

Routing Working Group
Internet-Draft
Intended status: Standards Track
Expires: July 30, 2017

A. Mishra
Ciena Corporation
M. Jethanandani
Cisco Systems
A. Saxena
Ciena Corporation
S. Pallagatti
Juniper Networks
M. Chen
Huawei
P. Fan
China Mobile
January 26, 2017

BFD Stability
draft-ashesh-bfd-stability-05.txt

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 frame loss.

Requirements Language

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

Status of This Memo

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

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

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

This Internet-Draft will expire on July 30, 2017.

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 (<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	2
2. Use Cases	3
3. BFD Null-Authentication TLV	3
4. Theory of Operations	3
4.1. Loss Measurement	3
5. IANA Requirements	4
6. Security Consideration	4
7. Contributors	4
8. Acknowledgements	4
9. Normative References	4
Authors' Addresses	4

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 detect lost frames 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 BFD extension to measure data traffic loss or delay on a link or tunnel and the scope is limited to BFD frames.

2. Use Cases

Legacy BFD cannot detect any BFD frame loss if loss does not last for dead interval. This draft proposes a method to detect a dropped frame on the receiver. For example, if the receiver receives BFD CC frame k at time t but receives frame k+3 at time t+10ms, and never receives frame k+1 and/or k+2, then it has experienced a drop.

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

In a faulty datapath scenario, operator can use BFD health information to trigger delay and loss measurement OAM protocol (Connectivity Fault Management (CFM) or Loss Measurement (LM)-Delay Measurement (DM)) to further isolate the issue.

3. BFD Null-Authentication TLV

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

4. Theory of Operations

This mechanism allows operator to measure the loss of BFD CC frames.

When using MD5 or SHA authentication, BFD uses authentication TLV that carries the Sequence Number. However, if non-meticulous authentication is being used, or no authentication is in use, then the non-authenticated BFD frames MUST include NULL-Auth TLV.

4.1. Loss Measurement

Loss measurement counts the number of BFD control frames 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 CC frames. The Sequence Number in each successive control frame generated on a BFD session by the transmitter is incremented by one.

The first BFD NULL-Auth TLV processed by the receiver that has a non-zero sequence number is used for bootstrapping the logic. Each successive frame after this is expected to have a Sequence Number that is one greater than the Sequence Number in the previous frame.

When the Sequence Number wraps around it should start from 1 instead of 0.

5. IANA Requirements

N/A

6. Security Consideration

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

7. Contributors

Manav Bhatia

8. Acknowledgements

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

9. 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-02 (work in progress), January
2017.
- [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>>.
- [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>>.

Authors' Addresses

Ashesh Mishra
Ciena Corporation
3939 North 1st Street
San Jose, CA 95134
USA

Email: mishra.ashesh@outlook.com
URI: www.ciena.com

Mahesh Jethanandani
Cisco Systems
170 W. Tasman Drive
San Jose, CA 95134
USA

Email: mjethanandani@gmail.com
URI: www.cisco.com

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

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

Santosh Pallagatti
Juniper Networks
Juniper Networks, Exora Business Park
Bangalore, Karnataka 560103
India

Email: santoshpk@juniper.net

Mach Chen
Huawei

Email: mach.chen@huawei.com

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

Email: fanp08@gmail.com

Internet Engineering Task Force
Internet-Draft
Updates: 5884 (if approved)
Intended status: Standards Track
Expires: April 6, 2015

V. Govindan
Cisco Systems
K. Rajaraman
G. Mirsky
Ericsson
N. Akiya
Cisco Systems
S. Aldrin
Huawei Technologies
October 3, 2014

Clarifications to RFC 5884
draft-grmas-bfd-rfc5884-clarifications-00

Abstract

This document clarifies the procedures for establishing, maintaining and removing multiple, concurrent BFD sessions for a given <MPLS LSP, FEC> described in RFC5884.

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 April 6, 2015.

Copyright Notice

Copyright (c) 2014 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. Background	2
1.1. Requirements Language	2
2. Theory of Operation	3
2.1. Procedures for establishment of multiple BFD sessions . . .	3
2.2. Procedures for maintenance of multiple BFD sessions . . .	3
2.3. Procedures for removing BFD sessions at the egress LSR . .	4
2.4. Changing discriminators for a BFD session	4
3. Backwards Compatibility	4
4. Encapsulation	5
5. Security Considerations	5
6. IANA Considerations	5
7. Acknowledgements	5
8. Normative References	5
Authors' Addresses	5

1. Background

[RFC5884] defines the procedures to bootstrap and maintain BFD sessions for a <MPLS FEC, LSP> using LSP ping. While Section 4 of [RFC5884] specifies that multiple BFD sessions can be established for a <MPLS FEC, LSP> tuple, the procedures to bootstrap and maintain multiple BFD sessions concurrently over a <MPLS FEC, LSP> are not clearly specified. Additionally, the procedures of removing BFD sessions bootstrapped on the egress LSR are unclear. This document provides those clarifications without deviating from the principles outlined in [RFC5884].

The ability for an ingress LSR to establish multiple BFD sessions for a <MPLS FEC, LSP> tuple is useful in scenarios such as Segment Routing based LSPs or LSPs having Equal-Cost Multipath (ECMP). The process used by the ingress LSR to determine the number of BFD session(s) to be bootstrapped for a <MPLS FEC, LSP> tuple and the mechanism of constructing those session(s) are outside the scope of this document.

1.1. 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 [RFC2119].

2. Theory of Operation

2.1. Procedures for establishment of multiple BFD sessions

Section 6 of [RFC5884] specifies the procedure for bootstrapping BFD sessions using LSP ping. It further states that a BFD session SHOULD be established for each alternate path that is discovered. This requirement has been the source of some ambiguity as the procedures of establishing concurrent, multiple sessions have not been explicitly specified. This ambiguity can also be attributed in part to the text in Section 7 of [RFC5884] forbidding either end to change local discriminator values in BFD control packets after the session reaches the UP state. The following procedures are described to clarify the ambiguity based on the interpretation of the authors's reading of the referenced sections:

At the ingress LSR:

MPLS LSP ping can be used to bootstrap multiple BFD sessions for a given <MPLS FEC, LSP>. Each LSP ping MUST carry a different discriminator value in the BFD discriminator TLV [RFC4379].

The egress LSR needs to perform the following:

If the validation of the FEC in the MPLS Echo request message succeeds, check the discriminator specified in the BFD discriminator TLV of the MPLS Echo request. If there is no local session that corresponds to the discriminator (remote) received in the MPLS Echo request, a new session is bootstrapped and a local discriminator is allocated. Since the BFD local discriminator of either ends cannot change as long as the session is in the UP state, a new discriminator received in the LSP ping unambiguously conveys the intent of the LSR ingress to bootstrap a new BFD session for the FEC specified in the LSP ping.

Ensure the uniqueness of the <MPLS FEC, LSP, Remote Discriminator> tuple.

The remaining procedures of session establishment are as specified in [RFC5884].

2.2. Procedures for maintenance of multiple BFD sessions

Both the ingress LSR and egress LSR use the YourDiscriminator of the received BFD packet to demultiplex BFD sessions.

2.3. Procedures for removing BFD sessions at the egress LSR

[RFC5884] does not specify an explicit procedure for deleting BFD sessions. The procedure for removing a BFD session established by an out-of-band discriminator exchange using the MPLS LSP ping can improve resource management (like memory etc.) especially in scenarios involving thousands or more of such sessions. A few options are possible here:

The BFD session MAY be removed in the egress LSR if the BFD session transitions from UP to DOWN. This can be done after the expiry of a configurable timer started after the BFD session state transitions from UP to DOWN at the egress LSR.

The BFD session on the egress LSR MAY be gracefully removed by the ingress LSR by using the BFD diagnostic code AdminDown(7) specified in [RFC5880]. When the ingress LSR wants to gracefully remove a session, it MAY transmit BFD packets containing the diagnostic code AdminDown(7) detectMultiplier number of times. Upon receiving such a packet, the egress LSR MAY remove the BFD session gracefully, without triggering a change of state.

Ed Note: The procedures to be followed at the egress LSR when the BFD session never transitions to UP from DOWN state are yet to be clarified

Regardless of the option chosen to proceed, all BFD sessions established with the FEC MUST be removed automatically if the FEC is removed.

2.4. Changing discriminators for a BFD session

The discriminators of a BFD session established over an MPLS LSP cannot be changed when it is in UP state. The BFD session could be removed after a graceful transition to AdminDown state using the BFD diagnostic code AdminDown. A new session could be established with a different discriminator. The initiation of the transition from the Up to Down state can be done either by the ingress LSR or the egress LSR.

3. Backwards Compatibility

The procedures clarified by this document are fully backward compatible with an existing implementation of [RFC5884]. While the capability to bootstrap and maintain multiple BFD sessions may not be present in current implementations, the procedures outlined by this document can be implemented as a software upgrade without affecting existing sessions. In particular, the egress LSR needs to support

multiple BFD sessions per <MPLS FEC, LSP> before the ingress LSR is upgraded.

4. Encapsulation

The encapsulation of BFD packets are the same as specified by [RFC5884].

5. Security Considerations

This document clarifies the mechanism to bootstrap multiple BFD sessions per <MPLS FEC, LSP>. BFD sessions, naturally, use system and network resources. More BFD sessions means more resources will be used. It is highly important to ensure only minimum number of BFD sessions are provisioned per FEC, and bootstrapped BFD sessions are properly deleted when no longer required. Additionally security measures described in [RFC4379] and [RFC5884] are to be followed.

6. IANA Considerations

This document does not make any requests to IANA.

7. Acknowledgements

The authors would like to thank Mudigonda Mallik, Rajaguru Veluchamy and Carlos Pignataro of Cisco Systems for their review comments.

8. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC4379] Kompella, K. and G. Swallow, "Detecting Multi-Protocol Label Switched (MPLS) Data Plane Failures", RFC 4379, February 2006.
- [RFC5880] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD)", RFC 5880, June 2010.
- [RFC5884] Aggarwal, R., Kompella, K., Nadeau, T., and G. Swallow, "Bidirectional Forwarding Detection (BFD) for MPLS Label Switched Paths (LSPs)", RFC 5884, June 2010.

Authors' Addresses

Vengada Prasad Govindan
Cisco Systems

Email: venggovi@cisco.com

Kalyani Rajaraman
Ericsson

Email: kalyani.rajaraman@ericsson.com

Gregory Mirsky
Ericsson

Email: gregory.mirsky@ericsson.com

Nobo Akiya
Cisco Systems

Email: nobo@cisco.com

Sam Aldrin
Huawei Technologies

Email: aldrin.ietf@gmail.com

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`.

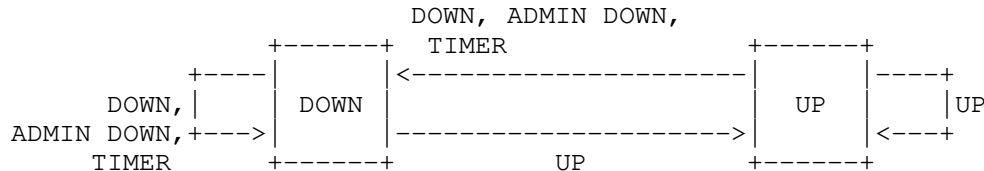
`bfd.DemandMode`

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

MPLS Working Group
Internet-Draft
Intended status: Standards Track
Expires: February 9, 2016

G. Mirsky
J. Tantsura
Ericsson
I. Varlashkin
Google
M. Chen
Huawei
August 8, 2015

Bidirectional Forwarding Detection (BFD) Directed Return Path
draft-mirsky-mpls-bfd-directed-04

Abstract

Bidirectional Forwarding Detection (BFD) is expected to monitor bi-directional paths. When a BFD session monitors in its forward direction an explicitly routed path there is a need to be able to direct egress BFD peer to use specific path as reverse direction of the BFD session.

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 February 9, 2016.

Copyright Notice

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

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

to this document. Code Components extracted from this document must 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. Direct Reverse BFD Path	4
3.1. Case of MPLS Data Plane	4
3.1.1. BFD Reverse Path TLV	4
3.1.2. Static and RSVP-TE sub-TLVs	5
3.1.3. Segment Routing Tunnel sub-TLV	5
3.2. Case of IPv6 Data Plane	6
3.3. Bootstrapping BFD session with BFD Reverse Path over Segment Routed tunnel	6
3.4. Return Codes	7
4. Use Case Scenario	7
5. IANA Considerations	7
5.1. TLV	8
5.2. Sub-TLV	8
5.3. Return Codes	8
6. Security Considerations	9
7. Acknowledgements	9
8. Normative References	9
Authors' Addresses	10

1. Introduction

RFC 5880 [RFC5880], RFC 5881 [RFC5881], and RFC 5883 [RFC5883] established the BFD protocol for IP networks and RFC 5884 [RFC5884] set rules of using BFD asynchronous mode over IP/MPLS LSPs. All standards implicitly assume that the egress BFD peer will use the shortest path route regardless of route being used to send BFD control packets towards it. As result, if the ingress BFD peer sends its BFD control packets over explicit path that is diverging from the best route, then reverse direction of the BFD session is likely not to be on co-routed bi-directional path with the forward direction of the BFD session. And because BFD control packets are not guaranteed to cross the same links and nodes in both directions detection of Loss of Continuity (LoC) defect in forward direction may demonstrate positive negatives.

This document defines the extension to LSP Ping [RFC4379], BFD Reverse Path TLV, and proposes that it to be used to instruct the egress BFD peer to use explicit path for its BFD control packets associated with the particular BFD session. The TLV will be allocated from the TLV and sub-TLV registry defined by RFC 4379 [RFC4379]. As a special case, forward and reverse directions of the BFD session can form bi-directional co-routed associated channel.

1.1. Conventions used in this document

1.1.1. Terminology

BFD: Bidirectional Forwarding Detection

MPLS: Multiprotocol Label Switching

LSP: Label Switching Path

LoC: Loss of Continuity

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 [RFC2119].

2. Problem Statement

BFD is best suited to monitor bi-directional co-routed paths. In most cases, given stable environments, the forward and reverse direction between two nodes is likely to be co-routed, this fulfilling the implicit BFD requirements. If BFD is used to monitor unidirectional explicitly routed paths, e.g. MPLS-TE LSPs, its control packets in forward direction would be in-band using the mechanism defined in [RFC5884] and [RFC5586]. But the reverse direction of the BFD session would still follow the shortest path route and that might lead to the following problems detecting failures on the unidirectional explicit path:

- o detection of a failure on the reverse path cannot reliably be interpreted as bi-directional defect and thus trigger, for example, protection switchover of the forward direction;
- o if a failure of the reverse path had been ignored, the ingress node would not receive indication of forward direction failure from its egress peer.

To address these challenges the egress BFD peer should be instructed to use specific path for its control packets.

3. Direct Reverse BFD Path

3.1. Case of MPLS Data Plane

LSP ping, defined in [RFC4379], uses BFD Discriminator TLV [RFC5884] to bootstrap a BFD session over an MPLS LSP. This document defines a new TLV, BFD Reverse Path TLV, that MUST contain a single sub-TLV that can be used to carry information about reverse path for the specified in BFD Discriminator TLV session.

3.1.1. BFD Reverse Path TLV

The BFD Reverse Path TLV is an optional TLV within the LSP ping protocol. However, if used, the BFD Discriminator TLV MUST be included in an Echo Request message as well. If the BFD Discriminator TLV is not present when the BFD Reverse Path TLV is included, then it MUST be treated as malformed Echo Request, as described in [RFC4379].

The BFD Reverse Path TLV carries the specified path that BFD control packets of the BFD session referenced in the BFD Discriminator TLV are required to follow. The format of the BFD Reverse Path TLV is as presented in Figure 1.

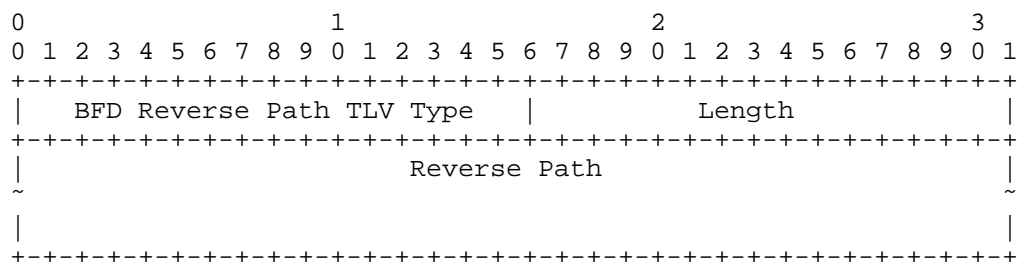


Figure 1: BFD Reverse Path TLV

BFD Reverse Path TLV Type is 2 octets in length and value to be assigned by IANA.

Length is 2 octets in length and defines the length in octets of the Reverse Path field.

Reverse Path field contains a sub-TLV. Any Target FEC sub-TLV, already or in the future defined, from IANA sub-registry Sub-TLVs for TLV Types 1, 16, and 21 of MPLS LSP Ping Parameters registry MAY be

used in this field. Only one sub-TLV MUST be included in the Reverse Path TLV. If more than one sub-TLVs are present in the Reverse Path TLV, then only the first sub-TLV MUST be used and the rest MUST be silently discarded.

If the egress LSR cannot find path specified in the Reverse Path TLV it MUST send Echo Reply with the received Reverse Path TLV and set the return code to "Failed to establish the BFD session. The specified reverse path was not found" Section 3.4. The egress LSR MAY establish the BFD session over IP network according to [RFC5884].

3.1.2. Static and RSVP-TE sub-TLVs

When explicit path on MPLS data plane set either as Static or RSVP-TE LSP respective sub-TLVs defined in [RFC7110] identify explicit return path.

3.1.3. Segment Routing Tunnel sub-TLV

In addition to Static and RSVP-TE, Segment Routing with MPLS data plane can be used to set explicit path. In this case a new sub-TLV is defined in this document as presented in Figure 2.

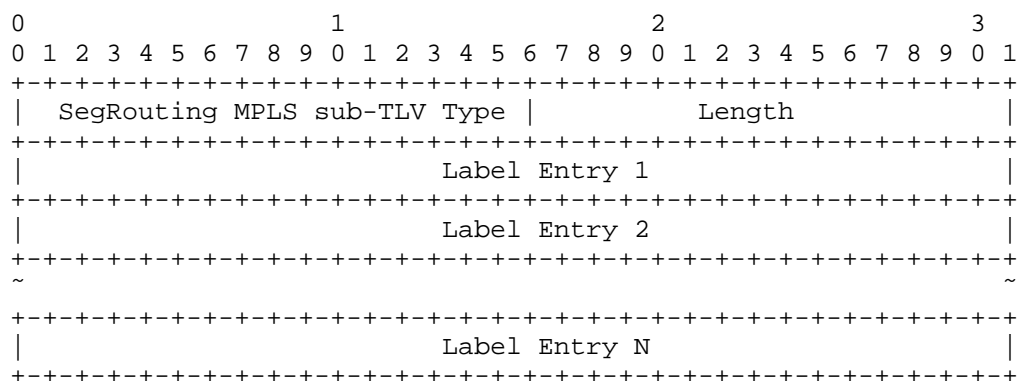


Figure 2: Segment Routing MPLS Tunnel sub-TLV

The Segment Routing Tunnel sub-TLV Type is two octets in length, and will be allocated by IANA.

The egress LSR MUST use the Value field as label stack for BFD control packets for the BFD session identified by source IP address and value in BFD Discriminator TLV.

The Segment Routing Tunnel sub-TLV MAY be used in Reply Path TLV defined in [RFC7110]

3.2. Case of IPv6 Data Plane

IPv6 can be data plane of choice for Segment Routed tunnels [I-D.previdi-6man-segment-routing-header]. In such networks the BFD Reverse Path TLV described in Section 3.1.1 can be used as well. To specify reverse path of a BFD session in IPv6 environment the BFD Discriminator TLV MUST be used along with the BFD Reverse Path TLV. The BFD Reverse Path TLV in IPv6 network MUST include sub-TLV.

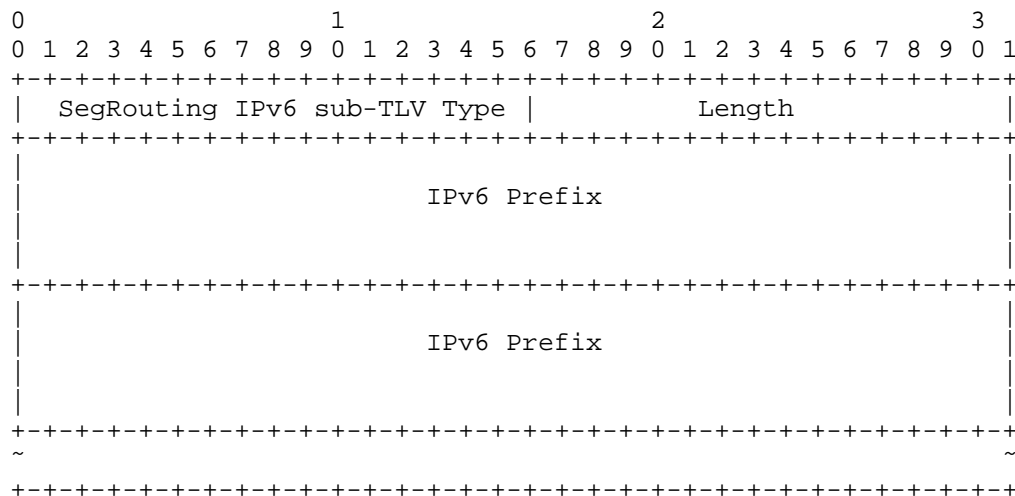


Figure 3: Segment Routing IPv6 Tunnel sub-TLV

3.3. Bootstrapping BFD session with BFD Reverse Path over Segment Routed tunnel

As discussed in [I-D.kumarkini-mpls-spring-lsp-ping] introduction of Segment Routing network domains with MPLS data plane adds three new sub-TLVs that may be used with Target FEC TLV. Section 6.1 addresses use of new sub-TLVs in Target FEC TLV in LSP ping and LSP traceroute. For the case of LSP ping the [I-D.kumarkini-mpls-spring-lsp-ping] states that:

"Initiator MUST include FEC(s) corresponding to the destination segment.

Initiator, i.e. ingress LSR, MAY include FECs corresponding to some or all of segments imposed in the label stack by the ingress LSR to communicate the segments traversed. "

When LSP ping is used to bootstrap BFD session this document updates this and defines that LSP Ping MUST include the FEC corresponding to the destination segment and SHOULD NOT include FECs corresponding to some or all of segment imposed by the ingress LSR. Operationally such restriction would not cause any problem or uncertainty as LSP ping with FECs corresponding to some or all segments or traceroute MAY precede the LSP ping that bootstraps the BFD session.

3.4. Return Codes

This document defines the following Return Codes:

- o "Failed to establish the BFD session. The specified reverse path was not found", (TBD4). When a specified reverse path is not available at the egress LSR, an Echo Reply with the return code set to "Failed to establish the BFD session. The specified reverse path was not found" MUST be sent back to the ingress LSR . (Section 3.1.1)

4. Use Case Scenario

In network presented in Figure 4 node A monitors two tunnels to node H: A-B-C-D-G-H and A-B-E-F-G-H. To bootstrap BFD session to monitor the first tunnel, node A MUST include BFD Discriminator TLV with Discriminator value foobar-1 and MAY include BFD Reverse Path TLV that references H-G-D-C-B-A tunnel. To bootstrap BFD session to monitor the second tunnel, node A MUST include BFD Discriminator TLV with Discriminator value foobar-2 [I-D.ietf-bfd-rfc5884-clarifications] and MAY include BFD Reverse Path TLV that references H-G-F-E-B-A tunnel.

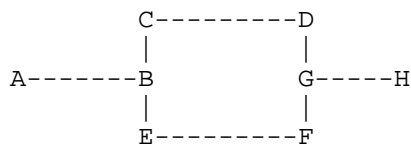


Figure 4: Use Case for BFD Reverse Path TLV

If an operator needs node H to monitor path to node A, e.g. H-G-D-C-B-A tunnel, then by looking up list of known Reverse Paths it MAY find and use existing BFD sessions.

5. IANA Considerations

5.1. TLV

The IANA is requested to assign a new value for BFD Reverse Path TLV from the "Multiprotocol Label Switching Architecture (MPLS) Label Switched Paths (LSPs) Ping Parameters - TLVs" registry, "TLVs and sub-TLVs" sub-registry.

Value	Description	Reference
X (TBD1)	BFD Reverse Path TLV	This document

Table 1: New BFD Reverse Type TLV

5.2. Sub-TLV

The IANA is requested to assign two new sub-TLV types from "Multiprotocol Label Switching Architecture (MPLS) Label Switched Paths (LSPs) Ping Parameters - TLVs" registry, "Sub-TLVs for TLV Types 1, 16, and 21" sub-registry.

Value	Description	Reference
X (TBD2)	Segment Routing MPLS Tunnel sub-TLV	This document
X (TBD3)	Segment Routing IPv6 Tunnel sub-TLV	This document

Table 2: New Segment Routing Tunnel sub-TLV

5.3. Return Codes

The IANA is requested to 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 (TBD4)	Failed to establish the BFD session. The specified reverse path was not found.	This document

Table 3: New Return Code

6. Security Considerations

Security considerations discussed in [RFC5880], [RFC5884], and [RFC4379], apply to this document.

7. Acknowledgements

8. Normative References

- [I-D.ietf-bfd-rfc5884-clarifications]
Govindan, V., Rajaraman, K., Mirsky, G., Akiya, N., and S. Aldrin, "Clarifications to RFC 5884", draft-ietf-bfd-rfc5884-clarifications-02 (work in progress), June 2015.
- [I-D.kumarkini-mpls-spring-lsp-ping]
Kumar, N., Swallow, G., Pignataro, C., Akiya, N., Kini, S., Gredler, H., and M. Chen, "Label Switched Path (LSP) Ping/Trace for Segment Routing Networks Using MPLS Dataplane", draft-kumarkini-mpls-spring-lsp-ping-04 (work in progress), July 2015.
- [I-D.previdi-6man-segment-routing-header]
Previdi, S., Filsfils, C., Field, B., Leung, I., Vyncke, E., and D. Lebrun, "IPv6 Segment Routing Header (SRH)", draft-previdi-6man-segment-routing-header-07 (work in progress), July 2015.
- [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>>.
- [RFC4379] Kompella, K. and G. Swallow, "Detecting Multi-Protocol Label Switched (MPLS) Data Plane Failures", RFC 4379, DOI 10.17487/RFC4379, February 2006, <<http://www.rfc-editor.org/info/rfc4379>>.
- [RFC5586] Bocci, M., Ed., Vigoureux, M., Ed., and S. Bryant, Ed., "MPLS Generic Associated Channel", RFC 5586, DOI 10.17487/RFC5586, June 2009, <<http://www.rfc-editor.org/info/rfc5586>>.
- [RFC5880] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD)", RFC 5880, DOI 10.17487/RFC5880, June 2010, <<http://www.rfc-editor.org/info/rfc5880>>.

- [RFC5881] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD) for IPv4 and IPv6 (Single Hop)", RFC 5881, DOI 10.17487/RFC5881, June 2010, <<http://www.rfc-editor.org/info/rfc5881>>.
- [RFC5883] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD) for Multihop Paths", RFC 5883, DOI 10.17487/RFC5883, June 2010, <<http://www.rfc-editor.org/info/rfc5883>>.
- [RFC5884] Aggarwal, R., Kompella, K., Nadeau, T., and G. Swallow, "Bidirectional Forwarding Detection (BFD) for MPLS Label Switched Paths (LSPs)", RFC 5884, DOI 10.17487/RFC5884, June 2010, <<http://www.rfc-editor.org/info/rfc5884>>.
- [RFC7110] Chen, M., Cao, W., Ning, S., Jounay, F., and S. Delord, "Return Path Specified Label Switched Path (LSP) Ping", RFC 7110, DOI 10.17487/RFC7110, January 2014, <<http://www.rfc-editor.org/info/rfc7110>>.

Authors' Addresses

Greg Mirsky
Ericsson

Email: gregory.mirsky@ericsson.com

Jeff Tantsura
Ericsson

Email: jeff.tantsura@ericsson.com

Ilya Varlashkin
Google

Email: Ilya@nobulus.com

Mach(Guoyi) Chen
Huawei

Email: mach.chen@huawei.com