous state change.";  
    }  
  
    leaf dest-addr {  
      type inet:ip-address;  
      description "BFD peer address.";  
    }  
  
    leaf source-addr {  
      type inet:ip-address;  
      description "BFD local address.";  
    }  
  
    leaf session-index {  
      type uint32;  
      description "An index used to uniquely identify BFD sessions.";  
    }  
  }
```



```
    leaf path-type {  
      type identityref {  
        base path-type;  
      }  
      description "BFD path type.";  
    }  
  }  
}
```

<CODE ENDS>

2.14. BFD top-level YANG Module

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

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

```
module ietf-bfd {  
  
  yang-version 1.1;  
  
  namespace "urn:ietf:params:xml:ns:yang:ietf-bfd";  
  
  prefix "bfd";  
  
  // RFC Ed.: replace occurrences of XXXX with actual RFC number and  
  // remove this note  
  
  import ietf-bfd-types {  
    prefix "bfd-types";  
    reference "RFC XXXX: YANG Data Model for BFD";  
  }  
  
  import ietf-routing {  
    prefix "rt";  
    reference  
      "RFC 8349: A YANG Data Model for Routing Management  
      (NMDA version)";  
  }  
  
  organization "IETF BFD Working Group";  
  
  contact  
    "WG Web:  <http://tools.ietf.org/wg/bfd>  
    WG List:  <rtg-bfd@ietf.org>  
  
    Editors:  Reshad Rahman (rrahman@cisco.com),
```

Lianshu Zheng (vero.zheng@huawei.com),
Mahesh Jethanandani (mjethanandani@gmail.com)";

description

"This module contains the YANG definition for BFD parameters as per RFC 5880.

Copyright (c) 2018 IETF Trust and the persons
identified as authors of the code. All rights reserved.

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

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

reference "RFC XXXX";

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

augment "/rt:routing/rt:control-plane-protocols/"
 + "rt:control-plane-protocol" {
 when "derived-from-or-self(rt:type, 'bfd-types:bfdv1')" {
 description
 "This augmentation is only valid for a control-plane protocol
 instance of BFD (type 'bfdv1').";
 }
 description "BFD augmentation.";

container bfd {
 description "BFD top level container.";
 uses bfd-types:session-statistics-summary;
 }
}

<CODE ENDS>

2.15. BFD IP single-hop YANG Module

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

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

```
module ietf-bfd-ip-sh {  
    yang-version 1.1;  
  
    namespace "urn:ietf:params:xml:ns:yang:ietf-bfd-ip-sh";  
  
    prefix "bfd-ip-sh";  
  
    // RFC Ed.: replace occurrences of XXXX with actual RFC number and  
    // remove this note  
  
    import ietf-bfd-types {  
        prefix "bfd-types";  
        reference "RFC XXXX: YANG Data Model for BFD";  
    }  
  
    import ietf-bfd {  
        prefix "bfd";  
        reference "RFC XXXX: YANG Data Model for BFD";  
    }  
  
    import ietf-interfaces {  
        prefix "if";  
        reference  
            "RFC 8343: A YANG Data Model for Interface Management";  
    }  
  
    import ietf-inet-types {  
        prefix "inet";  
        reference "RFC 6991: Common YANG Data Types";  
    }  
  
    import ietf-routing {  
        prefix "rt";  
        reference  
            "RFC 8349: A YANG Data Model for Routing Management  
            (NMDA version)";  
    }  
  
    organization "IETF BFD Working Group";
```

contact

"WG Web: <<http://tools.ietf.org/wg/bfd>>
WG List: <rtg-bfd@ietf.org>

Editors: Reshad Rahman (rrahman@cisco.com),
Lianshu Zheng (vero.zheng@huawei.com),
Mahesh Jethanandani (mjethanandani@gmail.com)";

description

"This module contains the YANG definition for BFD IP single-hop as per RFC 5881.

Copyright (c) 2018 IETF Trust and the persons identified as authors of the code. All rights reserved.

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

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

reference "RFC XXXX";

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

/*
 * Augments
 */

augment "/rt:routing/rt:control-plane-protocols/"
 + "rt:control-plane-protocol/bfd:bfd" {
 description "BFD augmentation for IP single-hop";
 container ip-sh {
 description "BFD IP single-hop top level container";

 uses bfd-types:session-statistics-summary;

 container sessions {
 description
 "BFD IP single-hop sessions.";
 list session {
 key "interface dest-addr";

```
    description "List of IP single-hop sessions.";
    leaf interface {
        type if:interface-ref;
        description
            "Interface on which the BFD session is running.";
    }
    leaf dest-addr {
        type inet:ip-address;
        description "IP address of the peer.";
    }
    leaf source-addr {
        type inet:ip-address;
        description "Local IP address.";
    }
    uses bfd-types:common-cfg-parms;
    uses bfd-types:all-session;
}
}
list interfaces {
    key "interface";
    description "List of interfaces.";
    leaf interface {
        type if:interface-ref;
        description
            "BFD information for this interface.";
    }
    uses bfd-types:auth-parms;
}
}
}

/*
 * Notifications
 */
notification singlehop-notification {
    description
        "Notification for BFD single-hop session state change. An " +
        "implementation may rate-limit notifications, e.g. when a " +
        "session is continuously changing state.";
    uses bfd-types:notification-parms;
    leaf interface {
        type if:interface-ref;
        description "Interface to which this BFD session belongs to.";
    }
}
```

```
    }  
    leaf echo-enabled {  
      type boolean;  
      description "Was echo enabled for BFD.";  
    }  
  }  
}
```

<CODE ENDS>

2.16. BFD IP multihop YANG Module

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

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

```
module ietf-bfd-ip-mh {  
  yang-version 1.1;  
  namespace "urn:ietf:params:xml:ns:yang:ietf-bfd-ip-mh";  
  prefix "bfd-ip-mh";  
  // RFC Ed.: replace occurrences of XXXX with actual RFC number and  
  // remove this note  
  import ietf-bfd-types {  
    prefix "bfd-types";  
    reference "RFC XXXX: YANG Data Model for BFD";  
  }  
  import ietf-bfd {  
    prefix "bfd";  
    reference "RFC XXXX: YANG Data Model for BFD";  
  }  
  import ietf-inet-types {  
    prefix "inet";  
    reference "RFC 6991: Common YANG Data Types";  
  }  
  import ietf-routing {  
    prefix "rt";  
  }
```

```
reference
  "RFC 8349: A YANG Data Model for Routing Management
  (NMDA version)";
}

organization "IETF BFD Working Group";

contact
  "WG Web:    <http://tools.ietf.org/wg/bfd>
  WG List:    <rtg-bfd@ietf.org>

  Editors:    Reshad Rahman (rrahman@cisco.com),
               Lianshu Zheng (vero.zheng@huawei.com),
               Mahesh Jethanandani (mjethanandani@gmail.com)";

description
  "This module contains the YANG definition for BFD IP multi-hop
  as per RFC 5883.

  Copyright (c) 2018 IETF Trust and the persons
  identified as authors of the code. All rights reserved.

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

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

reference "RFC XXXX";

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

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

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

container session-groups {
  description
    "BFD IP multi-hop session groups.";
  list session-group {
    key "source-addr dest-addr";
    description
      "Group of BFD IP multi-hop sessions (for ECMP). A " +
      "group of sessions is between 1 source and 1 " +
      "destination, each session has a different field " +
      "in UDP/IP hdr for ECMP.";

    leaf source-addr {
      type inet:ip-address;
      description
        "Local IP address.";
    }
    leaf dest-addr {
      type inet:ip-address;
      description
        "IP address of the peer.";
    }
    uses bfd-types:common-cfg-parms;

    leaf tx-ttl {
      type bfd-types:hops;
      default 255;
      description "Hop count of outgoing BFD control packets.";
    }
    leaf rx-ttl {
      type bfd-types:hops;
      mandatory true;
      description
        "Minimum allowed hop count value for incoming BFD control
        packets. Control packets whose hop count is lower than
        this value are dropped.";
    }
    list sessions {
      config false;
      description
        "The multiple BFD sessions between a source and a " +
        "destination.";
      uses bfd-types:all-session;
    }
  }
}
```



```
    }  
  
    /*  
    * Notifications  
    */  
    notification multihop-notification {  
        description  
            "Notification for BFD multi-hop session state change. An " +  
            "implementation may rate-limit notifications, e.g. when a " +  
            "session is continuously changing state.";  
  
        uses bfd-types:notification-parms;  
    }  
}  
  
<CODE ENDS>
```

2.17. BFD over LAG YANG Module

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

```
<CODE BEGINS> file "ietf-bfd-lag@2018-08-01.yang"  
  
module ietf-bfd-lag {  
  
    yang-version 1.1;  
  
    namespace "urn:ietf:params:xml:ns:yang:ietf-bfd-lag";  
  
    prefix "bfd-lag";  
  
    // RFC Ed.: replace occurrences of XXXX with actual RFC number and  
    // remove this note  
  
    import ietf-bfd-types {  
        prefix "bfd-types";  
        reference "RFC XXXX: YANG Data Model for BFD";  
    }  
  
    import ietf-bfd {  
        prefix "bfd";  
        reference "RFC XXXX: YANG Data Model for BFD";  
    }  
  
    import ietf-interfaces {  
        prefix "if";  
    }  
}
```

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

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

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

organization "IETF BFD Working Group";

contact
  "WG Web:    <http://tools.ietf.org/wg/bfd>
  WG List:    <rtg-bfd@ietf.org>

  Editors:    Reshad Rahman (rrahman@cisco.com),
               Lianshu Zheng vero.zheng@huawei.com),
               Mahesh Jethanandani (mjethanandani@gmail.com)";

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

  Copyright (c) 2018 IETF Trust and the persons
  identified as authors of the code. All rights reserved.

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

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

reference "RFC XXXX";

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

```
}

/*
 * Augments
 */
augment "/rt:routing/rt:control-plane-protocols/"
  + "rt:control-plane-protocol/bfd:bfd" {
  description "BFD augmentation for LAG";
  container lag {
    description "BFD over LAG top level container";

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

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

    leaf use-ipv4 {
      type boolean;
      description "Using IPv4 micro-BFD.";
    }
    leaf use-ipv6 {
      type boolean;
    }
  }
}
```

```

        description "Using IPv6 micro-BFD.";
    }

    list member-links {
        key "member-link";
        config false;
        description
            "Micro-BFD over LAG. This represents one member link.";

        leaf member-link {
            type if:interface-ref;
            description
                "Member link on which micro-BFD is running.";
        }
        container micro-bfd-ipv4 {
            when "../..use-ipv4 = 'true'" {
                description "Needed only if IPv4 is used.";
            }
            description
                "Micro-BFD IPv4 session state on member link.";
            uses bfd-types:all-session;
        }
        container micro-bfd-ipv6 {
            when "../..use-ipv6 = 'true'" {
                description "Needed only if IPv6 is used.";
            }
            description
                "Micro-BFD IPv6 session state on member link.";
            uses bfd-types:all-session;
        }
    }
}

/*
 * Notifications
 */
notification lag-notification {
    description
        "Notification for BFD over LAG session state change. " +
        "An implementation may rate-limit notifications, e.g. when a " +
        "session is continuously changing state.";

    uses bfd-types:notification-parms;

    leaf lag-name {

```

```
    type if:interface-ref;
    description "LAG interface name.";
  }

  leaf member-link {
    type if:interface-ref;
    description "Member link on which BFD is running.";
  }
}
```

<CODE ENDS>

2.18. BFD over MPLS YANG Module

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

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

```
module ietf-bfd-mpls {

  yang-version 1.1;

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

  prefix "bfd-mpls";

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

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

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

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

  import ietf-routing {
```

```
    prefix "rt";
    reference
      "RFC 8349: A YANG Data Model for Routing Management
      (NMDA version)";
  }

  organization "IETF BFD Working Group";

  contact
    "WG Web:    <http://tools.ietf.org/wg/bfd>
    WG List:    <rtg-bfd@ietf.org>

    Editors:    Reshad Rahman (rrahman@cisco.com),
                Lianshu Zheng (vero.zheng@huawei.com),
                Mahesh Jethanandani (mjethanandani@gmail.com)";

  description
    "This module contains the YANG definition for BFD parameters for
    MPLS LSPs as per RFC 5884.

    Copyright (c) 2018 IETF Trust and the persons
    identified as authors of the code. All rights reserved.

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

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

  reference "RFC XXXX";

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

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

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

/*
 * Groupings
 */
grouping encap-cfg {
  description "Configuration for BFD encapsulation";

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

grouping mpls-dest-address {
  description "Destination address as per RFC 5884.";

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

/*
 * Augments
 */
augment "/rt:routing/rt:control-plane-protocols/"
  + "rt:control-plane-protocol/bfd:bfd" {
  description "BFD augmentation for MPLS.";
  container mpls {
    description "BFD MPLS top level container.";

    uses bfd-types:session-statistics-summary;

    container egress {
      description "Egress configuration.";

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

```

    uses bfd-types:auth-parms;
  }

  container session-groups {
    description
      "BFD over MPLS session groups.";
    list session-group {
      key "mpls-fec";
      description
        "Group of BFD MPLS sessions (for ECMP). A group of " +
        "sessions is for 1 FEC, each session has a different " +
        "field in UDP/IP hdr for ECMP.";
      leaf mpls-fec {
        type inet:ip-prefix;
        description "MPLS FEC.";
      }

      uses bfd-types:common-cfg-parms;

      list sessions {
        config false;
        description
          "The BFD sessions for an MPLS FEC. Local " +
          "discriminator is unique for each session in the " +
          "group.";
        uses bfd-types:all-session;

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

/*
 * Notifications
 */
notification mpls-notification {
  description
    "Notification for BFD over MPLS FEC session state change. " +
    "An implementation may rate-limit notifications, e.g. when a " +
    "session is continuously changing state.";

  uses bfd-types:notification-parms;

  leaf mpls-dest-address {
    type inet:ip-address;
    description

```



```
        "Destination address as per RFC 5884.  
        Needed if IP encapsulation is used.";  
    }  
}  
}
```

<CODE ENDS>

2.19. BFD over MPLS-TE YANG Module

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

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

```
module ietf-bfd-mpls-te {  
  
    yang-version 1.1;  
  
    namespace "urn:ietf:params:xml:ns:yang:ietf-bfd-mpls-te";  
  
    prefix "bfd-mpls-te";  
  
    // RFC Ed.: replace occurrences of XXXX with actual RFC number and  
    // remove this note  
  
    import ietf-bfd-types {  
        prefix "bfd-types";  
        reference "RFC XXXX: YANG Data Model for BFD";  
    }  
  
    import ietf-bfd {  
        prefix "bfd";  
        reference "RFC XXXX: YANG Data Model for BFD";  
    }  
  
    import ietf-bfd-mpls {  
        prefix "bfd-mpls";  
        reference "RFC XXXX: YANG Data Model for BFD";  
    }  
  
    import ietf-te {  
        prefix "te";  
        // RFC Ed.: replace YYYY with actual RFC number of  
        // draft-ietf-teas-yang-te and remove this note.  
        reference  
            "RFC YYYY: A YANG Data Model for Traffic Engineering Tunnels and  
            Interfaces";  
    }  
}
```

```
}

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

organization "IETF BFD Working Group";

contact
  "WG Web:    <http://tools.ietf.org/wg/bfd>
  WG List:    <rtg-bfd@ietf.org>

  Editors:    Reshad Rahman (rrahman@cisco.com),
               Lianshu Zheng (vero.zheng@huawei.com),
               Mahesh Jethanandani (mjethanandani@gmail.com)";

description
  "This module contains the YANG definition for BFD parameters for
  MPLS Traffic Engineering as per RFC 5884.

  Copyright (c) 2018 IETF Trust and the persons
  identified as authors of the code.  All rights reserved.

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

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

reference "RFC XXXX";

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

/*
 * Augments
 */
augment "/rt:routing/rt:control-plane-protocols/"
  + "rt:control-plane-protocol/bfd:bfd" {
```

```
description "BFD augmentation for MPLS-TE.";
container mpls-te {
  description "BFD MPLS-TE top level container.";

  container egress {
    description "Egress configuration.";

    uses bfd-types:client-cfg-parms;

    uses bfd-types:auth-parms;
  }

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

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

  uses bfd-types:common-cfg-parms;

  uses bfd-mpls:encap-cfg;
}

augment "/te:te/te:lsps-state/te:lsp" {
  when "/te:te/te:lsps-state/te:lsp/te:origin-type != 'transit'" {
    description "BFD information not needed at transit points.";
  }
  description "BFD state information on MPLS-TE LSP.";

  uses bfd-types:all-session;

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

/*
 * Notifications
 */
notification mpls-te-notification {
  description
    "Notification for BFD over MPLS-TE session state change. " +
    "An implementation may rate-limit notifications, e.g. when a " +
    "session is continuously changing state.";

  uses bfd-types:notification-parms;

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

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

<CODE ENDS>

3. Data Model examples

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

3.1. IP single-hop

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

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

3.2. IP multihop

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

```

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

```

3.3. LAG

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

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

3.4. MPLS

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

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

4. Security Considerations

The YANG module specified in this document defines a schema for data that is designed to be accessed via network management protocols such as NETCONF [RFC6241] or RESTCONF [RFC8040]. The lowest NETCONF layer is the secure transport layer, and the mandatory-to-implement secure transport is Secure Shell (SSH) [RFC6242]. The lowest RESTCONF layer is HTTPS, and the mandatory-to-implement secure transport is TLS [RFC5246].

The NETCONF access control model [RFC6536] provides the means to restrict access for particular NETCONF or RESTCONF users to a preconfigured subset of all available NETCONF or RESTCONF protocol operations and content.

There are a number of data nodes defined in this YANG module that are writable/creatable/deletable (i.e., config true, which is the

default). These data nodes may be considered sensitive or vulnerable in some network environments. Write operations (e.g., edit-config) to these data nodes without proper protection can have a negative effect on network operations. These are the subtrees and data nodes and their sensitivity/vulnerability:

/routing/control-plane-protocols/control-plane-protocol/bfd/ip-sh/
sessions: the list specifies the IP single-hop BFD sessions.

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

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

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

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

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

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

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

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

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

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

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

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

Some of the readable data nodes in this YANG module may be considered sensitive or vulnerable in some network environments. It is thus important to control read access (e.g., via get, get-config, or

notification) to these data nodes. These are the subtrees and data nodes and their sensitivity/vulnerability:

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

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

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

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

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

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

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

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

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

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

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

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

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

5. IANA Considerations

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

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

Registrant Contact: The IESG.

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

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

Registrant Contact: The IESG.

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

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

Registrant Contact: The IESG.

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

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

Registrant Contact: The IESG.

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

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

Registrant Contact: The IESG.

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

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

Registrant Contact: The IESG.

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

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

Registrant Contact: The IESG.

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

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

Registrant Contact: The IESG.

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

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

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

Name: iana-bfd-types

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

Prefix: iana-bfd-types

Reference: RFC XXXX

Name: ietf-bfd-types

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

Prefix: bfd-types

Reference: RFC XXXX

Name: ietf-bfd

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

Prefix: bfd

Reference: RFC XXXX

Name: ietf-bfd-ip-sh

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

Prefix: bfd-ip-sh

Reference: RFC XXXX

Name: ietf-bfd-ip-mh

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

Prefix: bfd-ip-mh

Reference: RFC XXXX

Name: ietf-bfd-lag

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

Prefix: bfd-lag

Reference: RFC XXXX

Name: ietf-bfd-mpls

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

Prefix: bfd-mpls

Reference: RFC XXXX

Name: ietf-bfd-mpls-te

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

Prefix: bfd-mpls-te

Reference: RFC XXXX

5.1. IANA-Maintained iana-bfd-types module

This document defines the initial version of the IANA-maintained iana-bfd-types YANG module.

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

6. Acknowledgements

We would also like to thank Nobo Akiya and Jeff Haas for their encouragement on this work. We would also like to thank Rakesh Gandhi and Tarek Saad for their help on the MPLS-TE model. We would also like to thank Acee Lindem for his guidance.

7. References

7.1. Normative References

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

- [I-D.ietf-teas-yang-te]
Saad, T., Gandhi, R., Liu, X., Beeram, V., Shah, H., and
I. Bryskin, "A YANG Data Model for Traffic Engineering
Tunnels and Interfaces", draft-ietf-teas-yang-te-16 (work
in progress), July 2018.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate
Requirement Levels", BCP 14, RFC 2119,
DOI 10.17487/RFC2119, March 1997,
<<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688,
DOI 10.17487/RFC3688, January 2004,
<<https://www.rfc-editor.org/info/rfc3688>>.
- [RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security
(TLS) Protocol Version 1.2", RFC 5246,
DOI 10.17487/RFC5246, August 2008,
<<https://www.rfc-editor.org/info/rfc5246>>.
- [RFC5586] Bocci, M., Ed., Vigoureux, M., Ed., and S. Bryant, Ed.,
"MPLS Generic Associated Channel", RFC 5586,
DOI 10.17487/RFC5586, June 2009,
<<https://www.rfc-editor.org/info/rfc5586>>.
- [RFC5880] Katz, D. and D. Ward, "Bidirectional Forwarding Detection
(BFD)", RFC 5880, DOI 10.17487/RFC5880, June 2010,
<<https://www.rfc-editor.org/info/rfc5880>>.
- [RFC5881] Katz, D. and D. Ward, "Bidirectional Forwarding Detection
(BFD) for IPv4 and IPv6 (Single Hop)", RFC 5881,
DOI 10.17487/RFC5881, June 2010,
<<https://www.rfc-editor.org/info/rfc5881>>.
- [RFC5882] Katz, D. and D. Ward, "Generic Application of
Bidirectional Forwarding Detection (BFD)", RFC 5882,
DOI 10.17487/RFC5882, June 2010,
<<https://www.rfc-editor.org/info/rfc5882>>.
- [RFC5883] Katz, D. and D. Ward, "Bidirectional Forwarding Detection
(BFD) for Multihop Paths", RFC 5883, DOI 10.17487/RFC5883,
June 2010, <<https://www.rfc-editor.org/info/rfc5883>>.
- [RFC5884] Aggarwal, R., Kompella, K., Nadeau, T., and G. Swallow,
"Bidirectional Forwarding Detection (BFD) for MPLS Label
Switched Paths (LSPs)", RFC 5884, DOI 10.17487/RFC5884,
June 2010, <<https://www.rfc-editor.org/info/rfc5884>>.

- [RFC5885] Nadeau, T., Ed. and C. Pignataro, Ed., "Bidirectional Forwarding Detection (BFD) for the Pseudowire Virtual Circuit Connectivity Verification (VCCV)", RFC 5885, DOI 10.17487/RFC5885, June 2010, <<https://www.rfc-editor.org/info/rfc5885>>.
- [RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, DOI 10.17487/RFC6020, October 2010, <<https://www.rfc-editor.org/info/rfc6020>>.
- [RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <<https://www.rfc-editor.org/info/rfc6241>>.
- [RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure Shell (SSH)", RFC 6242, DOI 10.17487/RFC6242, June 2011, <<https://www.rfc-editor.org/info/rfc6242>>.
- [RFC6536] Bierman, A. and M. Bjorklund, "Network Configuration Protocol (NETCONF) Access Control Model", RFC 6536, DOI 10.17487/RFC6536, March 2012, <<https://www.rfc-editor.org/info/rfc6536>>.
- [RFC6991] Schoenwaelder, J., Ed., "Common YANG Data Types", RFC 6991, DOI 10.17487/RFC6991, July 2013, <<https://www.rfc-editor.org/info/rfc6991>>.
- [RFC7130] Bhatia, M., Ed., Chen, M., Ed., Boutros, S., Ed., Binderberger, M., Ed., and J. Haas, Ed., "Bidirectional Forwarding Detection (BFD) on Link Aggregation Group (LAG) Interfaces", RFC 7130, DOI 10.17487/RFC7130, February 2014, <<https://www.rfc-editor.org/info/rfc7130>>.
- [RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017, <<https://www.rfc-editor.org/info/rfc8040>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [RFC8177] Lindem, A., Ed., Qu, Y., Yeung, D., Chen, I., and J. Zhang, "YANG Data Model for Key Chains", RFC 8177, DOI 10.17487/RFC8177, June 2017, <<https://www.rfc-editor.org/info/rfc8177>>.

- [RFC8340] Bjorklund, M. and L. Berger, Ed., "YANG Tree Diagrams", BCP 215, RFC 8340, DOI 10.17487/RFC8340, March 2018, <<https://www.rfc-editor.org/info/rfc8340>>.
- [RFC8343] Bjorklund, M., "A YANG Data Model for Interface Management", RFC 8343, DOI 10.17487/RFC8343, March 2018, <<https://www.rfc-editor.org/info/rfc8343>>.
- [RFC8344] Bjorklund, M., "A YANG Data Model for IP Management", RFC 8344, DOI 10.17487/RFC8344, March 2018, <<https://www.rfc-editor.org/info/rfc8344>>.
- [RFC8349] Lhotka, L., Lindem, A., and Y. Qu, "A YANG Data Model for Routing Management (NMDA Version)", RFC 8349, DOI 10.17487/RFC8349, March 2018, <<https://www.rfc-editor.org/info/rfc8349>>.

7.2. Informative References

- [I-D.ietf-lime-yang-connectionless-oam]
Kumar, D., Wang, Z., Wu, Q., Rahman, R., and S. Raghavan, "Generic YANG Data Model for the Management of Operations, Administration, and Maintenance (OAM) Protocols that use Connectionless Communications", draft-ietf-lime-yang-connectionless-oam-18 (work in progress), November 2017.
- [I-D.ietf-rtgwg-lne-model]
Berger, L., Hopps, C., Lindem, A., Bogdanovic, D., and X. Liu, "YANG Model for Logical Network Elements", draft-ietf-rtgwg-lne-model-10 (work in progress), March 2018.
- [I-D.ietf-rtgwg-ni-model]
Berger, L., Hopps, C., Lindem, A., Bogdanovic, D., and X. Liu, "YANG Model for Network Instances", draft-ietf-rtgwg-ni-model-12 (work in progress), March 2018.
- [RFC8342] Bjorklund, M., Schoenwaelder, J., Shafer, P., Watsen, K., and R. Wilton, "Network Management Datastore Architecture (NMDA)", RFC 8342, DOI 10.17487/RFC8342, March 2018, <<https://www.rfc-editor.org/info/rfc8342>>.

Appendix A. Echo function configuration example

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

```
module: example-bfd-echo
  augment /rt:routing/rt:control-plane-protocols
    /rt:control-plane-protocol/bfd:bfd/bfd-ip-sh:ip-sh
    /bfd-ip-sh:sessions:
    +--rw echo {bfd-types:echo-mode}?
      +--rw desired-min-echo-tx-interval?   uint32
      +--rw required-min-echo-rx-interval?   uint32
```

A.1. Example YANG module for BFD echo function configuration

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

  prefix "example-bfd-echo";

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

  import ietf-bfd {
    prefix "bfd";
  }

  import ietf-bfd-ip-sh {
    prefix "bfd-ip-sh";
  }

  import ietf-routing {
    prefix "rt";
  }

  organization "IETF BFD Working Group";

  contact
    "WG Web:    <http://tools.ietf.org/wg/bfd>
    WG List:    <rtg-bfd@ietf.org>

    Editors:    Reshad Rahman (rrahman@cisco.com),
                 Lianshu Zheng (vero.zheng@huawei.com),
                 Mahesh Jethanandani (mjethanandani@gmail.com)";

  description
    "This module contains an example YANG augmentation for configuration
    of BFD echo function.

    Copyright (c) 2018 IETF Trust and the persons
    identified as authors of the code. All rights reserved."
```

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

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

```
revision 2018-08-01 {
  description "Initial revision.";
  reference
    "RFC XXXX: A YANG data model example augmentation for BFD echo
    function";
}

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

/*
 * Groupings
 */
grouping echo-cfg-parms {
  description "BFD grouping for echo config parameters";
  leaf desired-min-echo-tx-interval {
    type uint32;
    units microseconds;
    default 0;
    description
      "This is the minimum interval that the local system would like
      to use when transmitting BFD echo packets. If 0, the echo
      function as defined in BFD [RFC5880] is disabled.";
  }

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

augment "/rt:routing/rt:control-plane-protocols/"
  + "rt:control-plane-protocol/bfd:bfd/bfd-ip-sh:ip-sh/"
  + "bfd-ip-sh:sessions" {
```

```
    description "Augmentation for BFD echo function.";
    container echo {
      if-feature bfd-types:echo-mode;

      description "BFD echo function container";

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

Appendix B. Change log

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

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

- o Addressed IESG comments.

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

- o Added list of modules for YANG module registry.

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

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

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

- o Addressed missing/incorrect references in import statements.

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

- o Updated references for drafts which became RFCs recently.

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

- o Addressed comments from YANG Doctor review of rev11.

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

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

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

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

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

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

B.10. Changes between versions -07 and -08

- o Timer intervals in client-cfg-parms are not mandatory anymore.
- o Added list of interfaces under "ip-sh" node for authentication parameters.
- o Renamed replay-protection to meticulous.

B.11. Changes between versions -06 and -07

- o New ietf-bfd-types module.
- o Grouping for BFD clients to have BFD multiplier and interval values.
- o Change in ietf-bfd-mpls-te since MPLS-TE model changed.
- o Removed bfd- prefix from many names.

B.12. Changes between versions -05 and -06

- o Adhere to NMDA-guidelines.
- o Echo function config moved to appendix as example.
- o Added IANA YANG modules.
- o Addressed various comments.

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

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

B.14. Changes between versions -03 and -04

- o Updated author information.
- o Fixed YANG compile error in ietf-bfd-lag.yang which was due to incorrect when statement.

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

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

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

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

B.17. Changes between versions -00 and -01

- o Remove BFD configuration parameters from BFD clients, all BFD configuration parameters in BFD
- o YANG module split in multiple YANG modules (one per type of forwarding path)
- o For BFD over MPLS-TE we augment MPLS-TE model
- o For BFD authentication we now use YANG Data Model for Key Chains [RFC8177]

Authors' Addresses

Reshad Rahman (editor)
Cisco Systems
Canada

Email: rrahman@cisco.com

Lianshu Zheng (editor)
Huawei Technologies
China

Email: vero.zheng@huawei.com

Mahesh Jethanandani (editor)
Xoriant Corporation
1248 Reamwood Ave
Sunnyvale, California 94089
USA

Email: mjethanandani@gmail.com

Santosh Pallagatti
Rtbrick
India

Email: santosh.pallagatti@gmail.com

Greg Mirsky
ZTE Corporation

Email: gregimirsky@gmail.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: March 25, 2021

R. Bush
Internet Initiative Japan
J. Haas
J. Scudder
Juniper Networks, Inc.
A. Nipper
C. Dietzel
DE-CIX
September 21, 2020

Making Route Servers Aware of Data Link Failures at IXPs
draft-ietf-idr-rs-bfd-09

Abstract

When BGP route servers are used, the data plane is not congruent with the control plane. Therefore, peers at an Internet exchange can lose data connectivity without the control plane being aware of it, and packets are lost. This document proposes the use of a newly defined BGP Subsequent Address Family Identifier (SAFI) both to allow the route server to request its clients use BFD to track data plane connectivity to their peers' addresses, and for the clients to signal that connectivity state back to the route server.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" are to be interpreted as described in [RFC2119] only when they appear in all upper case. They may also appear in lower or mixed case as English words, without normative meaning.

Status of This Memo

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

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

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

This Internet-Draft will expire on March 25, 2021.

Copyright Notice

Copyright (c) 2020 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 (<https://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	2
2. Definitions	3
3. Overview	4
4. Next Hop Validation	5
4.1. ReachAsk	6
4.2. LocReach	6
4.3. ReachTell	7
4.4. NHIB	7
5. Advertising NH-Reach state in BGP	7
6. Client Procedures for NH-Reach Changes	9
7. Recommendations for Using BFD	9
8. Other Considerations	10
9. Acknowledgments	10
10. IANA Considerations	10
11. Security Considerations	10
12. References	11
12.1. Normative References	11
12.2. Informative References	12
Appendix A. Summary of Document Changes	12
Appendix B. Other Forms of Connectivity Checks	12
Authors' Addresses	13

1. Introduction

In configurations (typically Internet Exchange Points (IXPs)) where EBGp routing information is exchanged between client routers through the agency of a route server (RS) [RFC7947], but traffic is exchanged directly, operational issues can arise when partial data plane connectivity exists among the route server client routers. Since the

data plane is not congruent with the control plane, the client routers on the IXP can lose data connectivity without the control plane - the route server - being aware of it, resulting in significant data loss.

To remedy this, two basic problems need to be solved:

1. Client routers must have a means of verifying connectivity amongst themselves, and
2. Client routers must have a means of communicating the knowledge of the failure (and restoration) back to the route server.

The first can be solved by application of Bidirectional Forwarding Detection [RFC5880]. The second can be solved by exchanging BGP routes which use the NH-Reach Subsequent Address Family Identifier (SAFI) defined in this document.

Throughout this document, we generally assume that the route server being discussed is able to represent different RIBs towards different clients, as discussed in section 2.3.2.1 of [RFC7947]. If this is not the case, the procedures described here to allow BFD to be automatically provisioned between clients still have value; however, the procedures for signaling reachability back to the route server may not.

Throughout this document, we refer to the "route server", "RS" or just "server" and the "client" to describe the two BGP routers engaging in the exchange of information. We observe that there could be other applications for this extension. Our use of terminology is intended for clarity of description, and not to limit the future applicability of the proposal.

[I-D.ietf-idr-bgp-bestpath-selection-criteria] discusses enhancement of the route resolvability condition of section 9.1.2.1 of [RFC4271] to include next hop reachability and path availability checks. This specification represents in part an instance of such, implemented using BFD as the OAM mechanism.

2. Definitions

- o Indirect peer: If a route server is configured such that routes from a given client might be sent to some other client, or vice-versa, those two clients are considered to be indirect peers.
- o Indirect Peer's Address, IPA, next hop: We refer frequently to a next hop. It should generally be clear from context what is intended, almost always an address associated with an indirect peer (the exception, when an indirect peer sends a third party next hop, is discussed in Section 3). In Section 5 we discuss the

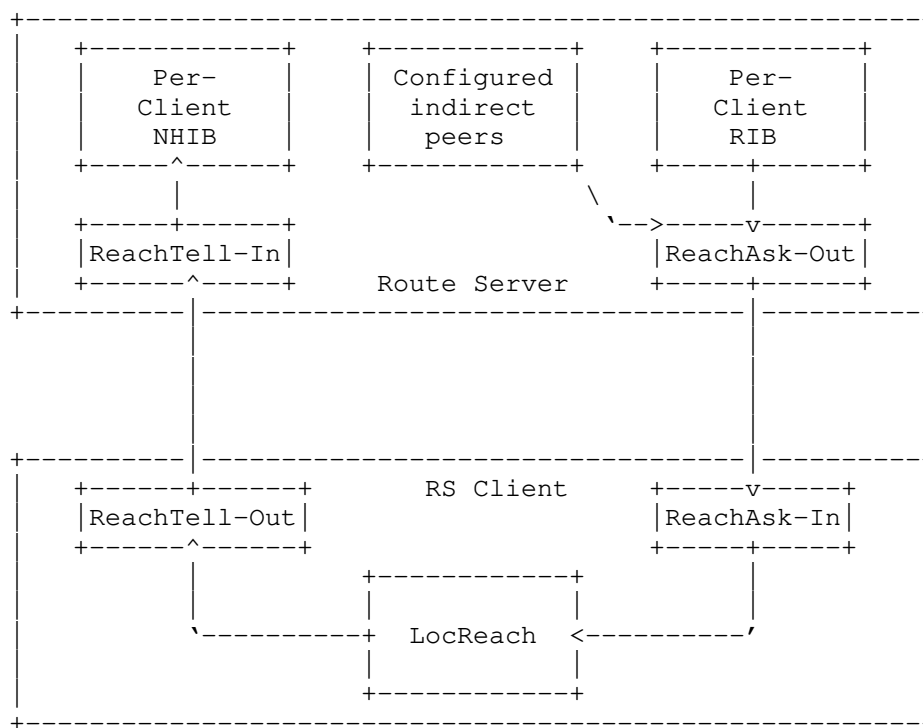
MP-BGP [RFC4760] Next Hop field; this is distinguished by its capitalization and should also be clear from context. Later in that section we define the Indirect Peer's Address field of the NLRI, also called "IPA". It will be clear to the reader that this refers to the "next hops" discussed elsewhere in the document, but we don't use the name "next hop" for this field to avoid confusion with the pre-existing next hop path attribute of [RFC4271] and attribute field of [RFC4760].

- o RS: Route Server. See [RFC7947].

3. Overview

As with the base BGP protocol, we model the function of this extension as the interaction between a conceptual set of databases:

- o ReachAsk: The reachability request database. A database of next hops (host addresses) for which data plane reachability is being queried.
- o ReachAsk-Out: A set of queries sent to the client.
- o ReachAsk-In: A set of queries received from the route server.
- o ReachTell: The reachability response database. A database of responses to ReachAsk queries, indicating what is known about data plane reachability.
- o ReachTell-Out: The responses being sent to the route server.
- o ReachTell-In: The response received from the client.
- o LocReach: The local reachability database.
- o NHIB: Next Hop Information Base. Stores what is known about the client's reachability to its next hops.



Route Server, RS Client, and Reachability Ask and Tell databases with In/Out Queues

In outline, the route server requests its client to track connectivity for all the potential next hops the RS might send to the client, by sending these next hops as ReachAsk "routes". The client tracks connectivity using BFD and reports its connectivity status to the RS using ReachTell "routes". Connectivity status may be that the next hop is reachable, unreachable, or unknown. Once the RS has been informed by the client of its connectivity, it uses this information to influence the route selection the RS performs on behalf of the client. Details are elaborated in the following sections.

4. Next Hop Validation

Below, we detail procedures where a route server tells its client router about other client next hops by sending it ReachAsk routes and the client router verifies connectivity to those other client routers and communicates its findings back to the RS using ReachTell routes. The RS uses the received ReachTell routes as input to the NHIB and hence the route selection process it performs on behalf of the client.

4.1. ReachAsk

The route server maintains a ReachAsk database for each client that supports this proposal, that is, for each client that has advertised support (Section 5) for the NH-Reach SAFI. This database is the union of:

- o The set of next hops found in the associated per-client Loc-RIB (see section 2.3.2.1 of [RFC7947]).
- o The set of addresses of this client's indirect peers (Section 2).
- o The RS MAY also add other entries, for example under configuration control.

We note that under most circumstances, the first (Loc-RIB next hops) set will be a subset of the second (indirect peers) set. For this not to be the case, a client would have to have sent a "third party" next hop [RFC4271] to the server. To cover such a case, an implementation MAY note any such next hops, and include them in its list of indirect peers. (This implies that if a third party next hop for client C is conveyed to client A, not only will C be placed in A's ReachAsk database, but A will be placed in C's ReachAsk database.)

The contents of the ReachAsk database are communicated to the client using the NLRI format and procedures described in Section 5.

4.2. LocReach

The client MUST attempt to track data plane connectivity to each host address depicted in the ReachAsk database. It MAY also track connectivity to other addresses. The use of BFD for this purpose is detailed in Section 6.

For each address being tracked, its state is maintained by the client in a LocReach entry. The state can be:

- o Unknown. Connectivity status is unknown. This may be due to a temporary or permanent lack of feasible OAM mechanism to determine the status.
- o Up. The address has been determined to be reachable.
- o Down. The address has been determined to be unreachable.

The LocReach database is used as input for the ReachTell database; it MAY also be used as input to the client's route resolvability condition (section 9.1.2.1 of [RFC4271]).

4.3. ReachTell

The ReachTell database contains an entry for every entry in the LocReach database.

The contents of the ReachTell database are communicated to the server using the NLRI format and procedures described in Section 5.

4.4. NHIB

The route server maintains a per-client Next Hop Information Base, or NHIB. This contains the information about next hop status received from ReachTell.

In computing its per-client Loc-RIB, the RS uses the content of the related per-client NHIB as input to the route resolvability condition (section 9.1.2.1 of [RFC4271]). The next hop being resolved is looked up in the NHIB and its state determined:

- o Up next hops are considered resolvable.
- o Unknown next hops MAY be considered resolvable. They MAY be less preferred for selection.
- o Down next hops MUST NOT be considered resolvable.
- o If a given next hop is not present in the NHIB, but is present in ReachAsk-Out, either the client has not responded yet (a transient condition) or an error exists. Similar to Unknown next hops, such routes MAY be considered resolvable; they MAY be less preferred.

5. Advertising NH-Reach state in BGP

A new BGP SAFI, the NH-Reach SAFI, is defined in this document. It has been assigned value TBD. A route server or a route server client using the procedures in this document MUST advertise support for this SAFI, for the IPv4 and/or IPv6 Address Family Identifier (AFI). The use of this SAFI with any other AFI is not defined by this document.

NH-Reach NLRI "routes" have a Length of Next Hop Network Address value of 0, therefore they have an empty Network Address of Next Hop field (section 3 of [RFC4760]).

Since as specified here, ReachTell "routes" from different clients populate distinct databases on the RS, there will generally be only a single path per "route"; this implies that route selection need not be performed (or equivalently, that it's trivial to perform).

In the other direction, a client might peer with multiple route servers and receive differing sets of ReachAsk routes from them. An implementation MAY handle this situation by implementing a distinct

ReachAsk and ReachTell per server, but it MAY also handle it by placing all servers' ReachAsk "routes" into a single ReachAsk, and sending the results to all servers from a single ReachTell. This would imply some route server(s) might get ReachTell results they had not asked for, but this is permissible in any case. Again, since the contents of ReachAsk are simply a set of host routes to be tested, route selection over a combined ReachAsk MAY be omitted.

ReachAsk and ReachTell entries are exchanged using the NH-Reach NLRI encoding:

```

      0               1               2               3
      0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
|T|Reserved|Sta|      Indirect Peer's Address (4 or 16 octets)      |
+-----+-----+-----+-----+-----+-----+-----+-----+
.      ... Indirect Peer's Address (4 or 16 octets) ...      .
.
+-----+-----+-----+-----+-----+-----+-----+-----+

```

NH-Reach NLRI Format

- o T: Type is a one-bit field that can take the value 0, meaning the NLRI is a ReachAsk entry, or 1, meaning it is a ReachTell entry.
- o Reserved: These five bits are reserved. They MUST be sent as zero and MUST be disregarded on receipt.
- o Sta: State is a two-bit field used to signal the LocReach (Section 4.2) state:
 - * 0 or 3: Unknown.
 - * 1: Up.
 - * 2: Down.

Although either 0 or 3 is to be interpreted as "Unknown", the value 0 MUST be used on transmission. The value 3 MUST be accepted as an alias for 0 on receipt.

- o The Indirect Peer's Address ("IPA") field is an IPv4 or IPv6 host route, depending on whether the AFI is IPv4 or IPv6.

ReachAsk and ReachTell entries MUST NOT be propagated from one BGP peering session to another; the routes are not transitive.

The IPA field is the key for the NH-Reach NLRI type; the information encoded in the top octet is non-key information. It is possible in principle (although unlikely) for two NLRI to be validly present in an UPDATE message with identical IPA fields but different types. However, two NLRI with the same IPA field and different State fields MUST NOT be encoded in the same UPDATE message. If such is

encountered, the receiver MUST behave as though the state "Unknown" was received for the IPA in question.

6. Client Procedures for NH-Reach Changes

When an entry is added to a route server client's ReachAsk-In for a route server peering session, the client will then attempt to verify connectivity to the host depicted by that entry. The procedure described in this specification utilizes BFD.

If no existing BFD session exists to this next hop, a BFD session is provisioned to that IP address and the LocReach reachability state (Section 4.2) is set to Unknown.

If the client cannot establish a BFD session with an entry in its ReachAsk-In, the next hop remains in LocReach with its Reachable state Unknown.

Once the BFD session moves to the Up state, the LocReach reachability state is set to Up.

When the BFD session transitions out of the Up state to the Down state, the LocReach reachability state is set to Down.

If the BFD session transitions out of the Up state to the AdminDown state, the LocReach reachability state is set to Unknown.

When entries are removed from the route server client's ReachAsk-In for a route server peering session, the client MAY delay de-provisioning the BFD peering session. If the client delays de-provisioning the session, it should remove it if the BFD session transitions to the Down or AdminDown states.

7. Recommendations for Using BFD

The RECOMMENDED way a client router can confirm the data plane connectivity to its next hops is available, is the use of BFD in asynchronous mode. Echo mode MAY be used if both client routers running a BFD session support this. The use of authentication in BFD is OPTIONAL as there is a certain level of trust between the operators of the client routers at a particular IXP. If trust cannot be assumed, it is recommended to use pair-wise keys (how this can be achieved is outside the scope of this document). The ttl/hop limit values as described in section 5 [RFC5881] MUST be obeyed in order to shield BFD sessions against packets coming from outside the IXP.

The following values of the BFD configuration of client routers (see section 6.8.1 [RFC5880]) are RECOMMENDED:

- o DesiredMinTxInterval: 1,000,000 (microseconds)
- o RequiredMinRxInterval: 1,000,000 (microseconds)
- o DetectMult: 3

A client router administrator MAY select more appropriate values to meet the special needs of a particular deployment.

8. Other Considerations

For purposes of routing stability, implementations may wish to apply hysteresis ("holddown") to next hops that have transitioned from reachable to unreachable and back.

Implementations MAY restrict the range of addresses with which they will attempt to form BFD relationships. For example, an implementation might by default only allow BFD relationships with peers that share a subnet with the route server. An implementation MAY apply such restrictions by default.

In a route-server environment, use of this feature SHOULD be restricted to consider only routes that are advertised from within the IXP network. This might include checks on AS_PATH length.

9. Acknowledgments

The authors would like to thank Thomas King for his contributions toward this work.

10. IANA Considerations

IANA is requested to allocate a value from the Subsequent Address Family Identifiers (SAFI) Parameters registry for this proposal. Its Description in that registry shall be NH-Reach with a Reference of this RFC.

11. Security Considerations

The mechanism in this document permits a route server client to influence the contents of the route server's Adj-Ribs-Out through its reports of next hop reachability state using the NH-Reach SAFI. Since this state is per-client, if a route server client is able to inject NH-Reach routes for another route server's BGP session to a client, it can cause the route server to select different forwarding than otherwise expected. This issue may be mitigated using transport security on the BGP sessions between the route server and its clients. See [RFC4272].

The NH-Reach SAFI enables the server to trigger creation of a BFD session on its client. A malicious or misbehaving server could trigger an unreasonable number of sessions, a potential resource exhaustion attack. The sedate default timers proposed in Section 7 mitigate this; they also mitigate concerns about use of the client as a source of packets in a flooding attack. An implementation MAY also impose limits on the number of BFD sessions it will create at the request of the server.

The reachability tests between route server clients themselves may be a target for attack. Such attacks may include forcing a BFD session Down through injecting false BFD state. A less likely attack includes forcing a BFD session to stay Up when its real state is Down. These attacks may be mitigated using the BFD security mechanisms defined in [RFC5880].

12. References

12.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC4271] Rekhter, Y., Ed., Li, T., Ed., and S. Hares, Ed., "A Border Gateway Protocol 4 (BGP-4)", RFC 4271, DOI 10.17487/RFC4271, January 2006, <<https://www.rfc-editor.org/info/rfc4271>>.
- [RFC4760] Bates, T., Chandra, R., Katz, D., and Y. Rekhter, "Multiprotocol Extensions for BGP-4", RFC 4760, DOI 10.17487/RFC4760, January 2007, <<https://www.rfc-editor.org/info/rfc4760>>.
- [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>>.
- [RFC7947] Jasinska, E., Hilliard, N., Raszuk, R., and N. Bakker, "Internet Exchange BGP Route Server", RFC 7947, DOI 10.17487/RFC7947, September 2016, <<https://www.rfc-editor.org/info/rfc7947>>.

12.2. Informative References

- [I-D.chen-bfd-unsolicited]
Chen, E., Shen, N., and R. Raszuk, "Unsolicited BFD for Sessionless Applications", draft-chen-bfd-unsolicited-02 (work in progress), January 2018.
- [I-D.ietf-idr-bgp-bestpath-selection-criteria]
Asati, R., "BGP Bestpath Selection Criteria Enhancement", draft-ietf-idr-bgp-bestpath-selection-criteria-12 (work in progress), June 2019.
- [RFC4272] Murphy, S., "BGP Security Vulnerabilities Analysis", RFC 4272, DOI 10.17487/RFC4272, January 2006, <<https://www.rfc-editor.org/info/rfc4272>>.
- [RFC7880] Pignataro, C., Ward, D., Akiya, N., Bhatia, M., and S. Pallagatti, "Seamless Bidirectional Forwarding Detection (S-BFD)", RFC 7880, DOI 10.17487/RFC7880, July 2016, <<https://www.rfc-editor.org/info/rfc7880>>.

Appendix A. Summary of Document Changes

idr-06: Refresh -05.
idr-04 to idr-05: Added reference to "BGP Bestpath Selection Criteria Enhancement" draft. Rename "next hop" field of NLRI to "Indirect Peer's Address". Add suggestion about AS_PATH length checks.
idr-03 to idr-04: Note other forms of connectivity checks.
idr-02 to idr-03: Substantial rewrite. Introduce NLRI format that embeds state.
idr-01 to idr-02: Move from BGP-LS to NH-Reach SAFI. Lots of editorial changes.
idr-00 to idr-01: Add BGP Capability. Move from NH-Cost to BGP-LS.
ymbk-01 to idr-00: No technical changes; adopted by IDR.
ymbk-00 to ymbk-01: Clarifications to BFD procedures. Use BFD state as an input to BGP route selection.

Appendix B. Other Forms of Connectivity Checks

RFC 5880/5881 BFD is a well-deployed feature. For this reason, it was chosen as the connectivity check utilized for nexthop reachability by this document. As other forms of BFD become more widely deployed, they may also be utilized to provide the connectivity check functionality.

Examples of other such BFD mechanisms include:

- o Seamless BFD [RFC7880]
- o Unsolicited BFD for Sessionless Applications
[I-D.chen-bfd-unsolicited]

Implementations MUST support RFC 5880/5881 BFD to be compliant with this specification. Implementations MAY support other forms of connectivity check, including those mechanisms listed above, so long as they provide the ability to fall-back to RFC 5880/5881 BFD.

Authors' Addresses

Randy Bush
Internet Initiative Japan
5147 Crystal Springs
Bainbridge Island, Washington 98110
US

Email: randy@psg.com

Jeffrey Haas
Juniper Networks, Inc.
1133 Innovation Way
Sunnyvale, CA 94089
US

Email: jhaas@juniper.net

John G. Scudder
Juniper Networks, Inc.
1133 Innovation Way
Sunnyvale, CA 94089
US

Email: jgs@juniper.net

Arnold Nipper
DE-CIX Management GmbH
Lichtstrasse 43i
Cologne 50825
Germany

Email: arnold.nipper@de-cix.net

Internet-Draft Making RSeS aware of IXP Data Link FailuresSeptember 2020

Christoph Dietzel
DE-CIX Management GmbH
Lichtstrasse 43i
Cologne 50825
Germany

Email: christoph.dietzel@de-cix.net

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: August 2, 2018

N. Gupta
A. Dogra
Cisco Systems, Inc.
C. Docherty
AT&T
G. Mirsky
J. Tantsura
Individual
January 29, 2018

Fast failure detection in VRRP with Point to Point BFD
draft-ietf-rtgwg-vrrp-bfd-p2p-00

Abstract

This document describes how Point to Point Bidirectional Forwarding Detection (BFD) can be used to support sub-second detection of a Master Router failure in the Virtual Router Redundancy Protocol (VRRP).

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 August 2, 2018.

Copyright Notice

Copyright (c) 2018 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. Requirements Language	4
3. Applicability of Point to Point BFD	5
3.1. Extension to VRRP protocol	5
3.2. VRRP Peer Table	6
3.3. VRRP BACKUP ADVERTISEMENT Packet Type	7
3.4. Sample configuration	8
3.5. Critical BFD session	9
3.6. Protocol State Machine	9
3.6.1. Parameters Per Virtual Router	9
3.6.2. Timers	10
3.6.3. VRRP State Machine with Point to Point BFD	10
4. Scalability Considerations	20
5. Operational Considerations	21
6. Applicability to VRRPv2	22
7. IANA Considerations	23
7.1. A New Name Space for VRRP Packet Types	23
8. Security Considerations	24
9. Acknowledgements	25
10. Normative References	26
Authors' Addresses	27

1. Introduction

The Virtual Router Redundancy Protocol (VRRP) provides redundant Virtual gateways in the Local Area Network (LAN), which is typically the first point of failure for end-hosts sending traffic out of the LAN. Fast failure detection of VRRP Master is critical in supporting high availability of services and improved Quality of Experience to users. In VRRP [RFC5798] specification, Backup routers depend on VRRP packets generated at a regular interval by the Master router, to detect the health of the VRRP Master. Faster failure detection can be achieved within VRRP protocol by reducing the Advertisement and Master Down Interval. However, sub second Advert timers, can put extra load on CPU and the network bandwidth which may not be desirable.

Since the VRRP protocol depends on the availability of Layer 3 IPv4 or IPv6 connectivity between redundant peers, the VRRP protocol can interact with the Layer 3 variant of BFD as described in [RFC5881] to achieve a much faster failure detection of the VRRP Master on the LAN. BFD, as specified by the [RFC5880] can provide a much faster failure detection in the range of 150ms, if implemented in the part of a Network device which scales better than VRRP when sub second Advert timers are used.

2. Requirements Language

In this document, several words are used to signify the requirements of the specification. 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. Applicability of Point to Point BFD

BFD for IPv4 or IPv6 (Single Hop) [RFC5881] requires that in order for a BFD session to be formed both peers participating in a BFD session need to know its peer IPv4 or IPV6 address. This poses a unique problem with the definition of the VRRP protocol, that makes the use of BFD for IPv4 or IPv6 [RFC5881] more challenging. In VRRP it is only the Master router that sends Advert packets. This means that a Master router is not aware of any Backup routers, and Backup routers are only aware of the Master router. This also means that a Backup router is not aware of any other Backup routers in the Network.

Since BFD for IPv4 or IPv6 [RFC5881] requires that a session be formed by both peers using a full destination and source address, there needs to be some external means to provide this information to BFD on behalf of VRRP. Once the peer information is made available, VRRP can form BFD sessions with its peer Virtual Router. The BFD session for a given Virtual Router is identified as the Critical Path BFD Session, which is the session that forms between the current VRRP Master router, and the highest priority Backup router. When the Critical Path BFD Session identified by VRRP as having changed state from Up to Down, then this will be interpreted by the VRRP state machine on the highest priority Backup router as a Master Down event. A Master Down event means that the highest priority Backup peer will immediately become the new Master for the Virtual Router.

NOTE: At all times, the normal fail-over mechanism defined in the VRRP [RFC5798] will be unaffected, and the BFD fail-over mechanism will always resort to normal VRRP fail-over.

This draft defines the mechanism used by the VRRP protocol to build a peer table that will help in forming of BFD session and the detection of Critical Path BFD session. If the Critical Path BFD session were to go down, it will signal a Master Down event and make the most preferred Backup router as the VRRP Master router. This requires an extension to the VRRP protocol.

This can be achieved by defining a new type in the VRRP Advert packet, and allowing VRRP peers to build a peer table in any of the operational state, Master or Backup.

3.1. Extension to VRRP protocol

In this mode of operation VRRP peers learn the adjacent routers, and form BFD session between the learnt routers. In order to build the peer table, all routers send VRRP Advert packets whilst in any of the operational states (Master or Backup). Normally VRRP peers only send

Advert packets whilst in the Master state, however in this mode VRRP Backup peers will also send Advert packets with the type field set to BACKUP ADVERTISEMENT type defined in Section 3.3 of this document. The VRRP Master router will still continue to send packets with the Advert type as ADVERTISEMENT as defined in the VRRP protocol. This is to maintain inter-operability with peers complying to VRRP protocol.

Additionally, Advert packets sent from Backup Peers must not use the Virtual router MAC address as the source address. Instead it must use the Interface MAC address as the source address from which the packet is sent from. This is because the source MAC override feature is used by the Master to send Advert packets from the Virtual Router MAC address, which is used to keep the bridging cache on LAN switches and bridging devices refreshed with the destination port for the Virtual Router MAC.

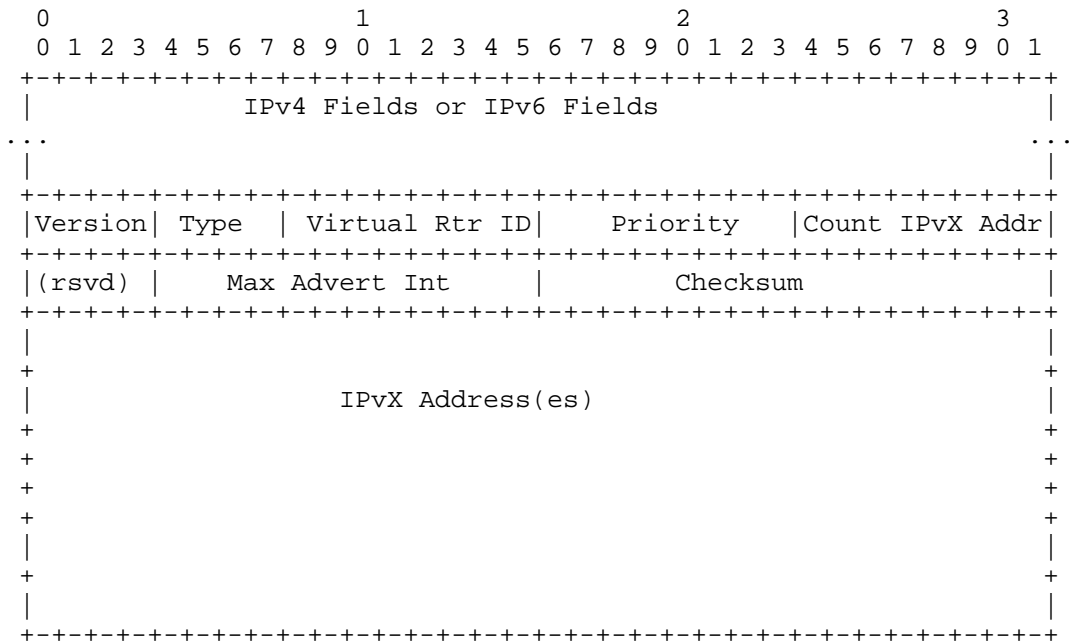
3.2. VRRP Peer Table

VRRP peers can now form the peer table by learning the source address in the ADVERTISEMENT or BACKUP ADVERTISEMENT packet sent by VRRP Master or Backup peers. This allows peers to create BFD sessions with other operational peers.

A peer entry should be removed from the peer table if Advert is not received from a peer for a period of (3 * the Advert interval).

3.3. VRRP BACKUP ADVERTISEMENT Packet Type

The following figure shows the VRRP packet as defined in VRRP [RFC5798] RFC.



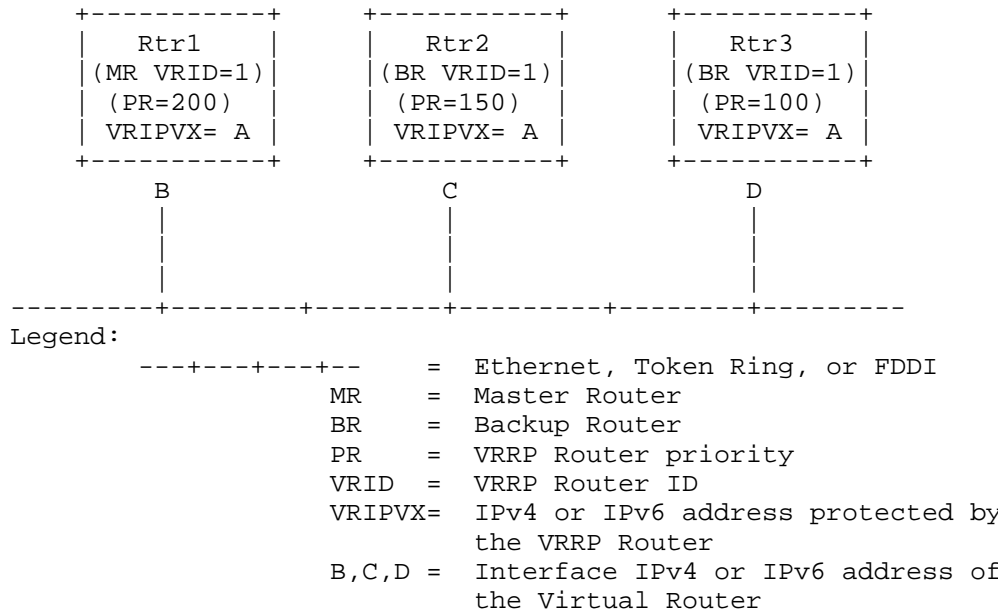
The type field specifies the type of this VRRP packet. The type field can have two values. Type 1 (ADVERTISEMENT) is used by the VRRP Master Router. Type 2 (BACKUP ADVERTISEMENT) is used by the VRRP Backup router. This is to distinguish the packets sent by the VRRP backup Router. VRRP Backup fills Backup_Advertisement_Interval in the Max Advert Int of BACKUP ADVERTISEMENT packet. Rest of the fields in Advert packet remain the same.

- 1 ADVERTISEMENT
- 2 BACKUP ADVERTISEMENT

A packet with unknown type MUST be discarded.

3.4. Sample configuration

The following figure shows a simple network with three VRRP routers implementing one virtual router.



In the above configuration there are three routers on the LAN protecting an IPv4 or IPv6 address associated to a Virtual Router ID 1. Rtr1 is the Master router since it has the highest priority compared to Rtr2 and Rtr3. Now if peer learning extension is enabled on all the peers. Rtr1 will send the Advert packet with type field set to 1. While Rtr2 and Rtr3 will send the Advert packet with type field set to 2. In the above configuration the peer table built at each router is shown below:

Rtr1 Peer table

Peer Address	Priority
C	150
D	100

Rtr2 Peer table

Peer Address	Priority
B	200
D	100

Rtr3 Peer table

Peer Address	Priority
B	200
C	150

Once the peer tables are formed, VRRP on each router can form a BFD sessions with the learnt peers.

3.5. Critical BFD session

The Critical BFD Session is determined to be the session between the VRRP Master and the next best VRRP Backup. Failure of the Critical BFD session indicates that the Master is no longer available and the most preferred Backup will now become Master.

In the above example the Critical BFD session is shared between Rtr1 and Rtr2. If the BFD Session goes from Up to Down state, Rtr2 can treat it as a Master down event and immediately assume the role of VRRP Master router for VRID 1 and Rtr3 will become the critical Backup. If the priorities of two Backup routers are same then the primary IPvX Address of the sender is used to determine the highest priority Backup. Where higher IPvX address has higher priority.

3.6. Protocol State Machine

3.6.1. Parameters Per Virtual Router

Following parameters are added to the VRRP protocol to support this mode of operation.

Backup_Advertisement_Interval	Time interval between BACKUP ADVERTISEMENTS (centiseconds). Default is 100 centiseconds (1 second).
Backup_Adver_Interval	Advertisement interval contained in BACKUP ADVERTISEMENTS received from the Backup (centiseconds). This value is saved by virtual routers used, to compute Backup_Down_Interval.
Backup_Down_Interval	Time interval for VRRP instance to declare Backup down (centiseconds). Calculated as $(3 * \text{Backup_Adver_Interval})$ for each VRRP Backup.
Critical_Backup	Procedure outlined in section 3.4 of this document is used to determine the Critical_Backup at each VRRP Instance.
Critical_BFD_Session	The Critical BFD Session is the session between the VRRP Master and Critical_Backup.

3.6.2. Timers

Following timers are added to the VRRP protocol to support this mode of operation.

Backup_Down_Timer	Timer that fires when BACKUP ADVERTISEMENT has not been heard from a backup peer for Backup_Down_Interval.
Backup_Adver_Timer	Timer that fires to trigger sending of BACKUP ADVERTISEMENT based on Backup_Advertisement_Interval.

3.6.3. VRRP State Machine with Point to Point BFD

Following State Machine replaces the state Machine outlined in section 6.4 of the VRRP protocol [RFC5798] to support this mode of operation. Please refer to the section 6.4 of [RFC5798] for State description.

3.6.3.1. Initialize

Following state machine replaces the state machine outlined in section 6.4.1 of [RFC5798]

```
(100) If a Startup event is received, then:

    (105) - If the Priority = 255 (i.e., the router owns the IPvX
    address associated with the virtual router), then:

        (110) + Send an ADVERTISEMENT

        (115) + If the protected IPvX address is an IPv4 address, then:

            (120) * Broadcast a gratuitous ARP request containing the
            virtual router MAC address for each IP address associated
            with the virtual router.

        (125) + else // IPv6

            (130) * For each IPv6 address associated with the virtual
            router, send an unsolicited ND Neighbor Advertisement with
            the Router Flag (R) set, the Solicited Flag (S) unset, the
            Override flag (O) set, the target address set to the IPv6
            address of the virtual router, and the target link-layer
            address set to the virtual router MAC address.

        (135) +endif // was protected addr IPv4?

        (140) + Set the Adver_Timer to Advertisement_Interval

        (145) + Transition to the {Master} state

    (150) - else // rtr does not own virt addr

        (155) + Set Master_Adver_Interval to Advertisement_Interval

        (160) + Set the Master_Down_Timer to Master_Down_Interval

        (165) + Set Backup_Adver_Timer to Backup_Advertisement_Interval

        (170) + Transition to the {Backup} state

    (175) -endif // priority was not 255

(180) endif // startup event was recvd
```

3.6.3.2. Backup

Following state machine replaces the state machine outlined in section 6.4.2 of [RFC5798]

```
(300) While in this state, a VRRP router MUST do the following:

(305) - If the protected IPvX address is an IPv4 address, then:

    (310) + MUST NOT respond to ARP requests for the IPv4
    address(es) associated with the virtual router.

(315) - else // protected addr is IPv6

    (320) + MUST NOT respond to ND Neighbor Solicitation messages
    for the IPv6 address(es) associated with the virtual router.

    (325) + MUST NOT send ND Router Advertisement messages for the
    virtual router.

(330) -endif // was protected addr IPv4?

(335) - MUST discard packets with a destination link-layer MAC
address equal to the virtual router MAC address.

(340) - MUST NOT accept packets addressed to the
IPvX address(es) associated with the virtual router.

(345) - If a Shutdown event is received, then:

    (350) + Cancel the Master_Down_Timer.

    (355) + Cancel the Backup_Adver_Timer.

    (360) + Cancel Backup_Down_Timers.

    (365) + Remove Peer table.

    (370) + If Critical_BFD_Session Exists:

        (375) * Tear down the Critical_BFD_Session.

    (380) + endif // Critical_BFD_Session Exists?

    (385) + Send a BACKUP ADVERTISEMENT with Priority = 0.

    (390) + Transition to the {Initialize} state.
```

```
(395) -endif // shutdown recv

(400) - If the Master_Down_Timer fires or
      If Critical_BFD_Session transitions from UP to DOWN, then:

(405) + Send an ADVERTISEMENT

(415) + If the protected IPvX address is an IPv4 address, then:

      (420) * Broadcast a gratuitous ARP request on that interface
            containing the virtual router MAC address for each IPv4
            address associated with the virtual router.

(425) + else // ipv6

      (430) * Compute and join the Solicited-Node multicast
            address [RFC4291] for the IPv6 address(es) associated with
            the virtual router.

      (435) * For each IPv6 address associated with the virtual
            router, send an unsolicited ND Neighbor Advertisement with
            the Router Flag (R) set, the Solicited Flag (S) unset, the
            Override flag (O) set, the target address set to the IPv6
            address of the virtual router, and the target link-layer
            address set to the virtual router MAC address.

(440) +endif // was protected addr ipv4?

(445) + Set the Adver_Timer to Advertisement_Interval.

(450) + If the Critical_BFD_Session exists:

      (455) @ Tear Critical_BFD_Session.

(460) + endif // Critical_BFD_Session exists

(465) + Calculate the Critical_Backup.

(470) + If the Critical_Backup exists:

      (475) * Bootstrap Critical_BFD_Session with the
            Critical_Backup.

(480) + endif //Critical_Backup exists?

(485) + Transition to the {Master} state.

(490) -endif // Master_Down_Timer fired
```

```
(485) - If an ADVERTISEMENT is received, then:

    (490) + If the Priority in the ADVERTISEMENT is zero, then:

        (495) * Set the Master_Down_Timer to Skew_Time.

        (500) * If the Critical_BFD_Session exists:

            (505) * Tear Critical_BFD_Session with the Master.

        (510) * endif // Critical_BFD_Session exists

    (515) + else // priority non-zero

        (520) * If Preempt_Mode is False, or if the Priority in the
        ADVERTISEMENT is greater than or equal to the local
        Priority, then:

            (525) @ Set Master_Adver_Interval to Adver Interval
            contained in the ADVERTISEMENT.

            (530) @ Recompute the Master_Down_Interval.

            (535) @ Reset the Master_Down_Timer to
            Master_Down_Interval.

            (540) @ Determine Critical_Backup.

            (545) @ If Critical_BFD_Session does not exists and this
            instance is the Critical_Backup:

                (550) @+ Bootstrap Critical_BFD_Session with Master.

            (555) @ endif //Critical_BFD_Session exists check

        (560) * else // preempt was true or priority was less

            (565) @ Discard the ADVERTISEMENT.

        (570) *endif // preempt test

    (575) +endif // was priority zero?

(580) -endif // was advertisement rcv?

(585) - If a BACKUP ADVERTISEMENT is received, then:

    (590) + If the Priority in the BACKUP ADVERTISEMENT is zero,
```

```
        then:

(595) * Cancel Backup_Down_Timer.

(600) * Remove the Peer from Peer table.

(605) + else // priority non-zero

(610) * Update the peer table with peer information.

(615) * Set Backup_Adver_Interval to Adver Interval
contained in the BACKUP ADVERTISEMENT.

(620) * Recompute the Backup_Down_Interval.

(625) * Reset the Backup_Down_Timer to Backup_Down_Interval.

(630) +endif // was priority zero?

(635) + Recalculate Critical_Backup.

(640) + If Critical_BFD_Session exists and this
instance is not the Critical_Backup:

(645) * Tear Down the Critical_BFD_Session.

(650) + else If Critical_BFD_Session does not exists and this
instance is the Critical_Backup:

(655) * BootStrap Critical_BFD_Session with Master.

(660) + endif // Critical_Backup change

(665) -endif // was backup advertisement recv?

(670) - If Backup_Down_Timer fires, then:

(675) + Remove the Peer from Peer table.

(680) + If Critical_BFD_Session does not exist:

(685) @ Recalculate Critical_Backup.

(690) @ If This instance is the Critical_Backup:

(695) +@ BootStrap Critical_BFD_Session with Master.

(700) @ endif // Critical_Backup change
```

```
(705) + endif // Critical_BFD_Session does not exist?
(710) -endif // Backup_Down_Timer fires?
(715) - If Backup_Adver_Timer fires, then:
    (720) + Send a BACKUP ADVERTISEMENT.
    (725) + Reset the Backup_Adver_Timer to
            Backup_Advertisement_Interval.
(730) -endif // Backup_Down_Timer fires?
(735) endwhile // Backup state
```

3.6.3.3. Master

Following state machine replaces the state machine outlined in section 6.4.3 of [RFC5798]

```
(800) While in this state, a VRRP router MUST do the following:
    (805) - If the protected IPvX address is an IPv4 address, then:
        (810) + MUST respond to ARP requests for the IPv4 address(es)
                associated with the virtual router.
    (815) - else // ipv6
        (820) + MUST be a member of the Solicited-Node multicast
                address for the IPv6 address(es) associated with the virtual
                router.
        (825) + MUST respond to ND Neighbor Solicitation message for
                the IPv6 address(es) associated with the virtual router.
        (830) + MUST send ND Router Advertisements for the virtual
                router.
        (835) + If Accept_Mode is False: MUST NOT drop IPv6
                Neighbor Solicitations and Neighbor Advertisements.
    (840) -endif // ipv4?
    (845) - MUST forward packets with a destination link-layer MAC
            address equal to the virtual router MAC address.
```

(850) - MUST accept packets addressed to the IPvX address(es) associated with the virtual router if it is the IPvX address owner or if Accept_Mode is True. Otherwise, MUST NOT accept these packets.

(855) - If a Shutdown event is received, then:

(860) + Cancel the Adver_Timer.

(865) + Send an ADVERTISEMENT with Priority = 0,

(870) + Cancel Backup_Down_Timers.

(875) + Remove Peer table.

(880) + If Critical_BFD_Session Exists:

(885) * Tear down Critical_BFD_Session

(890) + endif // If Critical_BFD_Session Exists

(895) + Transition to the {Initialize} state.

(900) -endif // shutdown recv

(905) - If the Adver_Timer fires, then:

(910) + Send an ADVERTISEMENT.

(915) + Reset the Adver_Timer to Advertisement_Interval.

(920) -endif // advertisement timer fired

(925) - If an ADVERTISEMENT is received, then:

(930) -+ If the Priority in the ADVERTISEMENT is zero, then:

(935) -* Send an ADVERTISEMENT.

(940) -* Reset the Adver_Timer to Advertisement_Interval.

(945) -+ else // priority was non-zero

(950) -* If the Priority in the ADVERTISEMENT is greater than the local Priority,

(955) -* or


```
(960) -* If the Priority in the ADVERTISEMENT is equal to
the local Priority and the primary IPvX Address of the
sender is greater than the local primary IPvX Address, then:

(965) -@ Cancel Adver_Timer

(970) -@ Set Master_Adver_Interval to Adver Interval
contained in the ADVERTISEMENT

(975) -@ Recompute the Skew_Time

(980) @ Recompute the Master_Down_Interval

(985) @ Set Master_Down_Timer to Master_Down_Interval

(990) If Critical_BFD_Session Exists:

    (995) @+ Tear Critical_BFD_Session

(960) @ endif //Critical_BFD_Session Exists?

(965) @ Calculate Critical_Backup.

(970) @ If this instance is Critical_Backup:

    (975) @+ Bootstrap Critical_BFD_Session with new
        Master.

(980) @ endif // am i Critical_Backup?

(985) @ Transition to the {Backup} state

(990) * else // new Master logic

    (995) @ Discard ADVERTISEMENT

(1000) *endif // new Master detected

(1005) +endif // was priority zero?

(1010) -endif // advert recv

(1015) - If a BACKUP ADVERTISEMENT is received, then:

    (1020) + If the Priority in the BACKUP ADVERTISEMENT is
        zero, then:

        (1025) * Remove the Peer from peer table.
```

```
(1030) + else: // priority non-zero
    (1035) * Update the Peer info in peer table.
    (1040) * Recompute the Backup_Down_Interval
    (1045) * Reset the Backup_Down_Timer to
            Backup_Down_Interval
(1050) + endif // priority in backup advert zero
(1055) + Calculate the Critical_Backup
(1060) + If Critical_BFD_Session doesnot exist:
    (1065) * Bootstrap Critical_BFD_Session
(1070) + else if Critical_BFD_Session exist and
            Critical_Backup changes:
    (1075) + Tear Critical_BFD_Session with old Backup
    (1080) + Bootstrap Critical_BFD_Session with Critical_Backup
(1085) + endif // Critical_BFD_Session check?
(1090) - endif // backup advert recv
(1095) - If Critical_BFD_Session transitions from UP to DOWN,
then:
    (1100) + Cancel Backup_Down_Timer
    (1105) + Delete the Peer info from peer table
    (1200) + Calculate the Critical_Backup
    (1205) + Bootstrap Critical_BFD_Session with Critical_Backup
(1210) - endif // BFD session transition
(1215) endwhile // in Master
```

4. Scalability Considerations

To reduce the number of packets generated at a regular interval, Backup Advert packets may be sent at a reduced rate as compared to Advert packets sent by the VRRP Master.

5. Operational Considerations

A VRRP peer that forms a member of this Virtual Router, but does not support this feature or extension must be configured with the lowest priority, and will only operate as the Router of last resort on failure of all other VRRP routers supporting this functionality.

It is recommended that mechanism defined by this draft, to interface VRRP with BFD should be used when BFD can support more aggressive monitoring timers than VRRP. Otherwise it is desirable not to interface VRRP with BFD for determining the health of VRRP Master.

This Draft does not preclude the possibility of the peer table being populated by means of manual configuration, instead of using the BACKUP ADVERTISEMENT as defined by the Draft.

6. Applicability to VRRPv2

The workings of this Draft can be extended to VRRPv2 [RFC3768], with the introduction of BACKUP ADVERTISEMENT and Peer Table as outlined in the Draft.

7. IANA Considerations

This document requests IANA to create a new name space that is to be managed by IANA. The document defines a new VRRP Packet Type. The VRRP Packet Types are discussed below.

- a) Type 1 (ADVERTISEMENT) defined in section 5.2.2 of [RFC5798]
- b) Type 2 (BACKUP ADVERTISEMENT) defined in section 3.3 of this document

7.1. A New Name Space for VRRP Packet Types

This document defines in Section 3.3 a "BACKUP ADVERTISEMENT" VRRP Packet Type. The new name space has to be created by the IANA and they will maintain this new name space. The field for this namespace is 4-Bits, and IANA guidelines for assignments for this field are as follows:

ADVERTISEMENT	1
BACKUP ADVERTISEMENT	2

Future allocations of values in this name space are to be assigned by IANA using the "Specification Required" policy defined in [IANA-CONS]

8. Security Considerations

Security considerations discussed in [RFC5798], [RFC5880], apply to this document. There are no additional security considerations identified by this draft.

9. Acknowledgements

The authors gratefully acknowledge the contributions of Gerry Meyer, and Mouli Chandramouli, for their contributions to the draft. The authors will also like to thank Jeffrey Haas, Maik Pfeil, Chris Bowers, Vengada Prasad Govindan and Alexander Vainshtein for their comments and suggestions.

10. Normative References

- [RFC5880] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD)", RFC 5880, 2010.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", RFC 2119, 1997.
- [RFC5881] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD) for IPv4 and IPv6 (Single Hop)", RFC 5881, 2010.
- [RFC5798] Nadas, S., "Virtual Router Redundancy Protocol (VRRP) Version 3 for IPv4 and IPv6", RFC 5798, 2010.
- [RFC3768] Hinden, R., "Virtual Router Redundancy Protocol (VRRP)", RFC 3768, 2004.
- [IANA-CONS] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", RFC 2434, 1998.

Authors' Addresses

Nitish Gupta
Cisco Systems, Inc.
3265 CISCO Way
San Jose 95134
United States

Phone: +91 80 4429 2530
Email: nitisgup@cisco.com
URI: <http://www.cisco.com/>

Aditya Dogra
Cisco Systems, Inc.
Sarjapur Outer Ring Road
Bangalore 560103
India

Phone: +91 80 4429 2166
Email: adogra@cisco.com
URI: <http://www.cisco.com/>

Colin Docherty
AT&T
23 The Maltings
Haddington, Scotland EH414EF
United Kingdom

Email: colin.docherty@att.com

Greg Mirsky
Individual

Email: gregimirsky@gmail.com

Jeff Tantsura
Individual

Email: jefftant.ietf@gmail.com

BFD Working Group
Internet-Draft
Intended status: Informational
Expires: 8 September 2022

G. Mirsky
Ericsson
7 March 2022

BFD in Demand Mode over Point-to-Point MPLS LSP
draft-mirsky-bfd-mpls-demand-11

Abstract

This document describes procedures for using Bidirectional Forwarding Detection (BFD) in Demand mode to detect data plane failures in Multiprotocol Label Switching (MPLS) point-to-point Label Switched Paths.

Status of This Memo

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

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

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

This Internet-Draft will expire on 8 September 2022.

Copyright Notice

Copyright (c) 2022 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 (<https://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 Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction	2
2. Conventions used in this document	2
2.1. Terminology	2
3. Use of the BFD Demand Mode	2
3.1. The Applicability of BFD for Multipoint Networks	4
4. IANA Considerations	4
5. Security Considerations	4
6. Normative References	4
7. Informative References	5
Appendix A. Acknowledgements	5
Author's Address	6

1. Introduction

[RFC5884] defined use of the Asynchronous method of Bidirectional Detection (BFD) [RFC5880] to monitor and detect failures in the data path of a Multiprotocol Label Switching (MPLS) Label Switched Path (LSP). Use of the Demand mode, also specified in [RFC5880], has not been defined so far. This document describes procedures for using the Demand mode of BFD protocol to detect data plane failures in MPLS point-to-point (p2p) LSPs.

2. Conventions used in this document

2.1. Terminology

MPLS: Multiprotocol Label Switching

LSP: Label Switched Path

LER: Label switching Edge Router

BFD: Bidirectional Forwarding Detection

p2p: Point-to-Point

3. Use of the BFD Demand Mode

[RFC5880] defines that the Demand mode may be:

- * asymmetric, i.e. used in one direction of a BFD session;
- * switched to and from without bringing BFD session to Down state through using a Poll Sequence.

For the case of BFD over MPLS LSP, ingress Label switching Edge Router (LER) usually acts as Active BFD peer and egress LER acts as Passive BFD peer. The Active peer bootstraps the BFD session by using LSP ping. If the BFD session is configured to use the Demand mode, once the BFD session is in Up state the ingress LER switches to the Demand mode as defined in Section 6.6 [RFC5880]. The egress LER also follows procedures defined in Section 6.6 [RFC5880] and ceases further transmission of periodic BFD control packets to the ingress LER.

In this state BFD peers remains as long as the egress LER is in Up state. The ingress LER can periodically check continuity of a bidirectional path between the ingress and egress LERs by using the Poll Sequence, as described in Section 6.6 [RFC5880]. An implementation that supports using the Poll Sequence as the mechanism for bidirectional path continuity check must control the interval between consecutive Poll Sequences. The Rdefault value could be selected as 1 second.

If the Detection timer at the egress LER expires, the BFD system on LER sends BFD Control packet to the ingress LER with the Poll (P) bit set, Status (Sta) field set to the Down value, and the Diagnostic (Diag) field set to Control Detection Time Expired value. The egress LER periodically transmits these Control packets to the ingress LER until either it receives the valid for this BFD session control packet with the Final (F) bit set from the ingress LER or the defect condition clears and the BFD session state reaches Up state at the egress LER. An implementation that supports this specification provides control of the interval between consecutive Poll messages signaling the expiration of the Detection timer. The default value of the interval can be selected as 1 second.

The ingress LER transmits BFD Control packets over the MPLS LSP with the Demand (D) flag set at negotiated interval per [RFC5880], the greater of `bfd.DesiredMinTxInterval` and `bfd.RemoteMinRxInterval`, until it receives the valid BFD packet from the egress LER with the Poll (P) bit and the Diagnostic (Diag) field value Control Detection Time Expired. Reception of such BFD control packet by the ingress LER indicates that the monitored LSP has a failure and sending BFD control packet with the Final flag set to acknowledge failure indication is likely to fail. Instead, the ingress LER transmits the BFD Control packet to the egress LER over the IP network with:

- * destination IP address is set to the destination IP address of the LSP Ping Echo request message [RFC8029];
- * destination UDP port set to 4784 [RFC5883];

- * Final (F) flag in BFD control packet is set;
- * Demand (D) flag in BFD control packet is cleared.

The ingress LER changes the state of the BFD session to Down and changes rate of BFD Control packets transmission to one packet per second. The ingress LER in Down mode changes to Asynchronous mode until the BFD session comes to Up state once again. Then the ingress LER switches to the Demand mode.

3.1. The Applicability of BFD for Multipoint Networks

[RFC8562] defines the use of BFD in multipoint networks. This specification analyzes the case of p2p LSP. In that scenario, the ingress of the LSP acts as the MultipointHead, and the egress - as MultipointTail. The BFD state machines for MultipointHead, MultipointClient, and MultipointTail don't use the three-way handshakes for session establishment and teardown. As a result, the Init state is absent, and the session transitions to the Up state once the BFD session is administratively enabled. Hence, a BFD session over a p2p LSP, using principles of [RFC8562] or [RFC8563], can be established faster if the MultipointTail has been provisioned with the value of My Discriminator used by the MultipointHead for that BFD session. That value can be provided to the MultipointTail using different mechanisms, e.g., an extension to IGP. Description of mechanism to provide the value of My Discriminator used by the MultipointHead for the particular BFD session is outside the scope of this specification.

Unsolicited notification of the detected failure by the MultipointTail to the MultipointClient performs as described above for the case when the ingress BFD system switches the remote peer into the Demand mode.

4. IANA Considerations

TBD

5. Security Considerations

This document does not introduce new security aspects but inherits all security considerations from [RFC5880], [RFC5884], [RFC7726], [RFC8029], and [RFC6425].

6. Normative References

- [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>>.
- [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>>.
- [RFC6425] Saxena, S., Ed., Swallow, G., Ali, Z., Farrel, A., Yasukawa, S., and T. Nadeau, "Detecting Data-Plane Failures in Point-to-Multipoint MPLS - Extensions to LSP Ping", RFC 6425, DOI 10.17487/RFC6425, November 2011, <<https://www.rfc-editor.org/info/rfc6425>>.
- [RFC7726] Govindan, V., Rajaraman, K., Mirsky, G., Akiya, N., and S. Aldrin, "Clarifying Procedures for Establishing BFD Sessions for MPLS Label Switched Paths (LSPs)", RFC 7726, DOI 10.17487/RFC7726, January 2016, <<https://www.rfc-editor.org/info/rfc7726>>.
- [RFC8029] Kompella, K., Swallow, G., Pignataro, C., Ed., Kumar, N., Aldrin, S., and M. Chen, "Detecting Multiprotocol Label Switched (MPLS) Data-Plane Failures", RFC 8029, DOI 10.17487/RFC8029, March 2017, <<https://www.rfc-editor.org/info/rfc8029>>.
- [RFC8562] Katz, D., Ward, D., Pallagatti, S., Ed., and G. Mirsky, Ed., "Bidirectional Forwarding Detection (BFD) for Multipoint Networks", RFC 8562, DOI 10.17487/RFC8562, April 2019, <<https://www.rfc-editor.org/info/rfc8562>>.

7. Informative References

- [RFC8563] Katz, D., Ward, D., Pallagatti, S., Ed., and G. Mirsky, Ed., "Bidirectional Forwarding Detection (BFD) Multipoint Active Tails", RFC 8563, DOI 10.17487/RFC8563, April 2019, <<https://www.rfc-editor.org/info/rfc8563>>.

Appendix A. Acknowledgements

TBD

Author's Address

Greg Mirsky
Ericsson
Email: gregimirsky@gmail.com

BFD Working Group
Internet-Draft
Updates: 5798 (if approved)
Intended status: Standards Track
Expires: 27 September 2021

G. Mirsky
ZTE Corp.
J. Tantsura
Juniper Networks
G. Mishra
Verizon Inc.
26 March 2021

Bidirectional Forwarding Detection (BFD) for Multi-point Networks and
Virtual Router Redundancy Protocol (VRRP) Use Case
draft-mirsky-bfd-p2mp-vrrp-use-case-06

Abstract

This document discusses the use of Bidirectional Forwarding Detection (BFD) for multipoint networks to provide Virtual Router Redundancy Protocol (VRRP) with sub-second Master convergence and defines the extension to bootstrap point-to-multipoint BFD session.

This draft updates RFC 5798.

Status of This Memo

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

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

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

This Internet-Draft will expire on 27 September 2021.

Copyright Notice

Copyright (c) 2021 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 (<https://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	2
1.1. Conventions used in this document	3
1.1.1. Terminology	3
1.1.2. Requirements Language	3
2. Problem Statement	3
3. Applicability of p2mp BFD	3
3.1. Multipoint BFD Encapsulation	5
4. IANA Considerations	5
5. Security Considerations	5
6. Acknowledgements	5
7. Normative References	5
Authors' Addresses	6

1. Introduction

The [RFC5798] is the current specification of the Virtual Router Redundancy Protocol (VRRP) for IPv4 and IPv6 networks. VRRPv3 allows for a faster switchover to a Backup router. Using such capability with the software-based implementation of VRRP may prove challenging. But it still may be possible to deploy VRRP and provide sub-second detection of Master router failure by Backup routers.

Bidirectional Forwarding Detection (BFD) [RFC5880] had been originally defined detect failure of point-to-point (p2p) paths: single-hop [RFC5881], multihop [RFC5883]. Single-hop BFD may be used to enable Backup routers to detect a failure of the Master router within 100 msec or faster.

[RFC8562] extends [RFC5880] for multipoint and multicast networks, which is precisely characterizes deployment scenarios for VRRP over LAN segment. This document demonstrates how point-to-multipoint (p2mp) BFD can enable faster detection of Master failure and thus minimize service disruption in a VRRP domain. The document also defines the extension to VRRP [RFC5798] to bootstrap a VRRP Backup router to join in p2mp BFD session.

1.1. Conventions used in this document

1.1.1. Terminology

BFD: Bidirectional Forwarding Detection

p2mp: Pont-to-Multipoint

VRRP: Virtual Router Redundancy Protocol

1.1.2. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

2. Problem Statement

A router may be part of several Virtual Router Redundancy groups, as Master in some and as Backup in others. Supporting sub-second mode for VRRPv3 [RFC5798] for all these roles without specialized support in data plane may prove challenging. BFD already has many implementations based on HW that are capable of supporting multiple sub-second sessions concurrently.

3. Applicability of p2mp BFD

[RFC8562] may provide an efficient and scalable solution for fast-converging environment that uses the default route rather than dynamic routing. Each redundancy group presents itself as a p2mp BFD session with its Master being the root and Backup routers being tails of the p2mp BFD session. Figure 1 displays the extension of VRRP [RFC5798] to bootstrap tail of the p2mp BFD session. Master

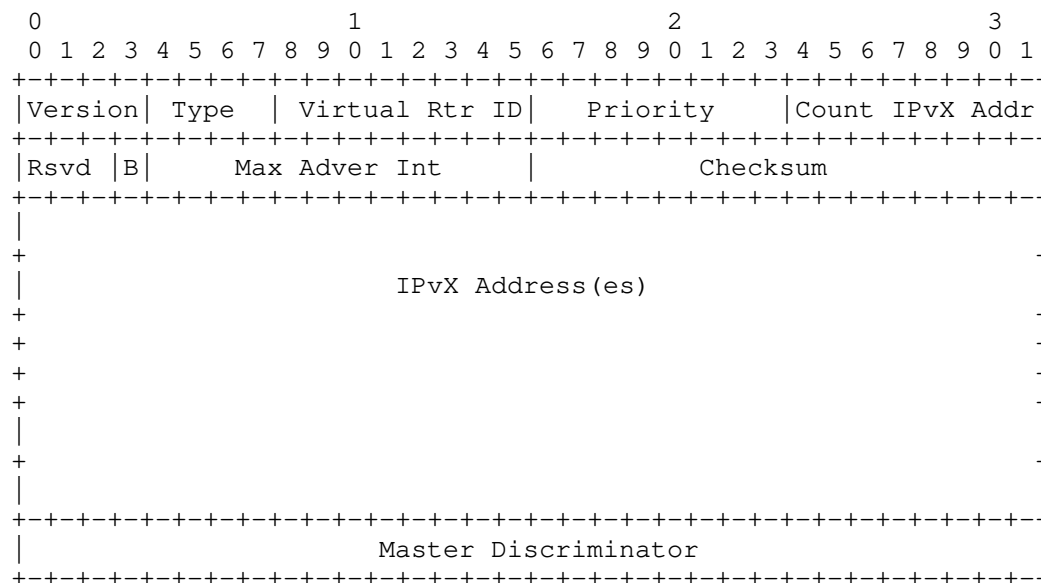


Figure 1: VRRP Extension to Bootstrap P2MP BFD session

where new fields are interpreted as:

B(FD) - a one-bit flag that indicates that the Master Discriminator field is appended to VRRP packet defined in [RFC5798];

Master Discriminator - My Discriminator value allocated by the root of the p2mp BFD session.

The Master router that is configured to use p2mp BFD to support faster convergence of VRRP starts transmitting BFD control packets with VRID as a source IP address and My Discriminator. The same value of My Discriminator MUST be set as the value of the Master Discriminator field. The BFD flag MUST be set in the VRRP packet. Backup router demultiplexes p2mp BFD test sessions based on VRID that it been configured with and the My Discriminator value it learns from the received VRRP packet. When a Backup router detects the failure of the Master router it re-evaluates its role in the VRID. As a result, the Backup router may become the Master router of the given VRID or continue as a Backup router. If the former is the case, then the new Master router MUST select My Discriminator and start transmitting p2mp BFD control packets using Master IP address as the source IP address for p2mp BFD control packets. If the latter is the case, then the Backup router MUST wait for the VRRP packet from the new VRRP Master router that will bootstrap the new p2mp BFD session.

3.1. Multipoint BFD Encapsulation

The MultipointHead of p2mp BFD session when transmitting BFD control packet:

MUST set TTL value to 1 (though note that VRRP packets have TTL set to 255);

SHOULD use group address VRRP ('224.0.0.18' for IPv4 and 'FF02:0:0:0:0:0:0:12' for IPv6) as destination IP address

MAY use network broadcast address for IPv4 or link-local all nodes multicast group for IPv6 as destination IP address;

MUST set destination UDP port value to 3784 when transmitting BFD control packets, as defined in [RFC8562];

MUST use the Master IP address as the source IP address.

4. IANA Considerations

This document makes no requests for IANA allocations. This section may be deleted by RFC Editor.

5. Security Considerations

Security considerations discussed in [RFC5798], [RFC5880], and [RFC8562], apply to this document.

6. Acknowledgements

7. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC5798] Nadas, S., Ed., "Virtual Router Redundancy Protocol (VRRP) Version 3 for IPv4 and IPv6", RFC 5798, DOI 10.17487/RFC5798, March 2010, <<https://www.rfc-editor.org/info/rfc5798>>.
- [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>>.
- [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>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [RFC8562] Katz, D., Ward, D., Pallagatti, S., Ed., and G. Mirsky, Ed., "Bidirectional Forwarding Detection (BFD) for Multipoint Networks", RFC 8562, DOI 10.17487/RFC8562, April 2019, <<https://www.rfc-editor.org/info/rfc8562>>.

Authors' Addresses

Greg Mirsky
ZTE Corp.

Email: gregimirsky@gmail.com, gregory.mirsky@ztetx.com

Jeff Tantsura
Juniper Networks

Email: jefftant.ietf@gmail.com

Gyan Mishra
Verizon Inc.

Email: gyan.s.mishra@verizon.com

MPLS Working Group
Internet-Draft
Updates: 5884 (if approved)
Intended status: Standards Track
Expires: 1 October 2021

G. Mirsky
Y. Zhao
ZTE Corporation
G. Mishra
Verizon Inc.
30 March 2021

Clarifying Use of LSP Ping to Bootstrap BFD over MPLS LSP
draft-mirsky-mpls-bfd-bootstrap-clarify-02

Abstract

This document, if approved, updates RFC 5884 by clarifying procedures for using MPLS LSP ping to bootstrap Bidirectional Forwarding Detection (BFD) over MPLS Label Switch Path.

Status of This Memo

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

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

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

This Internet-Draft will expire on 1 October 2021.

Copyright Notice

Copyright (c) 2021 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 (<https://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	2
2. Requirements Language	2
3. Use of Return Mode Field	2
4. Use of BFD Discriminator TLV in LSP Echo Reply	3
5. IANA Considerations	3
6. Security Considerations	3
7. Acknowledgements	3
8. Normative References	3
Authors' Addresses	4

1. Introduction

[RFC5884] defines how LSP Ping [RFC8029] uses BFD Discriminator TLV to bootstrap Bidirectional Forwarding Detection (BFD) session over MPLS Label Switch Path (LSP). Implementation and operational experiences suggest that two aspects of using LSP ping to bootstrap BFD session can benefit from clarification. This document updates [RFC5884] in use of Return mode field in MPLS LSP echo request message and use of BFD Discriminator TLV in MPLS LSP echo reply.

2. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

3. Use of Return Mode Field

[RFC5884] does not define the value to be used for the Return mode field [RFC8029] when LSP ping is used to bootstrap a BFD session of MPLS LSP. When LSP echo request is being used to detect defects in MPLS data plane and verify consistency between the control plane and the data plane echo reply is needed to confirm the correct state, provide positive acknowledgment. But when an LSP echo request is being used to bootstrap BFD session, then the positive acknowledgment, according to [RFC5884] is provided by the egress transmitting BFD control message. Thus LSP echo reply is not required to bootstrap BFD session and hence the Return mode field in echo request message SHOULD be set to 1 (Do not reply) [RFC8029] when LSP echo request used to bootstrap BFD session.

4. Use of BFD Discriminator TLV in LSP Echo Reply

[RFC5884] in section 6 defines that echo reply by the egress LSR to BFD bootstrapping echo request MAY include BFD Discriminator TLV with locally assigned discriminator value for the BFD session. But the [RFC5884] does not define how the ingress LSR may use the returned value. From a practical point, as discussed in Section 3, the returned value is not useful since the egress is required to send the BFD control message right after successfully validating the FEC and before sending an echo reply message. Secondly, identifying the corresponding BFD session at ingress without returning its discriminator presents an unnecessary challenge for the implementation. Thus the egress LSR SHOULD NOT include BFD Discriminator TLV if sending echo reply to BFD bootstrapping echo request.

5. IANA Considerations

This document does not require any action by IANA. This section may be removed.

6. Security Considerations

This document does not introduce new security aspects but inherits all security considerations from [RFC5880], [RFC5884], [RFC8029].

7. Acknowledgements

TBA

8. Normative References

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

[RFC8029] Kompella, K., Swallow, G., Pignataro, C., Ed., Kumar, N., Aldrin, S., and M. Chen, "Detecting Multiprotocol Label Switched (MPLS) Data-Plane Failures", RFC 8029, DOI 10.17487/RFC8029, March 2017, <<https://www.rfc-editor.org/info/rfc8029>>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.

Authors' Addresses

Greg Mirsky
ZTE Corporation

Email: gregimirsky@gmail.com, gregory.mirsky@ztetx.com

Yanhua Zhao
ZTE Corporation

Email: zhao.yanhua3@zte.com.cn

Gyan Mishra
Verizon Inc.

Email: gyan.s.mishra@verizon.com

MPLS Working Group
Internet-Draft
Intended status: Standards Track
Expires: 23 March 2022

G. Mirsky
Ericsson
G. Mishra
Verizon Inc.
D. Eastlake
Futurewei Technologies
19 September 2021

BFD for Multipoint Networks over Point-to-Multi-Point MPLS LSP
draft-mirsky-mpls-p2mp-bfd-15

Abstract

This document describes procedures for using Bidirectional Forwarding Detection (BFD) for multipoint networks to detect data plane failures in Multiprotocol Label Switching (MPLS) point-to-multipoint (p2mp) Label Switched Paths (LSPs) and Segment Routing (SR) point-to-multipoint policies with SR-MPLS data plane.

It also describes the applicability of LSP Ping, as in-band, and the control plane, as out-band, solutions to bootstrap a BFD session.

It also describes the behavior of the active tail for head notification.

Status of This Memo

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

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

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

This Internet-Draft will expire on 23 March 2022.

Copyright Notice

Copyright (c) 2021 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 (<https://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	2
2. Conventions used in this document	3
2.1. Terminology	3
2.2. Requirements Language	3
3. Multipoint BFD Encapsulation	3
3.1. IP Encapsulation of Multipoint BFD	4
3.2. Non-IP Encapsulation of Multipoint BFD	4
4. Bootstrapping Multipoint BFD	5
4.1. LSP Ping	5
4.2. Control Plane	6
5. Operation of Multipoint BFD with Active Tail over P2MP MPLS LSP	6
6. Security Considerations	7
7. IANA Considerations	8
8. Acknowledgements	8
9. References	8
9.1. Normative References	8
9.2. Informative References	10
Authors' Addresses	10

1. Introduction

[RFC8562] defines a method of using Bidirectional Detection (BFD) [RFC5880] to monitor and detect unicast failures between the sender (head) and one or more receivers (tails) in multipoint or multicast networks.

[RFC8562] added two BFD session types - MultipointHead and MultipointTail. Throughout this document, MultipointHead and MultipointTail refer to the value of the `bfd.SessionType` is set on a BFD endpoint.

This document describes procedures for using such modes of BFD protocol to detect data plane failures in Multiprotocol Label Switching (MPLS) point-to-multipoint (p2mp) Label Switched Paths (LSPs) and Segment Routing (SR) point-to-multipoint policies with SR-MPLS data plane

The document also describes the applicability of out-band solutions to bootstrap a BFD session in this environment.

It also describes the behavior of the active tail for head notification.

2. Conventions used in this document

2.1. Terminology

MPLS: Multiprotocol Label Switching

LSP: Label Switched Path

BFD: Bidirectional Forwarding Detection

p2mp: Point-to-Multipoint

FEC: Forwarding Equivalence Class

G-ACh: Generic Associated Channel

ACH: Associated Channel Header

GAL: G-ACh Label

LSR: Label Switching Router

SR: Segment Routing

SR-MPLS: SR with MPLS data plane

2.2. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

3. Multipoint BFD Encapsulation

[RFC8562] uses BFD in the Demand mode from the very start of a point-to-multipoint (p2mp) BFD session. Because the head doesn't receive any BFD Control packet from a tail, the head of the p2mp BFD session transmits all BFD Control packets with the value of Your Discriminator field set to zero. As a result, a tail cannot demultiplex BFD sessions using Your Discriminator, as defined in

[RFC5880]. [RFC8562] requires that to demultiplex BFD sessions, the tail uses the source IP address, My Discriminator, and the identity of the multipoint tree from which the BFD Control packet was received. If the BFD Control packet is encapsulated in IP/UDP, then the source IP address MUST be used to demultiplex the received BFD Control packet as described in Section 3.1. The non-IP encapsulation case is described in Section 3.2.

3.1. IP Encapsulation of Multipoint BFD

[RFC8562] defines IP/UDP encapsulation for multipoint BFD over p2mp MPLS LSP:

- * UDP destination port MUST be set to 3784;
- * destination IP address MUST be set to the loopback address 127.0.0.1/32 for IPv4, or the loopback address ::1/128 for IPv6 [RFC4291]. Note that that is different from how the destination IP address selection is defined in Section 7 [RFC5884]. Firstly, because only one loopback address ::1/128 is defined in IPv6. And also, it is recommended to use the Entropy Label [RFC6790] to discover multiple alternate paths in an MPLS network. Using a single loopback address both for IPv4 and IPv6 encapsulation makes it consistent and more straightforward for an implementation.

The Motivation section [RFC6790] lists several advantages of generating the entropy value by an ingress Label Switching Router (LSR) compared to when a transit LSR infers entropy using the information in the MPLS label stack or payload. Thus this specification further clarifies that:

if multiple alternative paths for the given p2mp LSP Forwarding Equivalence Class (FEC) exist, the MultipointHead SHOULD use Entropy Label [RFC6790] used for LSP Ping [RFC8029] to exercise those particular alternative paths;

or the MultipointHead MAY use the UDP port number as discovered by LSP Ping traceroute [RFC8029] as the source UDP port number to possibly exercise those particular alternate paths.

3.2. Non-IP Encapsulation of Multipoint BFD

In some environments, the overhead of extra IP/UDP encapsulations may be considered burdensome, making the use of more compact G-ACh encapsulation attractive. Also, the validation of the IP/UDP encapsulation of a BFD Control packet in a p2mp BFD session may fail because of a problem related to neither the MPLS label stack nor to BFD. Avoiding unnecessary encapsulation of p2mp BFD over an MPLS LSP

improves the accuracy of the correlation of the detected failure and defect in MPLS LSP. Non-IP encapsulation for multipoint BFD over p2mp MPLS LSP MUST use Generic Associated Channel (G-ACH) Label (GAL) (see [RFC5586]) at the bottom of the label stack followed by an Associated Channel Header (ACH). If a BFD Control packet in PW-ACH encapsulation (without IP/UDP Headers) is to be used in ACH, an implementation would not be able to verify the identity of the MultipointHead and, as a result, will not properly demultiplex BFD packets. Hence, a new channel type value is needed. The Channel Type field in ACH MUST be set to TBA1 value Section 7. To provide the identity of the MultipointHead for the particular multipoint BFD session, a Source Address TLV [RFC7212] MUST immediately follow a BFD Control message.

4. Bootstrapping Multipoint BFD

4.1. LSP Ping

LSP Ping is the part of the on-demand OAM toolset used to detect and localize defects in the data plane and verify the control plane against the data plane by ensuring that the LSP is mapped to the same FEC at both egress and ingress endpoints.

LSP Ping, as defined in [RFC6425], MAY be used to bootstrap MultipointTail. If LSP Ping is used, it MUST include the Target FEC TLV and the BFD Discriminator TLV defined in [RFC5884]. For the case of p2mp MPLS LSP, the Target FEC TLV MUST use sub-TLVs defined in Section 3.1 [RFC6425]. For the case of p2mp SR policy with SR-MPLS data plane, an implementation of this specification MUST follow procedures defined in [RFC8287]. Setting the value of Reply Mode field to "Do not reply" [RFC8029] for the LSP Ping to bootstrap MultipointTail of the p2mp BFD session is RECOMMENDED. Indeed, because BFD over a multipoint network uses BFD Demand mode, the LSP echo reply from a tail has no useful information to convey to the head, unlike in the case of the BFD over a p2p MPLS LSP [RFC5884]. A MultipointTail that receives an LSP Ping that includes the BFD Discriminator TLV:

- * MUST validate the LSP Ping;
- * MUST associate the received BFD Discriminator value with the p2mp LSP;
- * MUST create a p2mp BFD session and set bfd.SessionType = MultipointTail as described in [RFC8562];

- * MUST use the source IP address of LSP Ping, the value of BFD Discriminator from the BFD Discriminator TLV, and the identity of the p2mp LSP to properly demultiplex BFD sessions.

Besides bootstrapping a BFD session over a p2mp LSP, LSP Ping SHOULD be used to verify the control plane against the data plane periodically by checking that the p2mp LSP is mapped to the same FEC at the MultipointHead and all active MultipointTails. The rate of generation of these LSP Ping Echo request messages SHOULD be significantly less than the rate of generation of the BFD Control packets because LSP Ping requires more processing to validate the consistency between the data plane and the control plane. An implementation MAY provide configuration options to control the rate of generation of the periodic LSP Ping Echo request messages.

4.2. Control Plane

The BGP-BFD Attribute [RFC9026] MAY be used to bootstrap multipoint BFD session on a tail.

5. Operation of Multipoint BFD with Active Tail over P2MP MPLS LSP

[RFC8562] defined how the BFD Demand mode can be used in multipoint networks. When applied in MPLS, procedures specified in [RFC8562] allow an egress LSR to detect a failure of the part of the MPLS p2mp LSP from the ingress LSR. The ingress LSR is not aware of the state of the p2mp LSP. [RFC8563], using mechanisms defined in [RFC8562], defined an "active tail" behavior. An active tail might notify the head of the detected failure and responds to a poll sequence initiated by the head. The first method, referred to as Head Notification without Polling, is mentioned in Section 5.2.1 [RFC8563], is the simplest of all described in [RFC8563]. The use of this method in BFD over MPLS p2mp LSP is discussed in this document. Analysis of other methods of a head learning of the state of an MPLS p2mp LSP is outside the scope of this document.

As specified in [RFC8563] for the active tail mode, BFD variables MUST be as follows:

On an ingress LSR:

- * bfd.SessionType is MultipointHead;
- * bfd.RequiredMinRxInterval is set to nonzero, allowing egress LSRs to send BFD Control packets.

On an egress LSR:

- * bfd.SessionType is MultipointTail;
- * bfd.SilentTail is set to zero.

In Section 5.2.1 [RFC8563] is noted that "the tail sends unsolicited BFD packets in response to the detection of a multipoint path failure" but without the specifics on the information in the packet and frequency of transmissions. This document defines below the procedure of an active tail with unsolicited notifications for p2mp MPLS LSP.

Upon detecting the failure of the p2mp MPLS LSP, an egress LSR sends BFD Control packet with the following settings:

- * the Poll (P) bit is set;
- * the Status (Sta) field set to Down value;
- * the Diagnostic (Diag) field set to Control Detection Time Expired value;
- * the value of the Your Discriminator field is set to the value the egress LSR has been using to demultiplex that BFD multipoint session;
- * BFD Control packet MAY be encapsulated in IP/UDP with the destination IP address of the ingress LSR and the UDP destination port number set to 4784 per [RFC5883]. If non-IP encapsulation is used, then a BFD Control packet is encapsulated using PW-ACH encapsulation (without IP/UDP Headers) (0x0007) [RFC5885];
- * these BFD Control packets are transmitted at the rate of one per second until either it receives a control packet valid for this BFD session with the Final (F) bit set from the ingress LSR or the defect condition clears; however to improve the likelihood of notifying the ingress LSR of the failure of the p2mp MPLS LSP, the egress LSR SHOULD initially transmit three BFD Control packets defined above in short succession.

An ingress LSR that has received the BFD Control packet, as described above, sends the unicast IP/UDP encapsulated BFD Control packet with the Final (F) bit set to the egress LSR.

6. Security Considerations

This document does not introduce new security aspects but inherits all security considerations from [RFC5880], [RFC5884], [RFC7726], [RFC8562], [RFC8029], and [RFC6425].

Also, BFD for p2mp MPLS LSP MUST follow the requirements listed in section 4.1 [RFC4687] to avoid congestion in the control plane or the data plane caused by the rate of generating BFD Control packets. An operator SHOULD consider the amount of extra traffic generated by p2mp BFD when selecting the interval at which the MultipointHead will transmit BFD Control packets. The operator MAY consider the size of the packet the MultipointHead transmits periodically as using IP/UDP encapsulation, which adds up to 28 octets, more than 50% of the BFD Control packet length, comparing to G-ACh encapsulation.

7. IANA Considerations

IANA is requested to allocate value (TBA1) from its MPLS Generalized Associated Channel (G-ACh) Types registry.

Value	Description	Reference
TBA1	Multipoint BFD Session	This document

Table 1: Multipoint BFD Session G-ACh Type

8. Acknowledgements

The authors sincerely appreciate the comments received from Andrew Malis, Italo Busi, Shraddha Hegde, and thought stimulating questions from Carlos Pignataro.

9. References

9.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC5586] Bocci, M., Ed., Vigoureux, M., Ed., and S. Bryant, Ed., "MPLS Generic Associated Channel", RFC 5586, DOI 10.17487/RFC5586, June 2009, <<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>>.

- [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>>.
- [RFC6425] Saxena, S., Ed., Swallow, G., Ali, Z., Farrel, A., Yasukawa, S., and T. Nadeau, "Detecting Data-Plane Failures in Point-to-Multipoint MPLS - Extensions to LSP Ping", RFC 6425, DOI 10.17487/RFC6425, November 2011, <<https://www.rfc-editor.org/info/rfc6425>>.
- [RFC6790] Kompella, K., Drake, J., Amante, S., Henderickx, W., and L. Yong, "The Use of Entropy Labels in MPLS Forwarding", RFC 6790, DOI 10.17487/RFC6790, November 2012, <<https://www.rfc-editor.org/info/rfc6790>>.
- [RFC7212] Frost, D., Bryant, S., and M. Bocci, "MPLS Generic Associated Channel (G-ACh) Advertisement Protocol", RFC 7212, DOI 10.17487/RFC7212, June 2014, <<https://www.rfc-editor.org/info/rfc7212>>.
- [RFC7726] Govindan, V., Rajaraman, K., Mirsky, G., Akiya, N., and S. Aldrin, "Clarifying Procedures for Establishing BFD Sessions for MPLS Label Switched Paths (LSPs)", RFC 7726, DOI 10.17487/RFC7726, January 2016, <<https://www.rfc-editor.org/info/rfc7726>>.
- [RFC8029] Kompella, K., Swallow, G., Pignataro, C., Ed., Kumar, N., Aldrin, S., and M. Chen, "Detecting Multiprotocol Label Switched (MPLS) Data-Plane Failures", RFC 8029, DOI 10.17487/RFC8029, March 2017, <<https://www.rfc-editor.org/info/rfc8029>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.

- [RFC8287] Kumar, N., Ed., Pignataro, C., Ed., Swallow, G., Akiya, N., Kini, S., and M. Chen, "Label Switched Path (LSP) Ping/Traceroute for Segment Routing (SR) IGP-Prefix and IGP-Adjacency Segment Identifiers (SIDs) with MPLS Data Planes", RFC 8287, DOI 10.17487/RFC8287, December 2017, <<https://www.rfc-editor.org/info/rfc8287>>.
- [RFC8562] Katz, D., Ward, D., Pallagatti, S., Ed., and G. Mirsky, Ed., "Bidirectional Forwarding Detection (BFD) for Multipoint Networks", RFC 8562, DOI 10.17487/RFC8562, April 2019, <<https://www.rfc-editor.org/info/rfc8562>>.
- [RFC8563] Katz, D., Ward, D., Pallagatti, S., Ed., and G. Mirsky, Ed., "Bidirectional Forwarding Detection (BFD) Multipoint Active Tails", RFC 8563, DOI 10.17487/RFC8563, April 2019, <<https://www.rfc-editor.org/info/rfc8563>>.

9.2. Informative References

- [RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, DOI 10.17487/RFC4291, February 2006, <<https://www.rfc-editor.org/info/rfc4291>>.
- [RFC4687] Yasukawa, S., Farrel, A., King, D., and T. Nadeau, "Operations and Management (OAM) Requirements for Point-to-Multipoint MPLS Networks", RFC 4687, DOI 10.17487/RFC4687, September 2006, <<https://www.rfc-editor.org/info/rfc4687>>.
- [RFC9026] Morin, T., Ed., Kebler, R., Ed., and G. Mirsky, Ed., "Multicast VPN Fast Upstream Failover", RFC 9026, DOI 10.17487/RFC9026, April 2021, <<https://www.rfc-editor.org/info/rfc9026>>.

Authors' Addresses

Greg Mirsky
Ericsson

Email: gregimirsky@gmail.com

Gyan Mishra
Verizon Inc.

Email: gyan.s.mishra@verizon.com

Donald Eastlake, 3rd
Futurewei Technologies
2386 Panoramic Circle
Apopka, FL 32703
United States of America

Email: d3e3e3@gmail.com

PIM Working Group
Internet-Draft
Updates: 7761 (if approved)
Intended status: Standards Track
Expires: December 29, 2018

G. Mirsky
ZTE Corp.
J. Xiaoli
ZTE Corporation
June 27, 2018

Bidirectional Forwarding Detection (BFD) for Multi-point Networks and
Protocol Independent Multicast - Sparse Mode (PIM-SM) Use Case
draft-mirsky-pim-bfd-p2mp-use-case-02

Abstract

This document discusses the use of Bidirectional Forwarding Detection (BFD) for multi-point networks to provide nodes that participate in Protocol Independent Multicast - Sparse Mode (PIM-SM) with the sub-second convergence. Optional extension to PIM-SM Hello, as specified in RFC 7761, to bootstrap point-to-multipoint BFD session. also defined in this document.

Status of This Memo

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

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

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

This Internet-Draft will expire on December 29, 2018.

Copyright Notice

Copyright (c) 2018 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 (<https://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	2
1.1. Conventions used in this document	3
1.1.1. Terminology	3
1.1.2. Requirements Language	3
2. Problem Statement	3
3. Applicability of p2mp BFD	3
3.1. Multipoint BFD Encapsulation	4
4. IANA Considerations	5
5. Security Considerations	5
6. Acknowledgments	5
7. Normative References	5
Authors' Addresses	6

1. Introduction

Faster convergence in the control plane, in general, is beneficial and allows minimizing periods of traffic blackholing, transient routing loops and other scenarios that may negatively affect service data flow. That equally applies to unicast and multicast routing protocols.

[RFC7761] is the current specification of the Protocol Independent Multicast - Sparse Mode (PIM-SM) for IPv4 and IPv6 networks. Confirming implementation of PIM-SM elects a Designated Router (DR) on each PIM-SM interface. When a group of PIM-SM nodes is connected to shared-media segment, e.g. Ethernet, the one elected as DR is to act on behalf of directly connected hosts in context of the PIM-SM protocol. Failure of the DR impacts the quality of the multicast services it provides to directly connected hosts because the default failure detection interval for PIM-SM routers is 105 seconds. Introduction of Backup DR (BDR), proposed in [I-D.ietf-pim-dr-improvement] improves convergence time in the PIM-SM over shared-media segment but still depends on long failure detection interval.

Bidirectional Forwarding Detection (BFD) [RFC5880] had been originally defined to detect failure of point-to-point (p2p) paths - single-hop [RFC5881], multihop [RFC5883]. [I-D.ietf-bfd-multipoint] extends the BFD base specification [RFC5880] for multipoint and multicast networks, which precisely characterizes deployment scenarios for PIM-SM over LAN segment. This document demonstrates how point-to-multipoint (p2mp) BFD can enable faster detection of

PIM-SM router ailure and thus minimize multicast service disruption. The document also defines the extension to PIM-SM [RFC7761] to bootstrap a PIM-SM router to join in p2mp BFD session over shared-media link.

1.1. Conventions used in this document

1.1.1. Terminology

BFD: Bidirectional Forwarding Detection

BDR: Backup Designated Router

DR: Designated Router

p2mp: Pont-to-Multipoint

PIM-SM: Protocol Independent Multicast - Sparse Mode

1.1.2. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

2. Problem Statement

[RFC7761] does not provide a method for fast, e.g. sub-second, failure detection of a neighbor PIM-SM router. BFD already has many implementations based on HW that are capable to support multiple sub-second session concurrently.

3. Applicability of p2mp BFD

[I-D.ietf-bfd-multipoint] may provide the efficient and scalable solution for the fast-converging environment that has head-tails relationships. Each such group presents itself as p2mp BFD session with its head being the root and other routers being tails of the p2mp BFD session. Figure 1 displays the new BFD Discriminator TLV [RFC7761] to bootstrap tail of the p2mp BFD session.

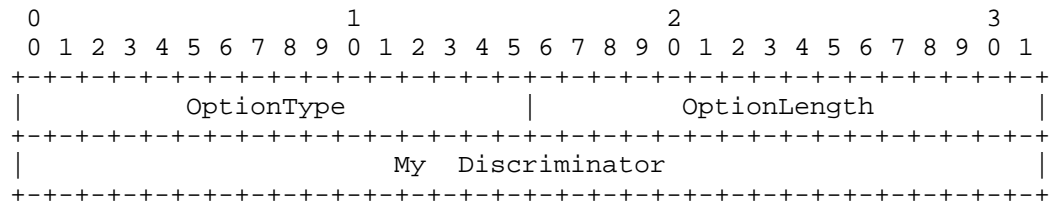


Figure 1: BFD Discriminator TLV to Bootstrap P2MP BFD session

where new fields are interpreted as:

OptionType is a value (TBA1) assigned by IANA Section 4 that identifies the TLV as BFD Discriminator TLV;

OptionLength value is always 4

My Discriminator - My Discriminator value allocated by the root of the p2mp BFD session.

If PIM-SM routers, that support this specification, are configured to use p2mp BFD for faster convergence, then the router to be monitored, referred to as 'head', MUST create BFD session MultipointHead, as defined in [I-D.ietf-bfd-multipoint]. The head MUST include BFD TLV in its PIM-Hello message and periodically transmit BFD control packets. Source IP address of the BFD control packet MUST be the same as the source IP address of the PIM-Hello with BFD TLV messages being transmitted by the head. The values of My Discriminator in the BFD control packet and My Discriminator field of the BFD TLV in PIM-Hello, transmitted by the head MUST be the same. When a PIM-SM router configured to monitor the head, referred to as 'tail', via p2mp BFD receives PIM-Hello packet with BFD TLV it MAY create p2mp BFD session as MultipointTail, as defined in [I-D.ietf-bfd-multipoint], and demultiplex p2mp BFD test session based on head's source IP address the My Discriminator value it learned from BFD Discriminator TLV. If the head ceased to include BFD TLV in its PIM-Hello message, tails MUST close the corresponding MultipointTail BFD session. If the tail detects MultipointHead failure it MUST remove the neighbor. If the failed head node was PIM-SM DR or BDR the tail MAY start DR Election process as specified in Section 4.3.2 [RFC7761] or in Section 4.1 [I-D.ietf-pim-dr-improvement] respectively.

3.1. Multipoint BFD Encapsulation

The MultipointHead of p2mp BFD session when transmitting BFD control packet:

MUST set TTL value to 1;

SHOULD use group address ALL-PIM-ROUTERS ('224.0.0.13' for IPv4 and 'ff02::d' for IPv6) as destination IP address

MAY use network broadcast address for IPv4 or link-local all nodes multicast group for IPv6 as the destination IP address;

MUST set destination UDP port value to 3784 when transmitting BFD control packets, as defined in [I-D.ietf-bfd-multipoint].

4. IANA Considerations

IANA is requested to allocate a new OptionType value from PIM Hello Options registry according to:

Value Name	Length Number	Name Protocol	Reference
TBA	4	BFD Discriminator	This document

Table 1: BFD Discriminator option type

5. Security Considerations

Security considerations discussed in [RFC7761], [RFC5880], and [I-D.ietf-bfd-multipoint], apply to this document.

6. Acknowledgments

Authors cannot say enough to express their appreciation of comments and suggestions we received from Stig Venaas.

7. Normative References

- [I-D.ietf-bfd-multipoint]
 Katz, D., Ward, D., Networks, J., and G. Mirsky, "BFD for Multipoint Networks", draft-ietf-bfd-multipoint-18 (work in progress), June 2018.
- [I-D.ietf-pim-dr-improvement]
 Zhang, Z., hu, f., Xu, B., and m. mishra, "PIM DR Improvement", draft-ietf-pim-dr-improvement-04 (work in progress), December 2017.

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [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>>.
- [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>>.
- [RFC7761] Fenner, B., Handley, M., Holbrook, H., Kouvelas, I., Parekh, R., Zhang, Z., and L. Zheng, "Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised)", STD 83, RFC 7761, DOI 10.17487/RFC7761, March 2016, <<https://www.rfc-editor.org/info/rfc7761>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.

Authors' Addresses

Greg Mirsky
ZTE Corp.

Email: gregimirsky@gmail.com

Ji Xiaoli
ZTE Corporation
No.50 Software Avenue, Yuhuatai District
Nanjing
China

Email: ji.xiaoli@zte.com.cn

SPRING Working Group
Internet-Draft
Intended status: Standards Track
Expires: October 28, 2020

G. Mirsky
ZTE Corp.
J. Tantsura
Apstra, Inc.
I. Varlashkin
Google
M. Chen
Huawei
J. Wenying
CMCC
April 26, 2020

Bidirectional Forwarding Detection (BFD) in Segment Routing Networks
Using MPLS Dataplane
draft-mirsky-spring-bfd-10

Abstract

Segment Routing (SR) architecture leverages the paradigm of source routing. It can be realized in the Multiprotocol Label Switching (MPLS) network without any change to the data plane. A segment is encoded as an MPLS label, and an ordered list of segments is encoded as a stack of labels. Bidirectional Forwarding Detection (BFD) is expected to monitor any existing path between systems. This document defines how to use Label Switched Path Ping to bootstrap a BFD session, control an SR Policy in the reverse direction of the SR-MPLS tunnel, and applicability of BFD Demand mode in the SR-MPLS domain. Also, the document describes the use of BFD Echo with BFD Control packet payload.

Status of This Memo

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

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

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

This Internet-Draft will expire on October 28, 2020.

Copyright Notice

Copyright (c) 2020 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 (<https://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	2
1.1. Conventions	3
1.1.1. Terminology	3
1.1.2. Requirements Language	3
2. Bootstrapping BFD Session over Segment Routed Tunnel with MPLS Data Plane	4
3. Use BFD Reverse Path TLV over Segment Routed MPLS Tunnel	5
4. Use Non-FEC Path TLV	5
5. BFD Reverse Path TLV over Segment Routed MPLS Tunnel with Dynamic Control Plane	7
6. Applicability of BFD Demand Mode in SR-MPLS Domain	7
7. Using BFD to Monitor Point-to-Multipoint SR Policy	7
8. Use of Echo BFD in SR-MPLS	8
9. IANA Considerations	8
9.1. Non-FEC Path TLV	8
9.2. Return Code	9
10. Implementation Status	10
11. Security Considerations	10
12. Contributors	11
13. Acknowledgments	11
14. References	11
14.1. Normative References	11
14.2. Informative References	13
Authors' Addresses	13

1. Introduction

[RFC5880], [RFC5881], and [RFC5883] defined the operation of Bidirectional Forwarding Detection (BFD) protocol between the two systems over IP networks. [RFC5884] and [RFC7726] set rules for using BFD Asynchronous mode over point-to-point (p2p) Multiprotocol

Label Switching (MPLS) Label Switched Path (LSP). These latter standards implicitly assume that the remote BFD system, which is at the egress Label Edge Router (LER), will use the shortest path route regardless of the path the BFD system at the ingress LER uses to send BFD Control packets towards it. Throughout this document, references to ingress LER and egress LER are used, respectively, as a shortened version of the "BFD system at the ingress/egress LER".

This document defines the use of LSP Ping for Segment Routing networks over MPLS data plane [RFC8287] to bootstrap and control path of a BFD session from the egress to ingress LER using Segment Routing tunnel with MPLS data plane (SR-MPLS).

1.1. Conventions

1.1.1. Terminology

BFD: Bidirectional Forwarding Detection

BSID: Binding Segment Identifier

FEC: Forwarding Equivalence Class

MPLS: Multiprotocol Label Switching

SR-MPLS Segment Routing with MPLS data plane

LSP: Label Switched Path

LER Label Edge Router

p2p Point-to-point

p2mp Point-to-multipoint

SID Segment Identifier

SR Segment Routing

1.1.2. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

2. Bootstrapping BFD Session over Segment Routed Tunnel with MPLS Data Plane

Use of an LSP Ping to bootstrap BFD over MPLS LSP is required, as documented in [RFC5884], to establish an association between a fault detection message, i.e., BFD Control message, and the Forwarding Equivalency Class (FEC) of a single label stack LSP in case of Penultimate Hop Popping or when the egress LER distributes the Explicit NULL label to the penultimate hop router. The Explicit NULL label is not advertised as a Segment Identifier (SID) by an SR node but, as demonstrated in section 3.1 [RFC8660] if the operation at the penultimate hop is NEXT; then the egress SR node will receive an IP encapsulated packet. Thus the conclusion is that LSP Ping MUST be used to bootstrap a BFD session in an SR-MPLS domain if there are no other means to bootstrap the BFD session, e.g., using an extension to a dynamic routing protocol as described in [I-D.ietf-bess-mvpn-fast-failover] and [I-D.ietf-pim-bfd-p2mp-use-case].

As demonstrated in [RFC8287], the introduction of Segment Routing network domains with an MPLS data plane requires three new sub-TLVs that MAY be used with Target FEC TLV. Section 6.1 addresses the use of the new sub-TLVs in Target FEC TLV in LSP ping and LSP traceroute. For the case of LSP ping, the [RFC8287] states that:

The initiator, i.e., ingress LER, MUST include FEC(s) corresponding to the destination segment.

The initiator MAY include FECs corresponding to some or all of segments imposed in the label stack by the ingress LER to communicate the segments traversed.

It has been noted in [RFC5884] that a BFD session monitors for defects particular <MPLS LSP, FEC> tuple. [RFC7726] clarified how to establish and operate multiple BFD sessions for the same <MPLS LSP, FEC> tuple. Because only the ingress LER is aware of the SR-based explicit route, the egress LER can associate the LSP ping with BFD Discriminator TLV with only one of the FECs it advertised for the particular segment. Thus this document clarifies that:

When LSP Ping is used to bootstrapping a BFD session for SR-MPLS tunnel the FEC corresponding to the segment to be associated with the BFD session MUST be as the very last sub-TLV in the Target FEC TLV.

If the target segment is an anycast prefix segment ([I-D.ietf-spring-mpls-anycast-segments]) the corresponding Anycast SID MUST be included in the Target TLV as the very last sub-TLV.

Also, for BFD Control packet the ingress SR node MUST use precisely the same label stack encapsulation, especially Entropy Label ([RFC6790]), as for the LSP ping with the BFD Discriminator TLV that bootstrapped the BFD session. Other operational aspects of using BFD to monitor the continuity of the path to the particular Anycast SID, advertised by a group of SR-MPLS capable nodes, will be considered in the future versions of the document.

Encapsulation of a BFD Control packet in Segment Routing network with MPLS data plane MUST follow Section 7 [RFC5884] when the IP/UDP header used and MUST follow Section 3.4 [RFC6428] without IP/UDP header being used.

3. Use BFD Reverse Path TLV over Segment Routed MPLS Tunnel

For BFD over MPLS LSP case, per [RFC5884], egress LER MAY send BFD Control packet to the ingress LER either over IP network or an MPLS LSP. Similarly, for the case of BFD over p2p SR-MPLS tunnel, the egress LER MAY route BFD Control packet over the IP network, as described in [RFC5883], or transmit over a segment tunnel, as described in Section 7 [RFC5884]. In some cases, there may be a need to direct egress LER to use a specific path for the reverse direction of the BFD session by using the BFD Reverse Path TLV and following all procedures as defined in [I-D.ietf-mpls-bfd-directed].

4. Use Non-FEC Path TLV

For the case of MPLS data plane, Segment Routing Architecture [RFC8402] explains that "a segment is encoded as an MPLS label. An ordered list of segments is encoded as a stack of labels."

This document defines a new optional Non-FEC Path TLV. The format of the Non-FEC Path TLV is presented in Figure 1

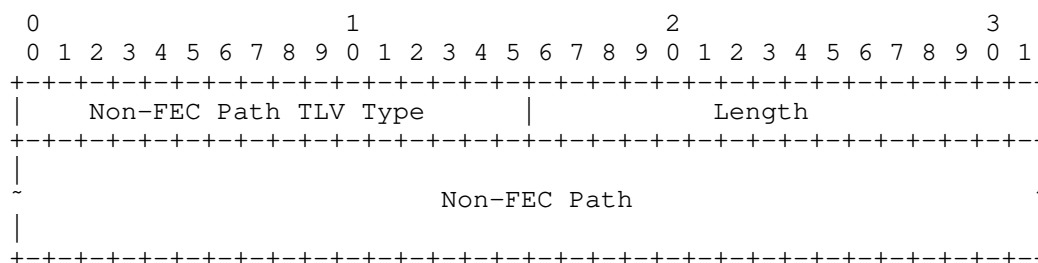


Figure 1: Non-FEC Path TLV Format

Non-FEC Path TLV Type is two octets in length and has a value of TBD1 (to be assigned by IANA as requested in Section 9.1).

Length field is two octets long and defines the length in octets of the Non-FEC Path field.

Non-FEC Path field contains a sub-TLV. Any Non-FEC Path sub-TLV (defined in this document or to be defined in the future) for Non-FEC Path TLV type MAY be used in this field. None or one sub-TLV MAY be included in the Non-FEC Path TLV. If no sub-TLV has been found in the Non-FEC Path TLV, the egress LER MUST revert to using the reverse path selected based on its local policy. If there is more than one sub-TLV, then the Return Code in echo reply MUST be set to value TBD3 "Too Many TLVs Detected" (to be assigned by IANA as requested in Table 4).

Non-FEC Path TLV MAY be used to specify the reverse path of the BFD session identified in the BFD Discriminator TLV. If the Non-FEC Path TLV is present in the echo request message the BFD Discriminator TLV MUST be present as well. If the BFD Discriminator TLV is absent when the Non-FEC Path TLV is included, then it MUST be treated as malformed Echo Request, as described in [RFC8029].

This document defines the Segment Routing MPLS Tunnel sub-TLV that MAY be used with the Non-FEC Path TLV. The format of the sub-TLV is presented in Figure 2.

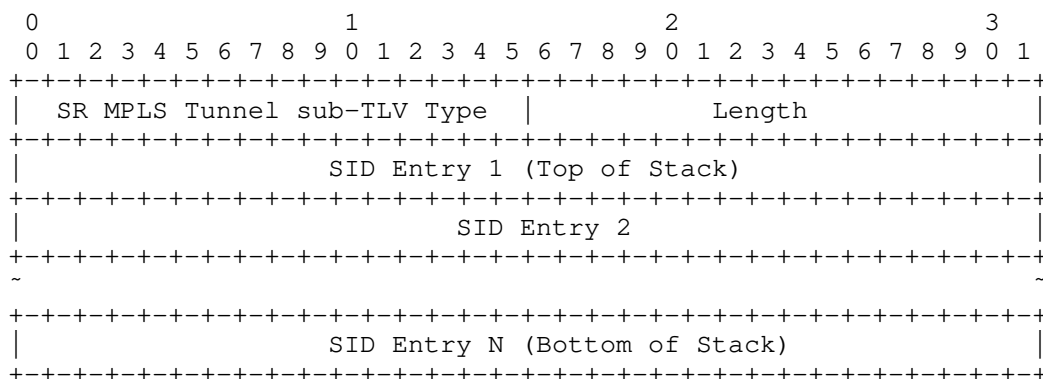


Figure 2: Segment Routing MPLS Tunnel sub-TLV

The Segment Routing MPLS Tunnel sub-TLV Type is two octets in length, and has a value of TBD2 (to be assigned by IANA as requested in Section 9.1).

The egress LER MUST use the Value field as label stack for BFD Control packets for the BFD session identified by the source IP

address of the MPLS LSP Ping packet and the value in the BFD Discriminator TLV. Label Entries MUST be in network order.

5. BFD Reverse Path TLV over Segment Routed MPLS Tunnel with Dynamic Control Plane

When Segment Routed domain with MPLS data plane uses distributed tunnel computation BFD Reverse Path TLV MAY use Target FEC sub-TLVs defined in [RFC8287].

6. Applicability of BFD Demand Mode in SR-MPLS Domain

[I-D.mirsky-bfd-mpls-demand] defines how Demand mode of BFD, specified in sections 6.6 and 6.18.4 of [RFC5880], can be used to monitor uni-directional MPLS LSP. Similar procedures can be following in SR-MPLS to monitor uni-directional SR tunnels:

- o an ingress SR node bootstraps BFD session over SR-MPLS in Async BFD mode;
- o once BFD session is Up, the ingress SR node switches the egress LER into the Demand mode by setting D field in BFD Control packet it transmits;
- o if the egress LER detects the failure of the BFD session, it sends its BFD Control packet to the ingress SR node over the IP network with a Poll sequence;
- o if the ingress SR node receives a BFD Control packet from the remote node in a Demand mode with Poll sequence and Diag field indicating the failure, the ingress SR node transmits BFD Control packet with Final over IP and switches the BFD over SR-MPLS back into Async mode, sending BFD Control packets one per second.

7. Using BFD to Monitor Point-to-Multipoint SR Policy

[I-D.voyer-spring-sr-p2mp-policy] defined variants of SR Policy to deliver point-to-multipoint (p2mp) services. For the given P2MP segment [RFC8562] can be used if, for example, leaves have an alternative source of the multicast service flow to select. In such a scenario, a leaf may switch to using the alternative flow after p2mp BFD detects the failure in the working multicast path. For scenarios where it is required for the root to monitor the state of the multicast tree [RFC8563] can be used. The root may use the detection of the failure of the multicast tree to the particular leaf to restore the path for that leaf or re-instantiate the whole multicast tree.

An essential part of using p2mp BFD is the bootstrapping the BFD session at all the leaves. The root, acting as the MultipointHead, MAY use LSP Ping with the BFD Discriminator TLV. Alternatively, extensions to routing protocols, e.g., BGP, or management plane, e.g., PCEP, MAY be used to associate the particular P2MP segment with MultipointHead's Discriminator. Extensions for routing protocols and management plane are for further study.

8. Use of Echo BFD in SR-MPLS

Echo-BFD [RFC5880] can be used to monitor an SR Policy between the local and the remote BFD peers. As defined in [RFC5880], the remote BFD system does not process the payload of an Echo BFD. Thus it is the local system that demultiplexes the Echo BFD packet matching it to the appropriate BFD session and detects missing Echo BFD packets. A BFD Control packet MAY be used as the payload of Echo BFD. This specification defines the use of Echo BFD in SR-MPLS network with BFD Control packet as the payload. The use of other types of Echo BFD payload is outside the scope of this document. Because the remote BFD system does not process Echo BFD, the value of the Your Discriminator field MUST be set to the discriminator the local BFD system assigned to the given BFD session. My Discriminator field MUST be zeroed. Authentication MUST be set according to the configuration of the BFD session. To ensure that the Echo BFD packet is returned to the sender without being processed, the sender MAY use a Binding SID (BSID) [RFC8402] that has been bound with the SR Policy that ensures the return of a packet to that particular node. A BSID MAY be associated with the SR Policy that is the reverse to the SR Policy programmed onto the BFD Echo packet by the sender.

9. IANA Considerations

9.1. Non-FEC Path TLV

IANA is requested to assign new TLV type from the from Standards Action range of the registry "Multiprotocol Label Switching Architecture (MPLS) Label Switched Paths (LSPs) Ping Parameters - TLVs" as defined in Table 1.

Value	TLV Name	Reference
TBD1	Non-FEC Path TLV	This document

Table 1: New Non-FEC Path TLV

IANA is requested to create new Non-FEC Path sub-TLV registry for the Non-FEC Path TLV, as described in Table 2.

Range	Registration Procedures	Note
0-16383	Standards Action	This range is for mandatory TLVs or for optional TLVs that require an error message if not recognized. Experimental RFC needed
16384-31743	Specification Required	
32768-49161	Standards Action	This range is for optional TLVs that can be silently dropped if not recognized. Experimental RFC needed
49162-64511	Specification Required	
64512-65535	Private Use	

Table 2: Non-FEC Path sub-TLV registry

IANA is requested to allocate the following values from the Non-FEC Path sub-TLV registry as defined in Table 3.

Value	Description	Reference
0	Reserved	This document
TBD2	Segment Routing MPLS Tunnel sub-TLV	This document
65535	Reserved	This document

Table 3: New Segment Routing Tunnel sub-TLV

9.2. Return Code

IANA is requested to create Non-FEC Path sub-TLV sub-registry for the new Non-FEC Path TLV and assign a new Return Code value from the "Multi-Protocol Label Switching (MPLS) Label Switched Paths (LSPs) Ping Parameters" registry, "Return Codes" sub-registry, as follows using a Standards Action value.

Value	Description	Reference
X TBD3	Too Many TLVs Detected.	This document

Table 4: New Return Code

10. Implementation Status

- The organization responsible for the implementation: ZTE Corporation.
- The implementation's name ROSng SW empowers traditional routers, e.g., ZXCTN 6000.
- A brief general description: A list of SIDs can be specified as the Return Path for an SR-MPLS tunnel.
- The implementation's level of maturity: production.
- Coverage: complete
- Version compatibility: draft-mirsky-spring-bfd-06.
- Licensing: proprietary.
- Implementation experience: Appreciate Early Allocation of values for Non-FEC TLV and Segment Routing MPLS Tunnel sub-TLV (using Private Use code points).
- Contact information: Qian Xin qian.xin2@zte.com.cn
- The date when information about this particular implementation was last updated: 12/16/2019

Note to RFC Editor: This section MUST be removed before publication of the document.

11. Security Considerations

Security considerations discussed in [RFC5880], [RFC5884], [RFC7726], and [RFC8029] apply to this document.

12. Contributors

Xiao Min
ZTE Corp.
Email: xiao.min2@zte.com.cn

13. Acknowledgments

Authors greatly appreciate the help of Qian Xin, who provided the information about the implementation of this specification.

14. References

14.1. Normative References

- [I-D.ietf-mpls-bfd-directed]
Mirsky, G., Tantsura, J., Varlashkin, I., and M. Chen,
"Bidirectional Forwarding Detection (BFD) Directed Return
Path", draft-ietf-mpls-bfd-directed-13 (work in progress),
December 2019.
- [I-D.mirsky-bfd-mpls-demand]
Mirsky, G., "BFD in Demand Mode over Point-to-Point MPLS
LSP", draft-mirsky-bfd-mpls-demand-06 (work in progress),
December 2019.
- [I-D.voyer-spring-sr-p2mp-policy]
daniel.voyer@bell.ca, d., Filsfils, C., Parekh, R.,
Bidgoli, H., and Z. Zhang, "SR Replication Policy for P2MP
Service Delivery", draft-voyer-spring-sr-p2mp-policy-03
(work in progress), July 2019.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate
Requirement Levels", BCP 14, RFC 2119,
DOI 10.17487/RFC2119, March 1997,
<<https://www.rfc-editor.org/info/rfc2119>>.
- [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>>.

- [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>>.
- [RFC6428] Allan, D., Ed., Swallow, G., Ed., and J. Drake, Ed., "Proactive Connectivity Verification, Continuity Check, and Remote Defect Indication for the MPLS Transport Profile", RFC 6428, DOI 10.17487/RFC6428, November 2011, <<https://www.rfc-editor.org/info/rfc6428>>.
- [RFC7726] Govindan, V., Rajaraman, K., Mirsky, G., Akiya, N., and S. Aldrin, "Clarifying Procedures for Establishing BFD Sessions for MPLS Label Switched Paths (LSPs)", RFC 7726, DOI 10.17487/RFC7726, January 2016, <<https://www.rfc-editor.org/info/rfc7726>>.
- [RFC8029] Kompella, K., Swallow, G., Pignataro, C., Ed., Kumar, N., Aldrin, S., and M. Chen, "Detecting Multiprotocol Label Switched (MPLS) Data-Plane Failures", RFC 8029, DOI 10.17487/RFC8029, March 2017, <<https://www.rfc-editor.org/info/rfc8029>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [RFC8287] Kumar, N., Ed., Pignataro, C., Ed., Swallow, G., Akiya, N., Kini, S., and M. Chen, "Label Switched Path (LSP) Ping/Traceroute for Segment Routing (SR) IGP-Prefix and IGP-Adjacency Segment Identifiers (SIDs) with MPLS Data Planes", RFC 8287, DOI 10.17487/RFC8287, December 2017, <<https://www.rfc-editor.org/info/rfc8287>>.
- [RFC8402] Filss, C., Ed., Previdi, S., Ed., Ginsberg, L., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing Architecture", RFC 8402, DOI 10.17487/RFC8402, July 2018, <<https://www.rfc-editor.org/info/rfc8402>>.
- [RFC8562] Katz, D., Ward, D., Pallagatti, S., Ed., and G. Mirsky, Ed., "Bidirectional Forwarding Detection (BFD) for Multipoint Networks", RFC 8562, DOI 10.17487/RFC8562, April 2019, <<https://www.rfc-editor.org/info/rfc8562>>.

- [RFC8563] Katz, D., Ward, D., Pallagatti, S., Ed., and G. Mirsky, Ed., "Bidirectional Forwarding Detection (BFD) Multipoint Active Tails", RFC 8563, DOI 10.17487/RFC8563, April 2019, <<https://www.rfc-editor.org/info/rfc8563>>.
- [RFC8660] Bashandy, A., Ed., Filsfils, C., Ed., Previdi, S., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing with the MPLS Data Plane", RFC 8660, DOI 10.17487/RFC8660, December 2019, <<https://www.rfc-editor.org/info/rfc8660>>.

14.2. Informative References

- [I-D.ietf-bess-mvpn-fast-failover]
Morin, T., Kebler, R., and G. Mirsky, "Multicast VPN fast upstream failover", draft-ietf-bess-mvpn-fast-failover-10 (work in progress), February 2020.
- [I-D.ietf-pim-bfd-p2mp-use-case]
Mirsky, G. and J. Xiaoli, "Bidirectional Forwarding Detection (BFD) for Multi-point Networks and Protocol Independent Multicast - Sparse Mode (PIM-SM) Use Case", draft-ietf-pim-bfd-p2mp-use-case-03 (work in progress), January 2020.
- [I-D.ietf-spring-mpls-anycast-segments]
Sarkar, P., Gredler, H., Filsfils, C., Previdi, S., Decraene, B., and M. Horneffer, "Anycast Segments in MPLS based Segment Routing", draft-ietf-spring-mpls-anycast-segments-02 (work in progress), January 2018.
- [RFC6790] Kompella, K., Drake, J., Amante, S., Henderickx, W., and L. Yong, "The Use of Entropy Labels in MPLS Forwarding", RFC 6790, DOI 10.17487/RFC6790, November 2012, <<https://www.rfc-editor.org/info/rfc6790>>.

Authors' Addresses

Greg Mirsky
ZTE Corp.

Email: gregimirsky@gmail.com

Jeff Tantsura
Apstra, Inc.

Email: jefftant.ietf@gmail.com

Ilya Varlashkin
Google

Email: Ilya@nobulus.com

Mach (Guoyi) Chen
Huawei

Email: mach.chen@huawei.com

Jiang Wenying
CMCC

Email: jiangwenying@chinamobile.com

Internet Engineering Task Force
Internet-Draft
Intended status: Standards Track
Expires: April 16, 2018

S. Pallagatti, Ed.
Independent Contributor
S. Paragiri
Juniper Networks
V. Govindan
M. Mudigonda
Cisco
G. Mirsky
ZTE Corp.
October 13, 2017

BFD for VXLAN
draft-spallagatti-bfd-vxlan-06

Abstract

This document describes use of Bidirectional Forwarding Detection (BFD) protocol in Virtual eXtensible Local Area Network (VXLAN) overlay network.

Status of This Memo

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

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

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

This Internet-Draft will expire on April 16, 2018.

Copyright Notice

Copyright (c) 2017 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 (<https://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	2
2. Conventions used in this document	3
2.1. Terminology	3
2.2. Requirements Language	3
3. Use cases	3
4. Deployment	4
5. BFD Packet Transmission over VXLAN Tunnel	5
5.1. BFD Packet Encapsulation in VXLAN	6
6. Reception of BFD packet from VXLAN Tunnel	7
6.1. Demultiplexing of the BFD packet	8
7. Use of reserved VNI	8
8. Echo BFD	8
9. IANA Considerations	8
10. Security Considerations	8
11. Contributors	8
12. Acknowledgments	9
13. Normative References	9
Authors' Addresses	10

1. Introduction

"Virtual eXtensible Local Area Network (VXLAN)" has been described in [RFC7348]. VXLAN provides an encapsulation scheme that allows virtual machines (VMs) to communicate in a data center network.

VXLAN is typically deployed in data centers interconnecting virtualized hosts, which may be spread across multiple racks. The individual racks may be part of a different Layer 3 network or they could be in a single Layer 2 network. The VXLAN segments/overlay networks are overlaid on top of these Layer 2 or Layer 3 networks.

A VM can communicate with another VM only if they are on the same VXLAN. VMs are unaware of VXLAN tunnels as VXLAN tunnel is terminated on VXLAN Tunnel End Point (VTEP) (hypervisor/TOR). VTEPs (hypervisor/TOR) are responsible for encapsulating and decapsulating frames exchanged among VMs.

Since underlay is a L3 network, ability to monitor path continuity, i.e. perform proactive continuity check (CC) for these tunnels is important. Asynchronous mode of BFD, as defined in [RFC5880], can be

used to monitor a VXLAN tunnel. Use of [I-D.ietf-bfd-multipoint] is for future study.

Also BFD in VXLAN can be used to monitor special service nodes that are designated to properly handle Layer 2 broadcast, unknown unicast, and multicast traffic. Such nodes, often referred "replicators", are usually virtual VTEPs can be monitored by physical VTEPs in order to minimize BUM traffic directed to unavailable replicator.

This document describes use of Bidirectional Forwarding Detection (BFD) protocol VXLAN to enable continuity monitoring between Network Virtualization Edges (NVEs) and/or availability of a replicator service node using BFD.

2. Conventions used in this document

2.1. Terminology

BFD - Bidirectional Forwarding Detection

CC - Continuity Check

NVE - Network Virtualization Edge

TOR - Top of Rack

VM - Virtual Machine

VTEP - VXLAN Tunnel End Point

VXLAN - Virtual eXtensible Local Area Network

2.2. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

3. Use cases

Main use case of BFD for VXLAN is for continuity check of a tunnel. By exchanging BFD control packets between VTEPs an operator exercises the VXLAN path in both in underlay and overlay thus ensuring the VXLAN path availability and VTEPs reachability. BFD failure detection can be used for maintenance. There are other use cases such as

Layer 2 VMs:

Most deployments will have VMs with only L2 capabilities that may not support L3. BFD being a L3 protocol can be used as tunnel CC mechanism, where BFD will start and terminate at the NVEs, e.g. VTEPs.

It is possible to aggregate the CC sessions for multiple tenants by running a BFD session between the VTEPs over VxLAN tunnel. In rest of this document terms NVE and VTEP are used interchangeably.

Fault localization:

It is also possible that VMs are L3 aware and can possibly host a BFD session. In these cases BFD sessions can be established among VMs for CC. In addition, BFD sessions can be established among VTEPs for tunnel CC. Having a hierarchical OAM model helps localize faults though requires additional consideration.

Service node reachability:

Service node is responsible for sending BUM traffic. In case of service node tunnel terminates at VTEP and it might not even host VM. BFD session between TOR/hypervisor and service node can be used to monitor service node reachability.

4. Deployment

Figure 1 illustrates the scenario with two servers, each of them hosting two VMs. These servers host VTEPs that terminate two VXLAN tunnels with VNI number 100 and 200. Separate BFD sessions can be established between the VTEPs (IP1 and IP2) for monitoring each of the VXLAN tunnels (VNI 100 and 200). No BFD packets, intended to Hypervisor VTEP, should be forwarded to a VM as VM may drop BFD packets leading to false negative. This method is applicable whether VTEP is a virtual or physical device.

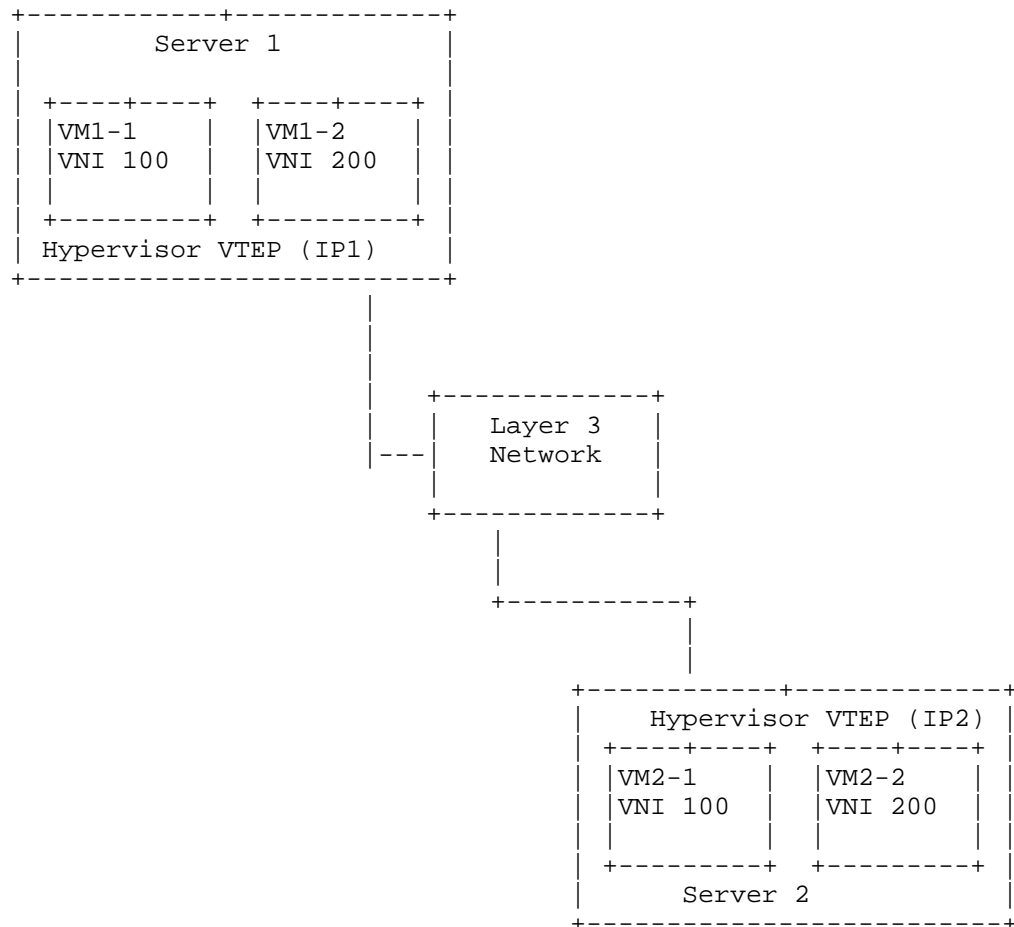


Figure 1: Reference VXLAN domain

5. BFD Packet Transmission over VXLAN Tunnel

BFD packet MUST be encapsulated and sent to a remote VTEP as explained in Section 5.1. Implementations SHOULD ensure that the BFD packets follow the same lookup path of VXLAN packets within the sender system.

5.1. BFD Packet Encapsulation in VXLAN

VXLAN packet format has been described in Section 5 of [RFC7348]. The Outer IP/UDP and VXLAN headers MUST be encoded by the sender as per [RFC7348].

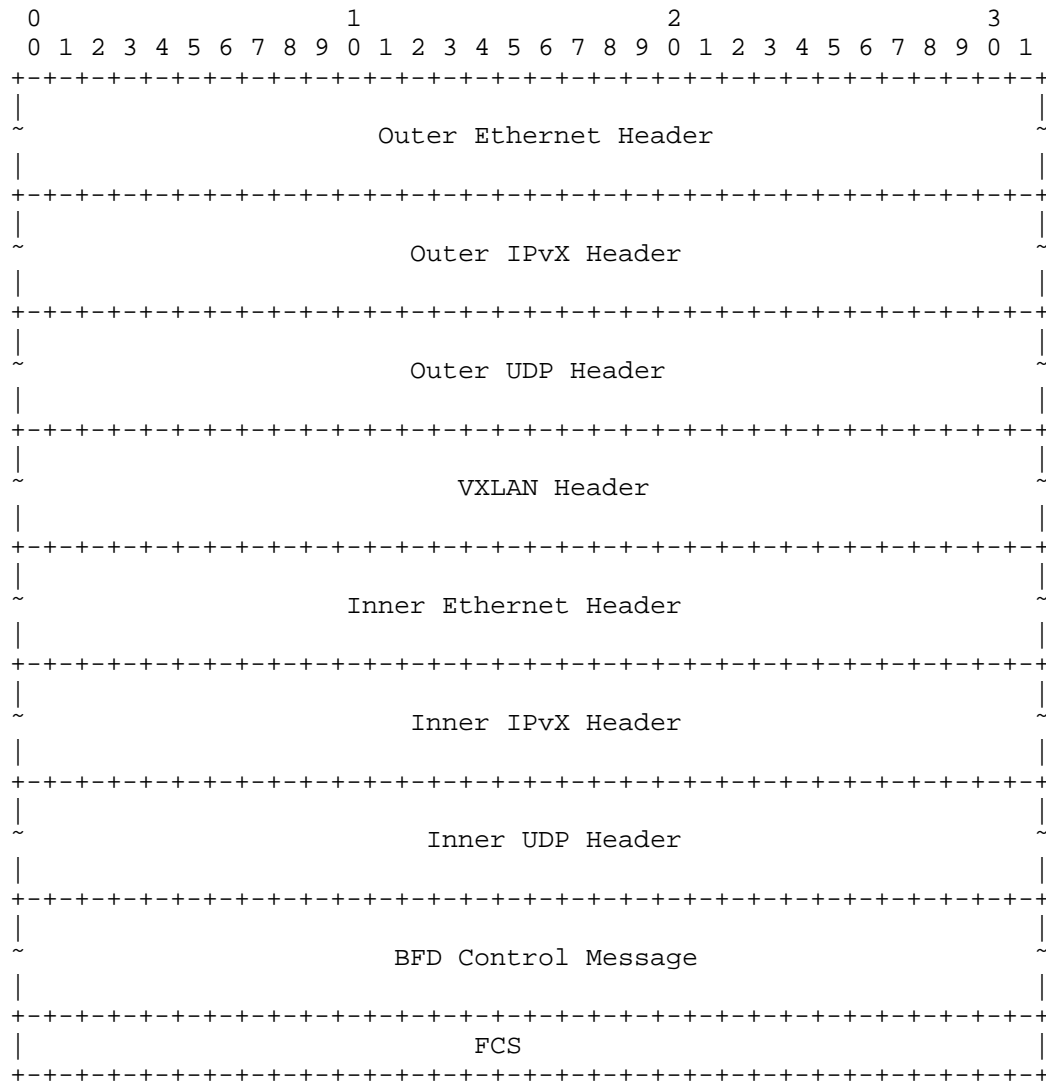


Figure 2: VXLAN Encapsulaion of BFD Control Message

The BFD packet MUST be carried inside the inner MAC frame of the VXLAN packet. The inner MAC frame carrying the BFD payload has the following format:

Ethernet Header:

Destination MAC: This MUST be a dedicated MAC (TBA) Section 9 or the MAC address of the destination VTEP. The details of how the MAC address of the destination VTEP is obtained are outside the scope of this document.

Source MAC: MAC address of the originating VTEP

IP header:

Source IP: IP address of the originating VTEP.

Destination IP: IP address of the terminating VTEP.

TTL: This MUST be set to 1. This is to ensure that the BFD packet is not routed within the L3 underlay network.

[Ed.Note]:Use of inner source and destination IP addresses needs more discussion by the WG.

The fields of the UDP header and the BFD control packet are encoded as specified in [RFC5881] for p2p VXLAN tunnels.

6. Reception of BFD packet from VXLAN Tunnel

Once a packet is received, VTEP MUST validate the packet as described in Section 4.1 of [RFC7348]. If the Destination MAC of the inner MAC frame matches the dedicated MAC or the MAC address of the VTEP the packet MUST be processed further.

The UDP destination port and the TTL of the inner Ethernet frame MUST be validated to determine if the received packet can be processed by BFD. BFD packet with inner MAC set to VTEP or dedicated MAC address MUST NOT be forwarded to VMs.

To ensure BFD detects the proper configuration of VXLAN Network Identifier (VNI) in a remote VTEP, a lookup SHOULD be performed with the MAC-DA and VNI as key in the Virtual Forwarding Instance (VFI) table of the originating/ terminating VTEP in order to exercise the VFI associated with the VNI.

6.1. Demultiplexing of the BFD packet

Demultiplexing of IP BFD packet has been defined in Section 3 of [RFC5881]. Since multiple BFD sessions may be running between two VTEPs, there needs to be a mechanism for demultiplexing received BFD packets to the proper session. The procedure for demultiplexing packets with Your Discriminator equal to 0 is different from [RFC5880]. For such packets, the BFD session MUST be identified using the inner headers, i.e. the source IP and the destination IP present in the IP header carried by the payload of the VXLAN encapsulated packet. The VNI of the packet SHOULD be used to derive interface related information for demultiplexing the packet. If BFD packet is received with non-zero Your Discriminator then BFD session MUST be demultiplexed only with Your Discriminator as the key.

7. Use of reserved VNI

BFD session MAY be established for the reserved VNI 0. One way to aggregate BFD sessions between VTEP's is to establish a BFD session with VNI 0. A VTEP MAY also use VNI 0 to establish a BFD session with a service node.

8. Echo BFD

Support for echo BFD is outside the scope of this document.

9. IANA Considerations

IANA is requested to assign a dedicated MAC address to be used as the Destination MAC address of the inner Ethernet which carries BFD control packet in IP/UDP encapsulation.

10. Security Considerations

Document recommends setting of inner IP TTL to 1 which could lead to DDoS attack, implementation MUST have throttling in place. Throttling MAY be relaxed for BFD packets based on port number.

Other than inner IP TTL set to 1 this specification does not raise any additional security issues beyond those of the specifications referred to in the list of normative references.

11. Contributors

Reshad Rahman
rrahman@cisco.com
Cisco

12. Acknowledgments

Authors would like to thank Jeff Hass of Juniper Networks for his reviews and feedback on this material.

Authors would also like to thank Nobo Akiya, Marc Binderberger and Shahram Davari for the extensive review.

13. Normative References

- [I-D.ietf-bfd-multipoint]
Katz, D., Ward, D., and J. Networks, "BFD for Multipoint Networks", draft-ietf-bfd-multipoint-10 (work in progress), April 2017.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [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>>.
- [RFC7348] Mahalingam, M., Dutt, D., Duda, K., Agarwal, P., Kreeger, L., Sridhar, T., Bursell, M., and C. Wright, "Virtual eXtensible Local Area Network (VXLAN): A Framework for Overlaying Virtualized Layer 2 Networks over Layer 3 Networks", RFC 7348, DOI 10.17487/RFC7348, August 2014, <<https://www.rfc-editor.org/info/rfc7348>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.

Authors' Addresses

Santosh Pallagatti (editor)
Independent Contributor

Email: santosh.pallagatti@gmail.com

Sudarsan Paragiri
Juniper Networks
1194 N. Mathilda Ave.
Sunnyvale, California 94089-1206
USA

Email: sparagiri@juniper.net

Vengada Prasad Govindan
Cisco

Email: venggovi@cisco.com

Mallik Mudigonda
Cisco

Email: mmudigon@cisco.com

Greg Mirsky
ZTE Corp.

Email: gregimirsky@gmail.com