

Internet Engineering Task Force
Internet-Draft
Intended status: Standards Track
Expires: December 28, 2014

N. Akiya
C. Pignataro
D. Ward
Cisco Systems
June 26, 2014

Seamless Bidirectional Forwarding Detection (S-BFD) for
IPv4, IPv6 and MPLS
draft-akiya-bfd-seamless-ip-03

Abstract

This document defines procedures to use Seamless Bidirectional Forwarding Detection (S-BFD) for IPv4, IPv6 and MPLS environments.

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 December 28, 2014.

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. Introduction	2
2. Initiator Procedures	2
2.1. Details of S-BFD Packet Sent by SBFDInitiator	3
2.2. Target vs. Remote Entity (S-BFD Discriminator)	3
3. Responder Procedures	4
3.1. Details of S-BFD Packet Sent by SBFDReflector	4
4. Security Considerations	4
5. IANA Considerations	4
6. Acknowledgements	4
7. Contributing Authors	4
8. Normative References	5
Authors' Addresses	5

1. Introduction

Seamless Bidirectional Forwarding Detection (S-BFD), [I-D.ietf-bfd-seamless-base], defines a generalized mechanism to allow network nodes to seamlessly perform connectivity checks to remote entities. This document defines necessary procedures to use S-BFD on IPv4, IPv6 and MPLS environments.

The reader is expected to be familiar with the IP, MPLS BFD and S-BFD terminologies and protocol constructs.

2. Initiator Procedures

S-BFD packets are transmitted with IP header, UDP header and BFD control header ([RFC5880]). When S-BFD packets are explicitly label switched, the former is prepended with a label stack. Note that this document does not make a distinction between a single-hop S-BFD scenario and a multi-hop S-BFD scenario, both scenarios are supported.

Necessary values in the UDP and BFD control headers are described in [I-D.ietf-bfd-seamless-base]. Section 2.1 describes necessary values in the IP and MPLS headers when an SBFDInitiator on the initiator is sending S-BFD packets.

2.1. Details of S-BFD Packet Sent by SBFDInitiator

- o Specification common to both IP routed S-BFD packets and explicitly label switched S-BFD packets:
 - * Source IP address field of the IP header MUST be set to a local IP address.
- o Specification for IP routed S-BFD packets:
 - * Destination IP address field of the IP header MUST set to an IP address of the target.
 - * TTL field of the IP header SHOULD be set to 255.
- o Specification for explicitly label switched S-BFD packets:
 - * S-BFD packets MUST have the label stack that is expected to reach the target.
 - * TTL field of the top most label SHOULD be 255.
 - * Destination IP address field of the IP header MUST be set to 127/8 for IPv4 and 0:0:0:0:0:FFFF:7F00/104 for IPv6.
 - * TTL field of the IP header MUST be set to 1.

Ed-Note: Discuss whether we want a new associated channel type for S-BFD.

2.2. Target vs. Remote Entity (S-BFD Discriminator)

Typically, an S-BFD packet will have "your discriminator" field corresponding to an S-BFD discriminator of the remote entity located on the target network node defined by the destination IP address or the label stack. It is, however, possible for an SBFDInitiator to carefully set "your discriminator" and TTL fields to perform a connectivity test towards a target but to a transit network node.

Section 2.1 intentionally uses the word "target", instead of "remote entity", to accommodate this possible S-BFD usage through TTL expiry. This also requires S-BFD packets not be dropped by the responder node due to TTL expiry. Thus implementations on the responder MUST allow received S-BFD packets taking TTL expiry exception path to reach corresponding reflector BFD session.

3. Responder Procedures

S-BFD packets are IP routed back to the initiator, and will have IP header, UDP header and BFD control header. Necessary values in the UDP and BFD control headers are described in [I-D.ietf-bfd-seamless-base]. Section 3.1 describes necessary values in the IP header when an SBFDRreflector on the responder is sending S-BFD packets.

3.1. Details of S-BFD Packet Sent by SBFDRreflector

- o Destination IP address field of the IP header MUST be copied from source IP address field of received S-BFD packet.
- o Source IP address field of the IP header MUST be set to a local IP address.
- o TTL field of the IP header SHOULD be set to 255.

4. Security Considerations

Security considerations for S-BFD are discussed in [I-D.ietf-bfd-seamless-base].

5. IANA Considerations

No action is required by IANA for this document.

6. Acknowledgements

Authors would like to thank Marc Binderberger from Cisco Systems for providing valuable comments.

7. Contributing Authors

Tarek Saad
Cisco Systems
Email: tsaad@cisco.com

Siva Sivabalan
Cisco Systems
Email: msiva@cisco.com

Nagendra Kumar
Cisco Systems
Email: naikumar@cisco.com

8. Normative References

[I-D.ietf-bfd-seamless-base]

Akiya, N., Pignataro, C., Ward, D., Bhatia, M., and J. Networks, "Seamless Bidirectional Forwarding Detection (S-BFD)", draft-ietf-bfd-seamless-base-00 (work in progress), June 2014.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC5880] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD)", RFC 5880, June 2010.

Authors' Addresses

Nobo Akiya
Cisco Systems

Email: nobo@cisco.com

Carlos Pignataro
Cisco Systems

Email: cpignata@cisco.com

Dave Ward
Cisco Systems

Email: wardd@cisco.com

Routing Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 1, 2015

A. Mishra
Ciena Corporation
S. Pallagatti
Juniper Networks
M. Jethanandani
Ciena Corporation
M. Chen
Huawei
A. Saxena
Ciena Corporation
June 30, 2014

BFD Stability
draft-ashesh-bfd-stability-00.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, of delays in frame transmitter and receiver engines, and of inter-frame delays that might explain issues with a BFD session.

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 <xref target="RFC2119">RFC 2119</xref>.

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 January 1, 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. Introduction	2
2. BFD Null-Authentication TLV	3
3. Theory of Operations	4
3.1. Frame Loss	4
3.2. Inter-Frame Gap	5
3.3. Frame Transmission Delay	5
4. IANA Requirements	5
5. Security Consideration	6
6. Acknowledgements	6
7. Normative References	6
Authors' Addresses	6

1. Introduction

The Bidirectional Forwarding Detection (BFD) 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 (the Detection Time as described in RFC 5880) is set to the smallest feasible value. In certain cases, this Detection Time is comparable to the inter-frame delays caused by random network events such as frame drops or frame processing (transmitter or receiver) delays.

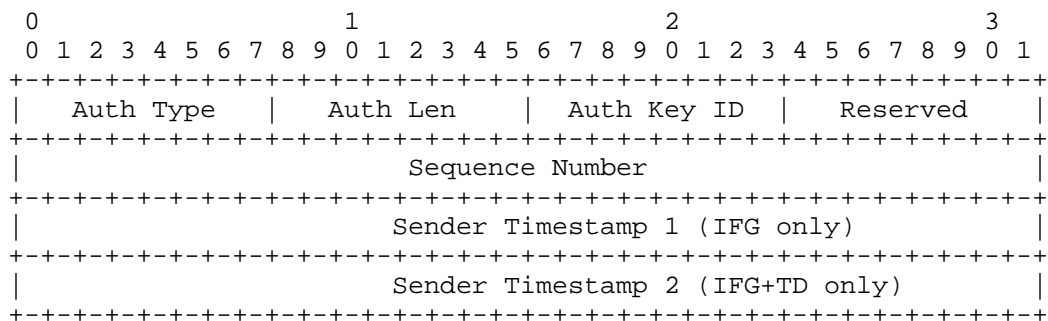
This document proposes a mechanism to measure such transient effects to detect instability in the receive direction of the data path from the session peer in addition to the datapath fault detection mechanisms of BFD. Such a mechanism presents significant value with the ability to measure the stability of BFD sessions and provides data to the operators.

In addition to stability measurement, the information exchanged between BFD peers can be used for rudimentary, but low-overhead, authentication.

2. BFD Null-Authentication TLV

The functionality proposed for BFD stability measurement is achieved by appending the Null-Authentication TLV to the BFD control frame.

The Null-Authentication TLV (called 0-Auth in this document) extends the existing BFD Authentication TLV structure by adding a new Auth-Type of <IANA Assigned>. This TLV carries the Sequence Number for frame loss measurement, and Sender Timestamps for delay measurements.



where:

Auth Type: The Authentication Type, which in this case is <IANA assigned> (Null Authentication).

Auth Len: The length of the Authentication Section, in bytes. For Loss Measurement only, the length is set to 4. For Loss and Inter-Frame Gap measurements, the length is set to 8. For Loss, Inter-Frame Gap and Transmission Delay on sender node, the length is set to 12.

Auth Key ID: The Authentication Key ID in use for this packet. This MUST be set to zero on transmit, and ignored on receipt.

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

Sequence Number: This indicates the sequence number for this packet and MUST be present in every 0-Auth TLV. This value is incremented by 1 for every frame transmitted while the session state is UP. A value of 0 indicates a request by sender to reset the sequence number

correlation logic at the receiver. The first frame transmitted by the sender MAY set this field to 0.

Inter-Frame Gap (IFG) Mode:

Sender Timestamp 1 (IFG-ST): This is the Inter-Frame Gap Sender Timestamp (IFG-ST) and is added at the last possible instance on the sender (preferably on the PHY). The difference between two such timestamps on consecutive frames is the Inter-Frame gap.

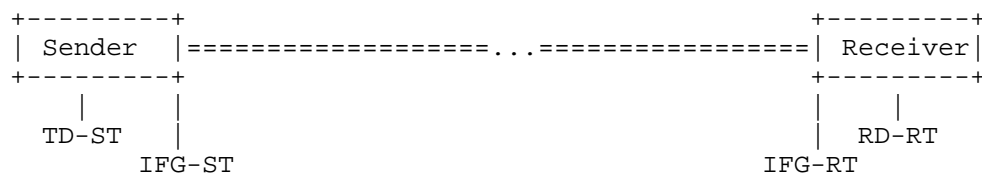
Inter-Frame Gap and Transmission Delay (IFG & TD) Mode:

Sender Timestamp 1 (TD-ST): This is the Transmission Delay Sender Timestamp (TD-ST) and is added at the first possible instance on the sender in the frame transmission engine. The difference between TD-ST and the IFG-ST that follows the TD-ST is the Sender Transmission Delay.

Sender Timestamp 2 (IFG-ST): This is the Inter-Frame Gap Sender Timestamp (IFG-ST) and is added at the last possible instance on the sender (preferably on the PHY). The difference between two such timestamps on consecutive frames is the Inter-Frame gap.

3. Theory of Operations

This mechanism allows operator to read three measures of stability of BFD: Frame Loss, Inter-Frame Gap and Transmission Delay. The Receiver Delay (interval between receipt of a frame on the PHY and the completion of processing in the receiver engine) can be measured using timestamps similar to the Sender Timestamps on the receiver node.



3.1. Frame Loss

This measurement counts the number of BFD control frames missed at the receiver due to a transient change in the network such as congestion. Frame-loss is detected by comparing the Sequence Number field in the 0-Auth TLV 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 Loss-Delay 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.

3.2. Inter-Frame Gap

This measurement is the difference between the IFG-ST on any two consecutive BFD CC frames that carry the 0-Auth TLV (IFG or IFG&TD mode only) for a session. This is a key metric to determine transient changes in stability of BFD transmission engine or to determine the systems capability of handling the existing load. A significant deviation of IFG from the negotiated transmission interval on the local node indicates potential instabilities in the BFD transmission engine. Based on the IFG measurements, the operator MAY take action to configure the system to maintain normal operation of the node.

Similar IFG measurements on the receiver can be made using timestamps (IFG-RT). In conjunction with IFG-ST measurements, these can indicate delays caused by data-path. While a constant delay may not be indicator of instability, large transient delays can decrease the BFD session stability significantly.

3.3. Frame Transmission Delay

This measurement (TD) is the interval between the timestamp (TD-ST) when the frame transmission timer expires, triggering the BFD control frame generation, and the timestamp (IFG-TD) when the frame reaches the last level in the frame processing logic on the transmitter where the frame can be manipulated. Large variations in the TD measurements over time are indicative of non-deterministic transmission behavior of the BFD engine and can be a pre-cursor to BFD engine instability.

Similar measurements for Receiver Delay (RD) can be made using IFG-RT and RD-RT timestamps, and indicate similar instabilities on the BFD receiver engine.

4. IANA Requirements

IANA is requested to assign new Auth-Type for the Null-Authentication TLV for BFD Stability Measurement. The following number is suggested.

Value Meaning

6 Null-Authentication TLV

5. Security Consideration

Since this method uses an authentication TLV to achieve the functionality, usage of this TLV will prevent the use of other authentication TLVs.

6. Acknowledgements

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

7. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC5880] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD)", RFC 5880, June 2010.

Authors' Addresses

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

Email: mishra.ashesh@gmail.com

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

Phone: +
Email: santoshpk@juniper.net

Mahesh Jethanandani
Ciena Corporation
3939 North 1st Street
San Jose, CA 95134
USA

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

Mach Chen
Huawei

Email: mach.chen@huawei.com

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

Email: ankurpsaxena@gmail.com

Internet Engineering Task Force
Internet-Draft
Updates: 5880 (if approved)
Intended status: Standards Track
Expires: December 28, 2014

N. Akiya
C. Pignataro
D. Ward
Cisco Systems
M. Bhatia
Ionos Networks
P. K. Santosh
Juniper Networks
June 26, 2014

Seamless Bidirectional Forwarding Detection (S-BFD)
draft-ietf-bfd-seamless-base-01

Abstract

This document defines a simplified mechanism to use Bidirectional Forwarding Detection (BFD) with large portions of negotiation aspects eliminated, thus providing benefits such as quick provisioning as well as improved control and flexibility to network nodes initiating the path monitoring.

This document updates RFC5880.

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 December 28, 2014.

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. Introduction	3
2. Terminology	3
3. Seamless BFD Overview	4
4. S-BFD UDP Port	5
5. S-BFD Discriminators	5
6. Reflector BFD Session	6
7. State Variables	7
7.1. New State Variables	7
7.2. State Variable Initialization and Maintenance	7
8. S-BFD Procedures	7
8.1. Initiator Procedures	7
8.1.1. SBFDInitiator State Machine	8
8.1.2. Details of S-BFD Packet Sent by SBFDInitiator	9
8.2. Responder Procedures	9
8.2.1. Responder Demultiplexing	10
8.2.2. Details of S-BFD Packet Sent by SBFDReflector	10
8.3. Diagnostic Values	10
8.4. The Poll Sequence	11
8.5. Control Plane Independent (C)	11
8.6. Additional SBFDInitiator Behaviors	11
8.7. Additional SBFDReflector Behaviors	12
9. Scaling Aspect	12
10. Co-existence with Traditional BFD	12
11. BFD Echo	12
12. Security Considerations	13
13. IANA Considerations	14
14. Acknowledgements	14
15. Contributing Authors	14
16. References	15
16.1. Normative References	15
16.2. Informative References	15

Appendix A. Loop Problem	16
Authors' Addresses	17

1. Introduction

Bidirectional Forwarding Detection (BFD), [RFC5880] and related documents, has efficiently generalized the failure detection mechanism for multiple protocols and applications. There are some improvements which can be made to better fit existing technologies. There is a possibility of evolving BFD to better fit new technologies. This document focuses on several aspects of BFD in order to further improve efficiency, to expand failure detection coverage and to allow BFD usage for wider scenarios. This document extends BFD to provide solutions to use cases listed in [I-D.ietf-bfd-seamless-use-case].

One key aspect of the mechanism described in this document eliminates the time between a network node wanting to perform a connectivity test and completing the connectivity test. In traditional BFD terms, the initial state changes from DOWN to UP is virtually nonexistent. Removal of this seam (i.e. time delay) in BFD provides applications a smooth and continuous operational experience. Therefore, "Seamless BFD" (S-BFD) has been chosen as the name for this mechanism.

2. Terminology

The reader is expected to be familiar with the BFD, IP and MPLS terminologies and protocol constructs. This section describes several new terminologies introduced by S-BFD.

- o S-BFD - Seamless BFD.
- o S-BFD packet - a BFD control packet on the well-known S-BFD port.
- o Entity - a function on a network node that S-BFD mechanism allows remote network nodes to perform connectivity test to. An entity can be abstract (ex: reachability) or specific (ex: IP addresses, router-IDs, functions).
- o SBFDInitiator - an S-BFD session on a network node that performs a connectivity test to a remote entity by sending S-BFD packets.
- o SBFDReflector - an S-BFD session on a network node that listens for incoming S-BFD packets to local entities and generates response S-BFD packets.
- o Reflector BFD session - synonymous with SBFDReflector.

- o S-BFD discriminator - a BFD discriminator allocated for a local entity and is being listened by an SBFDReflector.
- o BFD discriminator - a BFD discriminator allocated for an SBFDInitiator.
- o Initiator - a network node hosting an SBFDInitiator.
- o Responder - a network node hosting an SBFDReflector.

Below figure describes the relationship between S-BFD terminologies.

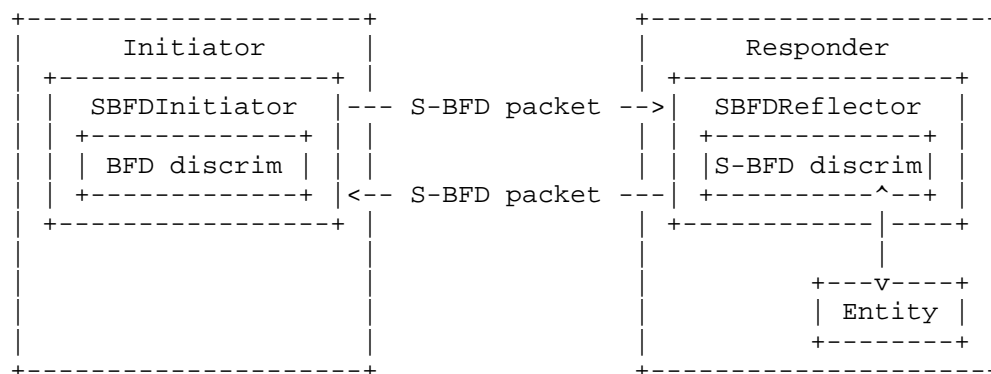


Figure 1: S-BFD Terminology Relationship

3. Seamless BFD Overview

An S-BFD module on each network node allocates one or more S-BFD discriminators for local entities, and creates a reflector BFD session. Allocated S-BFD discriminators may be advertised by applications (ex: OSPF/IS-IS). Required result is that applications, on other network nodes, possess the knowledge of the mapping from remote entities to S-BFD discriminators. The reflector BFD session is to, upon receiving an S-BFD packet targeted to one of local S-BFD discriminator values, transmit a response S-BFD packet back to the initiator.

Once above setup is complete, any network nodes, having the knowledge of the mapping from a remote entity to an S-BFD discriminator, can quickly perform a connectivity test to the remote entity by simply sending S-BFD packets with corresponding S-BFD discriminator value in the "your discriminator" field.

For example:

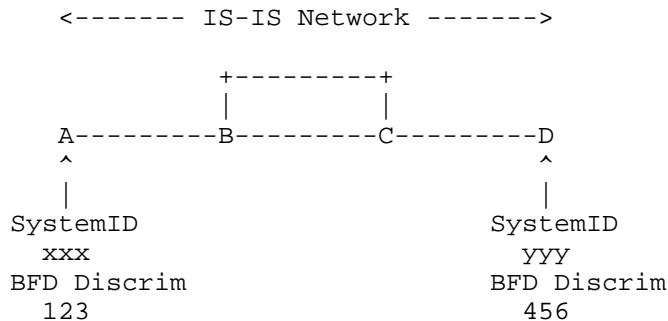


Figure 2: S-BFD for IS-IS Network

The IS-IS with SystemID xxx (node A) allocates an S-BFD discriminator 123, and advertises the S-BFD discriminator 123 in an IS-IS TLV. The IS-IS with SystemID yyy (node D) allocates an S-BFD discriminator 456, and advertises the S-BFD discriminator 456 in an IS-IS TLV. A reflector BFD session is created on both network nodes (node A and node D). When network node A wants to check the connectivity to network node D, node A can send an S-BFD packet, destined to node D, with "your discriminator" field set to 456. When the reflector BFD session on node D receives this S-BFD packet, then response S-BFD packet is sent back to node A, which allows node A to complete the connectivity test.

4. S-BFD UDP Port

S-BFD functions on a well-known UDP port: TBD1.

5. S-BFD Discriminators

Locally allocated S-BFD discriminator values for entities may be arbitrary allocated or derived from values provided by applications. These values may be protocol IDs (ex: System-ID, Router-ID) or network targets (ex: IP address). To minimize the collision of discriminator values between BFD and S-BFD, it is RECOMMENDED that discriminator pool be separate for BFD and S-BFD. Even when employing the separate discriminator pool approach, collision is still possible between one S-BFD application to another S-BFD application, that may be using different values and algorithms to derive S-BFD discriminator values. If the two applications are using S-BFD for a same purpose (ex: network reachability), then the colliding S-BFD discriminator value can be shared. If the two applications are using S-BFD for a different purpose, then the

collision must be addressed. How such collisions are addressed is outside the scope of this document.

One important characteristics of an S-BFD discriminator is that it MUST be unique within an administrative domain. If multiple network nodes allocated a same S-BFD discriminator value, then S-BFD packets falsely terminating on a wrong network node can result in a reflector BFD session to generate a response back, due to "your discriminator" matching. This is clearly not desirable. If only IP based S-BFD is considered, then it is possible for the reflector BFD session to require demultiplexing of incoming S-BFD packets with combination of destination IP address and "your discriminator". Then S-BFD discriminator only has to be unique within a local node. However, S-BFD is a generic mechanism defined to run on wide range of environments: IP, MPLS, etc. For other transports like MPLS, because of the need to use non-routable IP destination address, it is not possible for reflector BFD session to demultiplex using IP destination address. With PHP, there may not be any incoming label stack to aid in demultiplexing either. Thus, S-BFD imposes a requirement that S-BFD discriminators MUST be unique within an administrative domain.

6. Reflector BFD Session

Each network node creates one or more reflector BFD sessions. This reflector BFD session is a session which transmits S-BFD packets in response to received S-BFD packets with "your discriminator" having S-BFD discriminators allocated for local entities. Specifically, this reflector BFD session is to have following characteristics:

- o MUST NOT transmit any S-BFD packets based on local timer expiry.
- o MUST transmit an S-BFD packet in response to a received S-BFD packet having a valid S-BFD discriminator in the "your discriminator" field, unless prohibited by local policies (ex: administrative, security, rate-limiter, etc).
- o MUST be capable of sending only two states: UP and ADMINDOWN.

One reflector BFD session may be responsible for handling received S-BFD packets targeted to all locally allocated S-BFD discriminators, or few reflector BFD sessions may each be responsible for subset of locally allocated S-BFD discriminators. This policy is a local matter, and is outside the scope of this document.

Note that incoming S-BFD packets may be IPv4, IPv6 or MPLS based. How such S-BFD packets reach an appropriate reflector BFD session is also a local matter, and is outside the scope of this document.

7. State Variables

S-BFD introduces new state variables, and modifies the usage of existing ones.

7.1. New State Variables

A new state variable is added to the base specification in support of S-BFD.

- o `bfd.SessionType`: The type of this session. Allowable values are:
 - * `SBFDInitiator` - an S-BFD session on a network node that performs a connectivity test to a target entity by sending S-BFD packets.
 - * `SBFDReflector` - an S-BFD session on a network node that listens for incoming S-BFD packets to local entities and generates response S-BFD packets.

`bfd.SessionType` variable MUST be initialized to the appropriate type when an S-BFD session is created.

7.2. State Variable Initialization and Maintenance

Some state variables defined in section 6.8.1 of the BFD base specification need to be initialized or manipulated differently depending on the session type. Ed-Note: Anything else?.

- o `bfd.DemandMode`: This variable MUST be initialized to 1 for session type `SBFDInitiator`, and MUST be initialized to 0 for session type `SBFDReflector`.

8. S-BFD Procedures

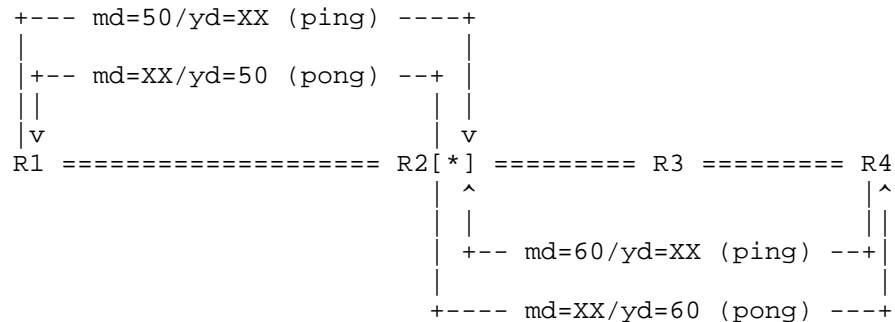
8.1. Initiator Procedures

S-BFD packets transmitted by an `SBFDInitiator` MUST set "your discriminator" field to an S-BFD discriminator corresponding to the remote entity.

S-BFD packets transmitted by an `SBFDInitiator` MUST NOT set "my discriminator" field to an S-BFD discriminator allocated for a local entity (and is being monitored by a local `SBFDReflector`). This is to prevent incoming response S-BFD packets, from a remote `SBFDReflector`, having "your discriminator" as a S-BFD discriminator of a local entity. Every `SBFDInitiator` is to have a unique "my discriminator", and SHOULD be allocated from the BFD discriminator pool if the

implementation employs the approach of having separate discriminator pools for BFD and S-BFD.

Below ASCII art describes high level concept of connectivity test using S-BFD. R2 allocates XX as the S-BFD discriminator for its network reachability purpose, and advertises XX to neighbors. ASCII art shows R1 and R4 performing a connectivity test to R2.



[*] Reflector BFD session on R2.
 === Links connecting network nodes.
 --- S-BFD packet traversal.

Figure 3: S-BFD Connectivity Test

8.1.1.1. SBFDInitiator State Machine

An SBFDInitiator may be a persistent session on the initiator with a timer for S-BFD packet transmissions. An SBFDInitiator may also be a module, a script or a tool on the initiator that transmits one or more S-BFD packets "when needed". For transient SBFDInitiators, the BFD state machine described in [RFC5880] may not be applicable. For persistent SBFDInitiators, the states and the state machine described in [RFC5880] will function but are more than necessary. The following diagram provides an optimized state machine for persistent SBFDInitiators. The notation on each arc represents the state of the SBFDInitiator (as received in the State field in the S-BFD packet) or indicates the expiration of the Detection Timer.

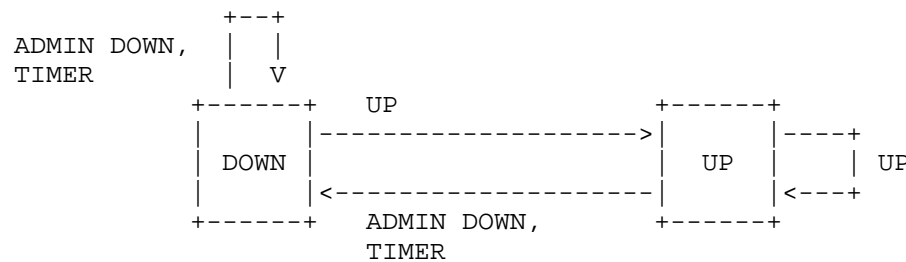


Figure 4: SBFDInitiator FSM

Note that the above state machine is different from the base BFD specification[RFC5880]. This is because the Init state is no longer applicable for the SBFDInitiator. Another important difference is the transition of the state machine from the Down state to the Up state when a packet with State Up is received by the initiator. The definitions of the states and the events have the same meaning as in the base BFD specification [RFC5880].

8.1.2. Details of S-BFD Packet Sent by SBFDInitiator

S-BFD packets sent by an SBFDInitiator is to have following contents:

- o Well-known UDP destination port assigned for S-BFD.
- o UDP source port as per described in [RFC5881], [RFC5883], [RFC5884] and [RFC5885].
- o "my discriminator" assigned by local node.
- o "your discriminator" corresponding to a remote entity.
- o "State" MUST be set to a value describing local state.
- o "Desired Min TX Interval" MUST be set to a value describing local desired minimum transmit interval.
- o "Required Min RX Interval" MUST be zero.
- o "Required Min Echo RX Interval" SHOULD be zero.
- o "Detection Multiplier" MUST be set to a value describing locally used multiplier value.
- o Demand (D) bit MUST be set.

8.2. Responder Procedures

A network node which receives S-BFD packets transmitted by an initiator is referred as responder. The responder, upon reception of S-BFD packets, is to perform necessary relevant validations described in [RFC5880], [RFC5881], [RFC5883], [RFC5884] and [RFC5885].

8.2.1. Responder Demultiplexing

A BFD control packet received by a responder is considered an S-BFD packet if the packet is on the well-known S-BFD port. When a responder receives an S-BFD packet, if the value in the "your discriminator" field is not one of S-BFD discriminators allocated for local entities, then this packet MUST NOT be considered for this mechanism. If the value in the "your discriminator" field is one of S-BFD discriminators allocated for local entities, then the packet is determined to be handled by a reflector BFD session responsible for the S-BFD discriminator. If the packet was determined to be processed further for this mechanism, then chosen reflector BFD session is to transmit a response BFD control packet using procedures described in Section 8.2.2, unless prohibited by local policies (ex: administrative, security, rate-limiter, etc).

8.2.2. Details of S-BFD Packet Sent by SBFDRreflector

S-BFD packets sent by an SBFDRreflector is to have following contents:

- o Well-known UDP destination port assigned for S-BFD.
- o UDP source port as described in [RFC5881], [RFC5883], [RFC5884] and [RFC5885].
- o "my discriminator" MUST be copied from received "your discriminator".
- o "your discriminator" MUST be copied from received "my discriminator".
- o "State" MUST be UP or ADMINDOWN. Clarification of reflector BFD session state is described in Section 8.7.
- o "Desired Min TX Interval" MUST be copied from received "Desired Min TX Interval".
- o "Required Min RX Interval" MUST be set to a value describing how many incoming control packets this reflector BFD session can handle. Further details are described in Section 8.7.
- o "Required Min Echo RX Interval" SHOULD be set to zero.
- o "Detection Multiplier" MUST be copied from received "Detection Multiplier".
- o Demand (D) bit MUST be cleared.

8.3. Diagnostic Values

Diagnostic value in both directions MAY be set to a certain value, to attempt to communicate further information to both ends. However, details of such are outside the scope of this specification.

8.4. The Poll Sequence

Poll sequence MAY be used in both directions. The Poll sequence MUST operate in accordance with [RFC5880]. An SBFDRreflector MAY use the Poll sequence to slow down that rate at which S-BFD packets are generated from an SBFDRinitiator. This is done by the SBFDRreflector using procedures described in Section 8.7 and setting the Poll (P) bit in the reflected S-BFD packet. The SBFDRinitiator is to then send the next S-BFD packet with the Final (F) bit set. If an SBFDRreflector receives an S-BFD packet with Poll (P) bit set, then the SBFDRreflector MUST respond with an S-BFD packet with Poll (P) bit cleared and Final (F) bit set.

8.5. Control Plane Independent (C)

Control plane independent (C) bit for an SBFDRinitiator sending S-BFD packets to a reflector BFD session MUST work according to [RFC5880]. Reflector BFD session also MUST work according to [RFC5880]. Specifically, if reflector BFD session implementation does not share fate with control plane, then response S-BFD packets transmitted MUST have control plane independent (C) bit set. If reflector BFD session implementation shares fate with control plane, then response S-BFD packets transmitted MUST NOT have control plane independent (C) bit set.

8.6. Additional SBFDRinitiator Behaviors

- o If the SBFDRinitiator receives a valid S-BFD packet in response to transmitted S-BFD packet to a remote entity, then the SBFDRinitiator SHOULD conclude that S-BFD packet reached the intended remote entity.
- o When a sufficient number of S-BFD packets have not arrived as they should, the SBFDRinitiator SHOULD declare loss of connectivity to the remote entity. The criteria for declaring loss of connectivity and the action that would be triggered as a result are outside the scope of this document.
- o Relating to above bullet item, it is critical for an implementation to understand the latency to/from the reflector BFD session on the responder. In other words, for very first S-BFD packet transmitted by the SBFDRinitiator, an implementation MUST NOT expect response S-BFD packet to be received for time equivalent to sum of latencies: initiator to responder and responder back to initiator.
- o If the SBFDRinitiator receives an S-BFD packet with Demand (D) bit set, the packet MUST be discarded.

8.7. Additional SBFDRReflector Behaviors

- o S-BFD packets transmitted by the SBFDRReflector MUST have "Required Min RX Interval" set to a value which expresses how many incoming S-BFD packets this SBFDRReflector can handle. The SBFDRReflector can control how fast SBFInitiators will be sending S-BFD packets to self by ensuring "Required Min RX Interval" indicates a value based on the current load.
- o If the SBFDRReflector wishes to communicate to some or all SBFInitiators that monitored local entity is "temporarily out of service", then S-BFD packets with "state" set to ADMINDOWN are sent to those SBFInitiators. The SBFInitiators, upon reception of such packets, MUST NOT conclude loss of connectivity to corresponding remote entity, and MUST back off packet transmission interval for the remote entity to an interval no faster than 1 second. If the SBFDRReflector is generating a response S-BFD packet for a local entity that is in service, then "state" in response BFD control packets MUST be set to UP.
- o If an SBFDRReflector receives an S-BFD packet with Demand (D) bit cleared, the packet MUST be discarded.

9. Scaling Aspect

This mechanism brings forth one noticeable difference in terms of scaling aspect: number of SBFDRReflector. This specification eliminates the need for egress nodes to have fully active BFD sessions when only one side desires to perform connectivity tests. With introduction of reflector BFD concept, egress no longer is required to create any active BFD session per path/LSP/function basis. Due to this, total number of BFD sessions in a network is reduced.

10. Co-existence with Traditional BFD

This mechanism has no issues being deployed with traditional BFDs ([RFC5881], [RFC5883], [RFC5884] and [RFC5885]) because S-BFD discriminators which allow this mechanism to function are explicitly reserved and separate UDP port values are used with S-BFD.

11. BFD Echo

BFD echo is outside the scope of this document.

12. Security Considerations

Same security considerations as [RFC5880], [RFC5881], [RFC5883], [RFC5884] and [RFC5885] apply to this document.

Additionally, implementing the following measures will strengthen security aspects of the mechanism described by this document.

- o Implementations MUST provide filtering capability based on source IP addresses of received S-BFD packets: [RFC2827].
- o Implementations MUST NOT act on received S-BFD packets containing Martian addresses as source IP addresses.
- o Implementations MUST ensure that response S-BFD packets generated to the initiator by the SBFDReflector have a reachable target (ex: destination IP address).
- o SBFDInitiator MAY pick crypto sequence number based on authentication mode configured.
- o SBFDReflector MUST NOT look at the crypto sequence number before accepting the packet.
- o SBFDReflector MAY look at the Key ID [I-D.ietf-bfd-generic-crypto-auth] in the incoming packet and verify the authentication data.
- o SBFDReflector MUST accept the packet if authentication is successful.
- o SBFDReflector MUST compute the Authentication data and MUST use the same sequence number that it received in the S-BFD packet that it is responding to.
- o SBFDInitiator MUST accept the S-BFD packet if it either comes with the same sequence number as it had sent or it's within the window that it finds acceptable (described in detail in [I-D.ietf-bfd-generic-crypto-auth])

Using the above method,

- o SBFDReflector continue to remain stateless despite using security.
- o SBFDReflector are not susceptible to replay attacks as they always respond to S-BFD packets irrespective of the sequence number carried.

- o An attacker cannot impersonate the responder since the SBFDInitiator will only accept S-BFD packets that come with the sequence number that it had originally used when sending the S-BFD packet.

13. IANA Considerations

A new value TBD1 is requested from the "Service Name and Transport Protocol Port Number Registry". The requested registry entry is:

```
Service Name (REQUIRED)
  s-bfd
Transport Protocol(s) (REQUIRED)
  udp
Assignee (REQUIRED)
  IESG <iesg@ietf.org>
Contact (REQUIRED)
  BFD Chairs <bfd-chairs@tools.ietf.org>
Description (REQUIRED)
  Seamless Bidirectional Forwarding Detection (S-BFD)
Reference (REQUIRED)
  draft-ietf-bfd-seamless-base
Port Number (OPTIONAL)
  TBD1 (Requesting 7784)
```

14. Acknowledgements

Authors would like to thank Jeffrey Haas for performing thorough reviews and providing number of suggestions. Authors would like to thank Girija Raghavendra Rao, Marc Binderberger, Les Ginsberg, Srihari Raghavan, Vanitha Neelamegam and Vengada Prasad Govindan from Cisco Systems for providing valuable comments. Authors would also like to thank John E. Drake for providing comments and suggestions.

15. Contributing Authors

Tarek Saad
Cisco Systems
Email: tsaad@cisco.com

Siva Sivabalan
Cisco Systems
Email: msiva@cisco.com

Nagendra Kumar
Cisco Systems
Email: naikumar@cisco.com

Mallik Mudigonda
Cisco Systems
Email: mmudigon@cisco.com

Sam Aldrin
Huawei Technologies
Email: aldrin.ietf@gmail.com

16. References

16.1. Normative References

- [I-D.ietf-bfd-seamless-use-case]
Aldrin, S., Bhatia, M., Mirsky, G., Kumar, N., and S. Matsushima, "Seamless Bidirectional Forwarding Detection (BFD) Use Case", draft-ietf-bfd-seamless-use-case-00 (work in progress), June 2014.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC5880] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD)", RFC 5880, June 2010.
- [RFC5881] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD) for IPv4 and IPv6 (Single Hop)", RFC 5881, June 2010.
- [RFC5883] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD) for Multihop Paths", RFC 5883, 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.

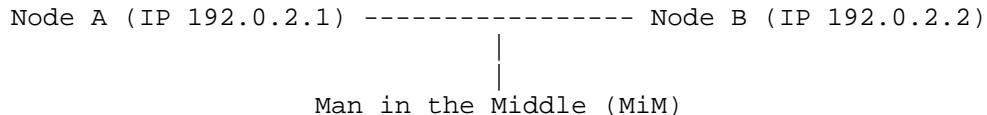
16.2. Informative References

- [I-D.ietf-bfd-generic-crypto-auth]
Bhatia, M., Manral, V., Zhang, D., and M. Jethanandani, "BFD Generic Cryptographic Authentication", draft-ietf-bfd-generic-crypto-auth-06 (work in progress), April 2014.
- [RFC2827] Ferguson, P. and D. Senie, "Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing", BCP 38, RFC 2827, May 2000.

[RFC5885] Nadeau, T. and C. Pignataro, "Bidirectional Forwarding Detection (BFD) for the Pseudowire Virtual Circuit Connectivity Verification (VCCV)", RFC 5885, June 2010.

Appendix A. Loop Problem

Consider a scenario where we have two nodes and both are S-BFD capable.



Assume node A reserved a discriminator 0x01010101 for target identifier 192.0.2.1 and has a reflector session in listening mode. Similarly node B reserved a discriminator 0x02020202 for its target identifier 192.0.2.2 and also has a reflector session in listening mode.

Suppose MiM sends a spoofed packet with MyDisc = 0x01010101, YourDisc = 0x02020202, source IP as 192.0.2.1 and dest IP as 192.0.2.2. When this packet reaches Node B, the reflector session on Node B will swap the discriminators and IP addresses of the received packet and reflect it back, since YourDisc of the received packet matched with reserved discriminator of Node B. The reflected packet that reached Node A will have MyDisc=0x02020202 and YourDisc=0x01010101. Since YourDisc of the received packet matched the reserved discriminator of Node A, Node A will swap the discriminators and reflects the packet back to Node B. Since reflectors MUST set the TTL of the reflected packets to 255, the above scenario will result in an infinite loop with just one malicious packet injected from MiM.

FYI: Packet fields do not carry any direction information, i.e., if this is Ping packet or reply packet.

Solutions

The current proposals to avoid the loop problem are:

- o Overload "D" bit (Demand mode bit): Initiator always sets the 'D' bit and reflector clears it. This way we can identify if a received packet was a reflected packet and avoid reflecting it back. However this changes the interpretation of 'D' bit.
- o Use of State field in the BFD control packets: Initiator will always send packets with State set to "DOWN" and reflector will send back packets with state field set to "UP". Reflectors will

never reflect any received packets with state as "UP". However the only issue is the use of state field differently i.e. state in the S-BFD control packet from initiator does not reflect the local state which is anyway not significant at reflector.

- o Use of local discriminator as My Disc at reflector: Reflector will always fill in My Discriminator with a locally allocated discriminator value (not reserved discriminators) and will not copy it from the received packet.

Authors' Addresses

Nobo Akiya
Cisco Systems

Email: nobo@cisco.com

Carlos Pignataro
Cisco Systems

Email: cpignata@cisco.com

Dave Ward
Cisco Systems

Email: wardd@cisco.com

Manav Bhatia
Ionos Networks

Email: manav@ionosnetworks.com

Santosh
Juniper Networks

Email: santoshpk@juniper.net

INTERNET-DRAFT
Intended Status: Informational
Expires: December 13, 2014

Sam Aldrin
(Huawei)
Manav Bhatia
(Ionos)
Greg Mirsky
(Ericsson)
Nagendra Kumar
(Cisco)
Satoru Matsushima
(Softbank)

June 11, 2014

Seamless Bidirectional Forwarding Detection (BFD) Use Case
draft-ietf-bfd-seamless-use-case-00

Abstract

This document provides various use cases for Bidirectional Forwarding Detection (BFD) such that simplified solution and extensions could be developed for detecting forwarding failures.

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>

Copyright and License 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. Introduction	3
1.1. Terminology	3
2. Introduction to Seamless BFD	3
3. Use Cases	4
3.1. Unidirectional Forwarding Path Validation	4
3.2. Validation of forwarding path prior to traffic switching	5
3.3. Centralized Traffic Engineering	5
3.4. BFD in Centralized Segment Routing	6
3.5. BFD to Efficiently Operate under Resource Constraints	6
3.6. BFD for Anycast Address	7
3.7. BFD Fault Isolation	7
3.8. Multiple BFD Sessions to Same Target	7
3.9. MPLS BFD Session Per ECMP Path	8
4. Security Considerations	9
5. IANA Considerations	9
6. References	9
6.1. Normative References	9
7. Authors' Addresses	9
8. Contributors	10

1. Introduction

Bidirectional Forwarding Detection (BFD) is a lightweight protocol, as defined in [RFC5880], used to detect forwarding failures. Various protocols and applications rely on BFD for failure detection. Even though the protocol is simple and lightweight, there are certain use cases, where a much faster setting up of sessions and continuity check of the data forwarding paths is necessary. This document identifies those use cases such that necessary enhancements could be made to BFD protocol to meet those requirements.

There are various ways to detecting faults and BFD protocol was designed to be a lightweight "Hello" protocol to detect data plane failures. With dynamic provisioning of forwarding paths at a large scale, establishing BFD sessions for each of those paths creates complexity, not only from operations point of view, but also the speed at which these sessions could be established or deleted. The existing session establishment mechanism of the BFD protocol need to be enhanced in order to minimize the time for the session to come up and validate the forwarding path.

This document specifically identifies those cases where certain requirements could be derived to be used as reference, so that, protocol enhancements could be developed to address them. Whilst the use cases could be used as reference for certain requirements, it is outside the scope of this document to identify all of the requirements for all possible enhancements. Specific solutions and enhancement proposals are outside the scope of this document as well.

1.1. Terminology

The reader is expected to be familiar with the BFD, IP, MPLS and SR terminology and protocol constructs. This section identifies only the new terminology introduced.

2. Introduction to Seamless BFD

BFD as defined in standard [RFC5880] requires two network nodes, as part of handshake, exchange discriminators. This will enable the sender and receiver of BFD packets of a session to be identified and check the continuity of the forwarding path. [RFC5881] defines single hop BFD whereas [RFC5883] and [RFC5884] defines multi-hop BFD.

In order to establish BFD sessions between network entities and seamlessly be able to have the session up and running, BFD protocol should be capable of doing that. These sessions have to be established a priori to traffic flow and ensure the forwarding path is available and connectivity is present. With handshake mechanism

within BFD protocol, establishing sessions at a rapid rate and ensuring the validity or existence of working forwarding path, prior to the session being up and running, becomes complex and time consuming. In order to achieve seamless BFD sessions, it requires a mechanism where the ability to specify the discriminators and the ability to respond to the BFD control packets by the network node, should already be negotiated ahead of the session becoming active. Seamless BFD by definition will be able to provide those mechanisms within the BFD protocol in order to meet the requirements and establish BFD sessions seamlessly, with minimal overhead, in order to detect forwarding failures.

As an example of how Seamless BFD (S-BFD) works, a set of network entities are first identified, to which BFD sessions have to be established. Each of those network nodes, will be assigned a special BFD discriminator, to establish a BFD session. These network nodes will also create a BFD session instance that listens for incoming BFD control packets. Mappings between selected network entities and corresponding special BFD discriminators are known to other network nodes belonging in the same network. A network node in such network is then able to send a BFD control packet to a particular target with corresponding special BFD discriminator. Target network node, upon reception of such BFD control packet, will transmit a response BFD control packet back to the sender.

3. Use Cases

As per the BFD protocol [RFC5880], BFD sessions are established using handshake mechanism prior to validating the forwarding path. This section outlines some of the use cases where the existing mechanism may not be able to satisfy the requirements. In addition, some of the use cases will also be identifying the need for expedited BFD session establishment with preserving benefits of forwarding failure detection using existing BFD specifications.

3.1. Unidirectional Forwarding Path Validation

Even though bidirectional verification of forwarding path is useful, there are scenarios when only one side of the BFD, not both, is interested in verifying continuity of the data plane between a pair of nodes. One such case is, when a static route uses BFD to validate reachability to the next-hop IP router. In this case, the static route is established from one network entity to another. The requirement in this case is only to validate the forwarding path for that statically established path. Validating the reverse direction is not required in this case. Many of these network scenarios are being proposed as part of segment routing [TBD]. Another example is when a unidirectional tunnel uses BFD to validate reachability to the egress

node.

If the traditional BFD is to be used, the target network entity has to be provisioned as well, even though the reverse path validation with BFD session is not required. But with unidirectional BFD, the need to provision on the target network entity is not needed. Once the mechanism within the BFD protocol is in place, where the source network entity knows the target network entity's discriminator, it starts the session right away. When the targeted network entity receives the packet, it knows that BFD packet, based on the discriminator and processes it. That do not require to have a bi-directional session establishment, hence the two way handshake to exchange discriminators is not needed as well.

The primary requirement in this use case is to enable session establishment from source network entity to target network entity. This translates to, the target network entity for the BFD session, upon receiving the BFD packet, should start processing for the discriminator received. This will enable the source network entity to establish a unidirectional BFD session without bidirectional handshake of discriminators for session establishment.

3.2. Validation of forwarding path prior to traffic switching

BFD provides data delivery confidence when reachability validation is performed prior to traffic utilizing specific paths/LSPs. However this comes with a cost, where, traffic is prevented to use such paths/LSPs until BFD is able to validate the reachability, which could take seconds due to BFD session bring-up sequences [RFC5880], LSP ping bootstrapping [RFC5884], etc. This use case does not require to have sequences for session negotiation and discriminator exchanges in order to establish the BFD session.

When these sequences for handshake are eliminated, the network entities need to know what the discriminator values to be used for the session. The same is the case for S-BFD, i.e., when the three-way handshake mechanism is eliminated during bootstrap of BFD sessions. Due to this faster reachability validation of BFD provisioned paths/LSPs could be achieved. In addition, it is expected that some MPLS technologies will require traffic engineered LSPs to get created dynamically, driven by external applications, e.g. in Software Defined Networks (SDN). It would be desirable to perform BFD validation very quickly to allow applications to utilize dynamically created LSPs in timely manner.

3.3. Centralized Traffic Engineering

Various technologies in the SDN domain have evolved which involves

controller based networks, where the intelligence, traditionally placed in the distributed and dynamic control plane, is separated from the data plane and resides in a logically centralized place. There are various controllers which perform this exact function in establishing forwarding paths for the data flow. Traffic engineering is one important function, where the traffic is engineered depending upon various attributes of the traffic as well as the network state.

When the intelligence of the network resides in the centralized entity, ability to manage and maintain the dynamic network becomes a challenge. One way to ensure the forwarding paths are valid and working is to establish BFD sessions within the network. When traffic engineering tunnels are created, it is operationally critical to ensure that the forwarding paths are working prior to switching the traffic onto the engineered tunnels. In the absence of control plane protocols, it is not only the desire to verify the forwarding path but also an arbitrary path in the network. With tunnels being engineered from the centralized entity, when the network state changes, traffic has to be switched without much latency and black holing of the data.

Traditional BFD session establishment and validation of the forwarding path must not become bottleneck in the case of centralized traffic engineering. If the controller or other centralized entity is able to instantly verify a forwarding path of the TE tunnel, it could steer the traffic onto the traffic engineered tunnel very quickly thus minimizing adverse effect on a service. This is especially useful and needed when the scale of the network and number of TE tunnels is too high. Session negotiation and establishment of BFD sessions to identify valid paths is way too high in terms of time and providing network redundancy becomes a critical issue.

3.4. BFD in Centralized Segment Routing

Centralized controller based Segment Routing network monitoring technique, is described in [I-D.geib-spring-oam-usecase]. In validating this use case, one of the requirements is to ensure the BFD packet's behavior is according to the requirement and monitoring of the segment, where the packet is U-turned at the expected node. One of the criterion is to ensure the continuity check to the adjacent segment-id.

3.5. BFD to Efficiently Operate under Resource Constraints

When BFD sessions are being setup, torn down or parameters (i.e. interval, multiplier, etc) are being modified, BFD protocol requires additional packets outside of scheduled packet transmissions to complete the negotiation procedures (i.e. P/F bits). There are

scenarios where network resources are constrained: a node may require BFD to monitor very large number of paths, or BFD may need to operate in low powered and traffic sensitive networks, i.e. microwave, low powered nano-cells, etc. In these scenarios, it is desirable for BFD to slow down, speed up, stop or resume at will without requiring additional BFD packets to be exchanged.

3.6. BFD for Anycast Address

BFD protocol requires the two endpoints to host BFD sessions, both sending packets to each other. This BFD model does not fit well with anycast address monitoring, as BFD packets transmitted from a network node to an anycast address will reach only one of potentially many network nodes hosting the anycast address.

3.7. BFD Fault Isolation

BFD multi-hop and BFD MPLS traverse multiple network nodes. BFD has been designed to declare failure upon lack of consecutive packet reception, which can be caused by any fault anywhere along the path. Fast failure detection provides great benefits, as it can trigger recovery procedures rapidly. However, operators often have to follow up, manually or automatically, to attempt to identify and localize the fault which caused the BFD sessions to fail. Usage of other tools to isolate the fault may cause the packets to traverse differently throughout the network (i.e. ECMP). In addition, longer it takes from BFD session failure to fault isolation attempt, more likely that fault cannot be isolated, i.e. fault can get corrected or routed around. If BFD had built-in fault isolation capability, fault isolation can get triggered at the earliest sign of fault and such packets will get load balanced in very similar way, if not the same, as BFD packets which went missing.

3.8. Multiple BFD Sessions to Same Target

BFD is capable of providing very fast failure detection, as relevant network nodes continuously transmitting BFD packets at negotiated rate. If BFD packet transmission is interrupted, even for a very short period of time, that can result in BFD to declare failure irrespective of path liveliness. It is possible, on a system where BFD is running, for certain events, intentionally or unintentionally, to cause a short interruption of BFD packet transmissions. With distributed architectures of BFD implementations, this can be protected, if a node was to run multiple BFD sessions to targets, hosted on different parts of the system (ex: different CPU instances). This can reduce BFD false failures, resulting in more stable network.

3.9. MPLS BFD Session Per ECMP Path

BFD for MPLS, defined in [RFC5884], describes procedures to run BFD as LSP in-band continuity check mechanism, through usage of MPLS echo request [RFC4379] to bootstrap the BFD session on the egress node. Section 4 of [RFC5884] also describes a possibility of running multiple BFD sessions per alternative paths of LSP. However, details on how to bootstrap and maintain correct set of BFD sessions on the egress node is absent.

When an LSP has ECMP segment, it may be desirable to run in-band monitoring that exercises every path of ECMP. Otherwise there will be scenarios where in-band BFD session remains up through one path but traffic is black-holing over another path. One way to achieve BFD session per ECMP path of LSP is to define procedures that update [RFC5884] in terms of how to bootstrap and maintain correct set of BFD sessions on the egress node. However, that may require constant use of MPLS Echo Request messages to create and delete BFD sessions on the egress node, when ECMP paths and/or corresponding load balance hash keys change. If a BFD session over any paths of the LSP can be instantiated, stopped and resumed without requiring additional procedures of bootstrapping via MPLS echo request, it would simplify implementations and operations, and benefits network devices as less processing are required by them.

4. Security Considerations

There are no new security considerations introduced by this draft.

5. IANA Considerations

There are no new IANA considerations introduced by this draft

6. References

6.1. 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)", RFC5880, June 2010.
- [RFC5881] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD)", RFC5881, June 2010.
- [RFC5883] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD) for Multihop Paths", RFC5883, June 2010.
- [RFC5884] Aggarwal, R., Kompella, K., Nadeau, T., and G. Swallow, "Bidirectional Forwarding Detection (BFD) for MPLS Label Switched Paths (LSPs)", RFC5884, June 2010.

7. Authors' Addresses

Sam Aldrin
Huawei Technologies
2330 Central Expressway
Santa Clara, CA 95051

Email: aldrin.ietf@gmail.com

Manav Bhatia
Ionos Networks

Email: manav@ionosnetworks.com

Satoru Matsushima
Softbank

Email: satoru.matsushima@g.softbank.co.jp

Greg Mirsky
Ericsson

Email: gregory.mirsky@ericsson.com

Nagendra Kumar
Cisco

Email: naikumar@cisco.com

8. Contributors

Carlos Pignataro
Cisco Systems

Email: cpignata@cisco.com

Glenn Hayden
ATT

Email: gh1691@att.com

Santosh P K
Juniper

Email: santoshpk@juniper.net

Mach Chen
Huawei

Email: mach.chen@huawei.com

Nobo Akiya
Cisco Systems

Email: nobo@cisco.com

MPLS Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 1, 2015

G. Mirsky
J. Tantsura
Ericsson
I. Varlashkin
EasyNet
June 30, 2014

Bidirectional Forwarding Detection (BFD) Directed Return Path
draft-mirsky-mpls-bfd-directed-00

Abstract

Bidirectional Forwarding Detection (BFD) is expected to monitor bi-directional paths. When forward direction of a BFD session is to monitor explicitly routed path there is a need to be able to direct far-end 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 January 1, 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. 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	3
3.1. Case of MPLS Data Plane	4
3.1.1. BFD Reverse Path TLV	4
3.1.2. Segment Routing Tunnel sub-TLV	4
3.2. Case of IPv6 Data Plane	5
4. IANA Considerations	6
4.1. TLV	6
4.2. Sub-TLV	6
5. Security Considerations	7
6. Acknowledgements	7
7. Normative References	7
Authors' Addresses	8

1. Introduction

The [RFC5880], [RFC5881], and the [RFC5883] established BFD protocol for IP networks and the [RFC5884] set rules of using BFD Asynchronous mode over IP/MPLS LSPs. All standards implicitly assume that the far-end BFD peer will use the best route regardless of route being used to send BFD control packets towards it. As result, if the near-end 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 is not guaranteed or free of positive negatives.

This document proposes to use BFD Return Path TLV extension to LSP Ping [RFC4379] to instruct the far-end BFD peer to use explicit path for its BFD control packets associated with the particular BFD session. 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, in IP and IP/MPLS networks the best route between two IP nodes is likely to be co-routed in the stable network environment so that implicit BFD requirement is being fulfilled. If BFD is tasked to monitor unidirectional explicitly routed path, e.g. MPLS LSP, its control packets in forward direction would be in-band due to mechanism defined in [RFC5884] and [RFC5586]. But the reverse direction of the BFD session would still follow the best route and that presents following problems in regard to detecting defects on the unidirectional explicit path:

- failure detection on the reverse path cannot be interpreted as bi-directional failure and thus trigger, for example, protection switchover of the forward direction;
- if reverse direction is in Down state, the head-end node would not receive indication of forward direction failure from its far-end peer.

To address these challenges the far-end 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.

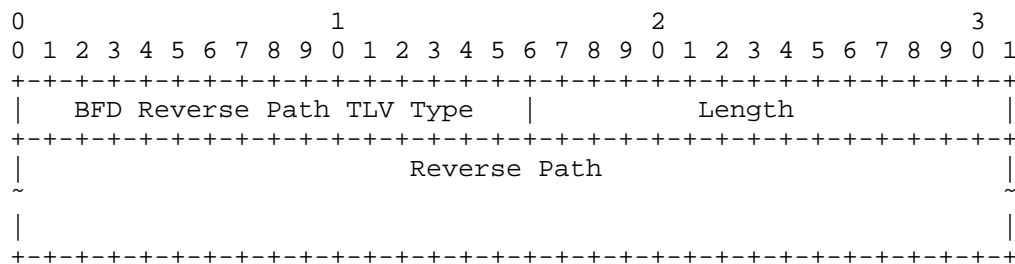


Figure 1: BFD Reverse Path TLV

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.

3.1.2. Segment Routing Tunnel sub-TLV

With MPLS data plane explicit path can be either Static or RSVP-TE LSP, or Segment Routing tunnel. In case of Static or RSVP-TE LSP [RFC7110] defined sub-TLVs to identify explicit return path. For the Segment Routing with MPLS data plane case a new sub-TLV is defined in this document as presented in Figure 2.

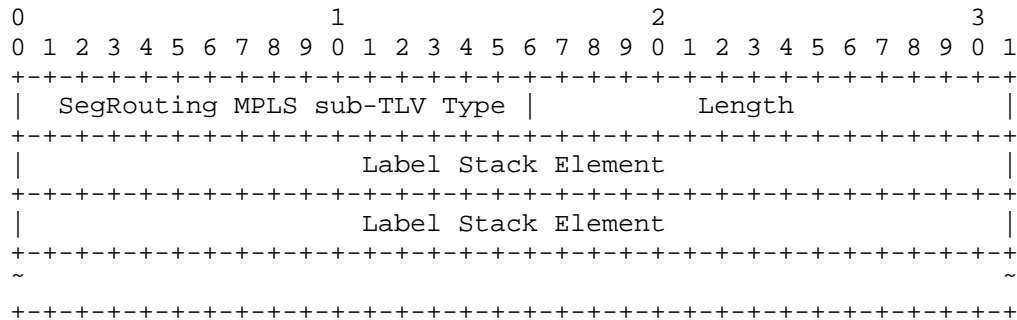


Figure 2: Segment Routing MPLS Tunnel sub-TLV

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

The 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. IP networks, unlike IP/MPLS, do not require use of LSP ping with BFD Discriminator TLV[RFC4379] to bootstrap BFD session. But 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.

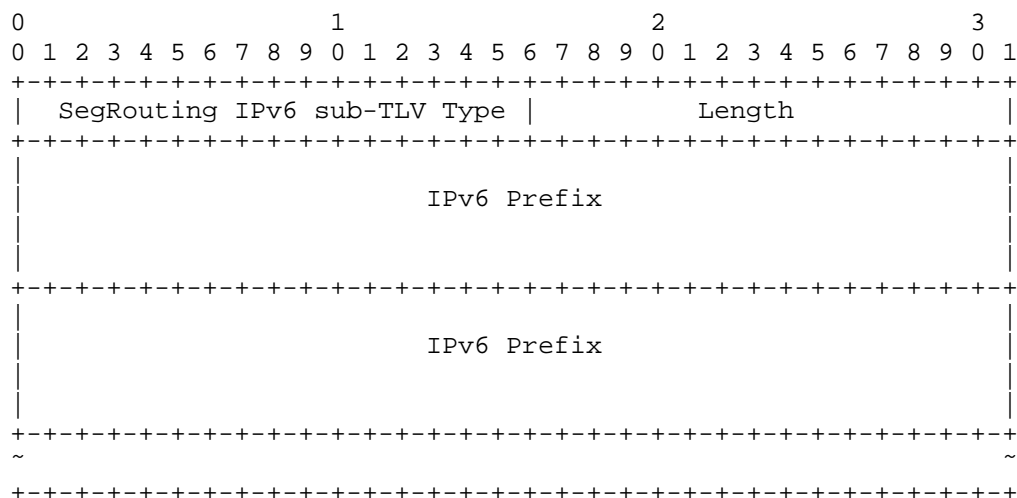


Figure 3: Segment Routing IPv6 Tunnel sub-TLV

4. IANA Considerations

4.1. 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

4.2. Sub-TLV

The IANA is requested to assign one new sub-TLV type from "Multiprotocol Label Switching Architecture (MPLS) Label Switched Paths (LSPs) Ping Parameters - TLVs" registry, "Sub-TLVs for TLV Type 1" 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. Security Considerations

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

6. Acknowledgements

7. Normative References

- [I-D.previdi-6man-segment-routing-header]
Previdi, S., Filsfils, C., Field, B., and I. Leung, "IPv6 Segment Routing Header (SRH)", draft-previdi-6man-segment-routing-header-01 (work in progress), June 2014.
- [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.
- [RFC5586] Bocci, M., Vigoureux, M., and S. Bryant, "MPLS Generic Associated Channel", RFC 5586, June 2009.
- [RFC5880] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD)", RFC 5880, June 2010.
- [RFC5881] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD) for IPv4 and IPv6 (Single Hop)", RFC 5881, June 2010.
- [RFC5883] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD) for Multihop Paths", RFC 5883, 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.

[RFC7110] Chen, M., Cao, W., Ning, S., Jounay, F., and S. Delord,
"Return Path Specified Label Switched Path (LSP) Ping",
RFC 7110, January 2014.

Authors' Addresses

Greg Mirsky
Ericsson

Email: gregory.mirsky@ericsson.com

Jeff Tantsura
Ericsson

Email: jeff.tantsura@ericsson.com

Ilya Varlashkin
EasyNet

Email: Ilya.Varlashkin@easynet.com

Internet Engineering Task Force
Internet-Draft
Intended status: Informational
Expires: November 6, 2014

B. Snyder
iDirect Technologies
N. Akiya
Cisco Systems
May 5, 2014

BFD Proxy for Connections over Monitored Links
draft-snyder-bfd-proxy-connections-monitored-links-00

Abstract

This document describes a Bidirectional Forwarding Detection (BFD) proxy mechanism to allow intermediate networking equipment (ex: Satellite HUB/Modem) to intercept BFD packets and to generate BFD packets to relay the health of connection monitored links.

Note that this is an informational document that does not propose any changes to the BFD protocol.

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 November 6, 2014.

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. Introduction	2
1.1. Terminology	2
1.2. Background	3
2. Overview	4
3. BFD Proxy Placement	5
4. BFD Proxy Procedures	5
4.1. BFD Control Packet Interception	5
4.2. OAM Object	6
4.3. BFD Proxying	6
4.4. Outroute Considerations	8
4.5. Inroute Considerations	8
5. Possible Integration Improvements	9
6. Security Considerations	9
7. IANA Considerations	10
8. Acknowledgements	10
9. References	10
9.1. Normative References	10
9.2. Informative References	10
Authors' Addresses	10

1. Introduction

1.1. Terminology

The following acronyms/terminologies are used in this document:

- o BFD - Bidirectional Forwarding Detection
- o DLEP - Dynamic Link Exchange Protocol
- o L2 - Layer 2
- o L3 - Layer 3
- o Outroute - The broadcast link from hub to modem(s) in a satellite network.

- o Downstream - Synonymous to Outroute.
- o OTA - Over the Air
- o Inroute - The unicast uplink that a modem transmits to the hub side on in a satellite network.
- o Upstream - Synonymous to Inroute.

1.2. Background

Bidirectional Forwarding Detection (BFD) is an application agnostic and link type independent keep alive protocol which has widely been implemented and deployed. The BFD protocol can be configured with a fast interval to provide rapid failure detection or configured with a slower interval to provide slower failure detection. The faster the interval, the more BFD packets are transmitted and received, consuming more system and network resources.

Some links have connection monitoring functionality of its own, and some of these connection monitored links have constraints (ex: limited or expensive bandwidth). Applications over such links often still desire rapid failure detection through exchanging keep-alive packets (ex: BFD). However, the consequence of such can significantly degenerate the value of the links. For example, running BFD over a link with limited bandwidth can result in a significant portion of the bandwidth being consumed by BFD packets.

One example of such scenario is:

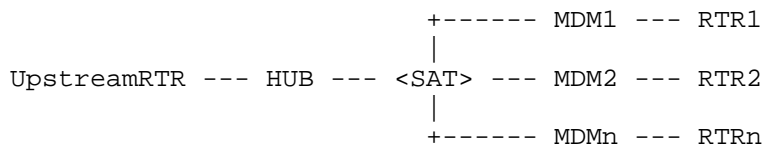


Figure 1: Star Satellite Network

The HUB components consist of a protocol stack which processes and inspects all outbound packets in order to optimize traffic for a high delay low bandwidth environments. (Ex: TCP proxy, compression, encryption). This stack also contains a L2 switch to demultiplex outroute traffic towards the proper modem via a MAC learning switch. In this component is also station keeping algorithms and QOS schedulers.

The MDM components have the same protocol stack (without the demultiplexing required) to optimize the traffic flow for the TDMA

inroutes. The interfaces of a modem are 1 RF interface and 1 to 8 ethernet interfaces.

When routers connected to HUB and MDMs run BFD to monitor the L3 reachability through the Satellite network, expensive Satellite bandwidth gets consumed with large number of BFD packets traversing over it.

Dynamic Link Exchange Protocol (DLEP), [I-D.ietf-manet-dlep], tackles this problem by introducing a protocol that can communicate the state of monitored links to routing devices. DLEP also maintains and communicates an extensive set of information (ex: link quality). A wide range of DLEP responsibilities result in a large effort for vendors to develop this protocol. DLEP, in addition, will require further effort to get integrated into various applications (ex: IGP) for the information to be beneficial.

BFD, on the other hand, has widely been implemented and deployed. If applications, already capable of speaking BFD, only require keeping track of a connection state over monitored links, and not any other information provided by DLEP, then the BFD proxy, described in this document, can be implemented on intermediate networking equipment to allow:

- o Connected network equipment (i.e. routers) to continue using BFD for continuity check.
- o BFD packets to consume minimal bandwidth on monitored links.

2. Overview

This informational document describes a BFD proxy mechanism that allows for connection monitoring intermediate networking equipment (ex: Satellite HUB/Modem) to use BFD packets in order to communicate the state of monitored links whilst significantly reducing the network bandwidth consumed by BFD packets.

The BFD proxy is a link state aware module that resides on the intermediate networking equipment, and intercepts all BFD packets coming in from the connected network equipment.

The first task of the BFD proxy is to transmit BFD control packets to the connected network equipment in order to communicate the state of monitored links, based on its knowledge of link state. The BFD proxy can inject BFD STATE change events towards the connected network equipment. When the device under monitoring is present, the BFD proxy can inject BFD packets with BFD_UP state. When the device

under monitoring has left the network, the BFD proxy can inject BFD packets with BFD_DOWN state.

The second task, to reduce network bandwidth is handled both at the BFD level (by proxying) and at the L3 level. By proxying the BFD control packets one can keep all the BFD overhead off the monitored (and often expensive) bandwidth links. The use of BFD also allows for network designers to configure L3 keep-alive/HELLO timers to be increased thereby reducing OTA bandwidth usage of un-proxied data flows. With BFD monitoring and alerting, L3 convergence is bound by a combination of link state awareness and IGP Hello time (in either direction). The monitored link's state (ex: satellite modem) can be immediately propagated when transitioning between in and out of network. Additionally, configurations and protocols will be discussed that have been determined to be optimal to this use case.

This document will also suggest multiple integration improvements that all interested parties (routing vendors and modem vendors) could implement to further optimize convergence time and bandwidth usage. The network configuration is that of a star design, where thousands of CE routers each behind a satellite remote will attach to one hub upstream router via desired L3 protocols. Whilst, many networks do utilize mobility and roaming, they are always aware of whom they are connecting too (either one or more possible HUBs, but only one at a time). As the goal is simply to assist the routers in understanding radio link state to optimize routing convergence, BFD is the optimal way of meeting this need.

3. BFD Proxy Placement

The BFD proxy module MUST be placed on a system such that it meets following two criteria:

1. The BFD proxy module can access the state of monitored links and neighbors reachable through it.
2. The BFD proxy module can access all single-hop BFD control packets coming in from the connected network equipment.

4. BFD Proxy Procedures

4.1. BFD Control Packet Interception

The BFD proxy module MUST intercept all single-hop BFD control packets (referred to as BFD packets from hereon) coming in from the connected network equipment. Criteria to identify single-hop BFD control packets are:

1. IP/UDP Packet
2. IP TTL 255 ([RFC5881] and for [RFC5082])
3. UDP destination port 3784 ([RFC5881])

4.2. OAM Object

The BFD proxy module SHOULD maintain an OAM object per neighbor reachable through monitored links. This OAM object is to have the state of the neighbor (i.e. available or not available), stores local BFD discriminator value and caches the latest BFD packet intercepted. When the BFD proxy module intercepts a BFD packet, destination MAC address is used to locate the OAM object. If corresponding OAM object is not found, then perform local checks to see if one should get created. If the check passes, create the OAM object. Otherwise do not create one.

4.3. BFD Proxying

Upon intercepting a BFD packet and locating a corresponding OAM object, the BFD proxy module is to follow procedures described in this sub-section.

1. If there is no OAM object, no further action is taken.
2. If the state of the neighbor in the OAM object is "not-available", then no further action is taken.
3. If the State field of intercepted BFD control packet is:
 - * ADMIN_DOWN: Forward the intercepted packet OTA to alert the real destination.
 - * DOWN: Create a BFD packet and copy the contents from intercepted packet, with the following modifications:
 - + Swap source and destination MAC addresses.
 - + Swap source and destination IP addresses.
 - + Set "my discriminator" field.
 - + Clear "your discriminator" field.

Send constructed BFD packet to the connected network equipment.

- * INIT: If "your discriminator" does not match expected value, then no further action is taken. Otherwise, create a BFD packet and copy the contents from the intercepted packet, with the following modifications:
 - + Swap source and destination MAC addresses.
 - + Swap source and destination IP addresses.
 - + Swap "my discriminator" and "your discriminator" fields.
 - + Set "State" field to UP.Send constructed BFD packet to the connected network equipment.
- * UP: If "your discriminator" does not match the expected value, then no further action is taken. Otherwise, create a BFD packet and copy the contents from the intercepted packet, with following modifications:
 - + Swap source and destination MAC addresses.
 - + Swap source and destination IP addresses.
 - + Swap "my discriminator" and "your discriminator" fields.Send constructed BFD packet to the connected network equipment.

In addition, following procedures MAY be applied:

- o When a BFD control packet is sent to the connected network equipment, the UDP checksum is set to 0 to avoid the recalculation.
- o When the state of the neighbor in the OAM object changes from "available" to "not-available", then the BFD proxy module SHOULD send unsolicited BFD control packet with state field as DOWN to the connected network equipment. If this is not done, then absence of a "reply" BFD control packet from the BFD proxy will cause the sending router to timeout the connection after 3 drops (or whatever the multiplier is set too).
- o Once the BFD proxy is intercepting BFD control packets and is in UP state, Poll sequence MAY be initiated to increase values in Minimum TX Interval and Minimum RX Interval fields to reduce the

number of BFD control packets on the link connecting the network equipment and the intermediate network equipment.

- o Since on a Satellite Star Network configuration the outroute and inroute have different bandwidth considerations, there are unique integration concerns which are described below

4.4. Outroute Considerations

In a star satellite network, the outroute is a broadcast channel which all remotes receive. While there need not be any restrictions on L3 routing protocols, it does naturally follow that an IGP is a good choice. Specifically, one which allows for asynchronous timers.

Terrestrial convergence timing with BFD (sub second) is in the most common error cases (rainfade, mobility switching) not a realistic goal as the RF algorithms that determine out of network will take on the order of seconds (15 in this specific case). Therefore should a modem leave the network for any reason, the minimum convergence time at the hub side is 15 seconds plus BFD timing to recognize the link loss. Hence, the goal being to minimize bandwidth overhead to make this as short as possible above layer 1 timing. A further consideration is convergence timing when a modem comes back into network. If the L3 timers are made too high, then it can take too long to recognize a positive network state. The outroute being a broadcast medium, can work well within these parameters if for instance the outroute L3 hello timing was every 5 seconds. That's only 1 multicast hello packet to cover the entire network and will bound the convergence time to within 5 seconds.

4.5. Inroute Considerations

On the inroute, network bandwidth is much harder to come by, because the aggregate throughput of all inroutes is shared amongst all modems (potentially numbering in the tens of thousands), and is very expensive. Also, it is unicast to the hub side only. Therefore any decisions made here on timing and data transmissions must scale to the tens of thousands in design principles. This fact is the catalyst for preferring asynchronous timers. Ideally, one can rely on the hello packet of a multicast outroute to kick off convergence, and the hello timing of the inroute can be tuned down as much as possible, to optimize inroute usage. This is possible with EIGRP and IS-IS protocols. Unfortunately, BGP and OSPF require synchronized timers, which means it is impossible to weigh equally the convergence timing while protecting inroute bandwidth.

Additionally, further integration simplicity can optionally be achieved if desired. Notice the timing of 15 seconds to recognize

modem link state is also 3 (a common multiplier setting) times the 5 seconds (common hello message timing). Therefore, it is possible, if one is only interested in monitoring link state, to not utilize BFD on all the remote LANs, as 15 seconds is enough time for the L3 messaging to alert the router to a network issue and just about the same time that the hub side will notice. This is useful to simplify operational complexity and management of the thousands or tens of thousands of installed networks. If one would like BFD to monitor modem LAN state as well, then it would be required regardless.

5. Possible Integration Improvements

The following improvements could help with overhead and convergence timing in all monitored network environments. They can require changes on routing or modem equipment to further optimize these types of networks:

- o BFD timer - Allowing for connected network equipment to configure a high BFD interval value. One of BFD's missions is to support sub second failure notification. This document puts forth a useful situation in which BFD is a great help, but does not require such strict timing. In fact, it would scale better with much looser restrictions on timer configuration.
- o BFD demand mode implementation - If vendors had implemented demand mode, it would be possible for the BFD proxy to send D bit to the connected network to significantly minimize BFD packets traversing over local link connected to the network equipment, without tweaking Minimum TX Interval and Minimum RX Interval values. This would reduce processing of BFD packets by the BFD proxy module even further.
- o BFD protocol - Adding into the core protocol the notion of a proxier could assist with support of authentication in this use case, if desired.

6. Security Considerations

The proxying by the BFD proxy module will require additional considerations (i.e. knowing authentication types/keys of each neighbor) to handle BFD packets with BFD authentication data (described in Section 6.7 of [RFC5880]. This document only describes procedures to handle BFD packets without BFD authentication data. However, because the mechanism is only applicable to single-hop BFD ([RFC5881]) and GTSM (i.e. check for TTL=255) already provides fairly strong security, lack of BFD authentication support is not considered threatening.

7. IANA Considerations

This document does not define any code points.

8. Acknowledgements

Authors would like to thank Adrian Farrel for providing a suggestion to generalize the solution to all monitored links.

9. References

9.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC5880] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD)", RFC 5880, June 2010.
- [RFC5881] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD) for IPv4 and IPv6 (Single Hop)", RFC 5881, June 2010.

9.2. Informative References

- [I-D.ietf-manet-dlep] Ratliff, S., Cisco, C., Harrison, G., Jury, S., and D. Satterwhite, "Dynamic Link Exchange Protocol (DLEP)", draft-ietf-manet-dlep-05 (work in progress), February 2014.
- [RFC5082] Gill, V., Heasley, J., Meyer, D., Savola, P., and C. Pignataro, "The Generalized TTL Security Mechanism (GTSM)", RFC 5082, October 2007.

Authors' Addresses

Brian Snyder
iDirect Technologies

Email: bsnyder@idirect.net

Nobo Akiya
Cisco Systems

Email: nobo@cisco.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 5, 2015

V. Govindan
S. Salam
A. Sajassi
Cisco Systems
July 4, 2014

Proactive fault detection in EVPN
draft-vgovindan-l2vpn-evpn-bfd-02

Abstract

This document proposes a proactive, in-band network OAM mechanism to detect connectivity faults that affect unicast and multi-destination paths in an EVPN network. The multi-destination paths are used by Broadcast, unknown Unicast and Multicast (BUM) traffic. The mechanisms proposed in the draft use the principles of the widely adopted Bidirectional Forwarding Detection (BFD) protocol.

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 January 5, 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. Introduction	2
1.1. Terminology	3
1.2. Motivation for running BFD at the network layer of EVPN .	3
2. Scope of fault detection mechanisms proposed in this document	4
2.1. Fault Detection of BUM traffic using ingress replication (MP2P)	4
2.1.1. Bootstrapping BFD sessions at the head of the MP2P tunnel	4
2.1.2. Bootstrapping BFD sessions at the tail nodes of the MP2P tunnel	5
2.2. Fault Detection of BUM traffic using P2MP tunnels (LSM) .	5
2.2.1. Bootstrapping BFD sessions at the root of the P2MP tunnel	6
2.2.2. Bootstrapping BFD sessions at the tail nodes of the P2MP tunnel	6
2.3. Fault Detection of unicast traffic	6
3. BFD packet encapsulation	6
3.1. Using GAL/G-ACh encapsulation without IP headers	6
3.1.1. Ingress replication	6
3.1.2. LSM	7
3.1.3. Unicast	7
3.2. Using IP headers	7
4. Scalability Considerations	7
5. Security Considerations	7
6. IANA Considerations	7
7. Acknowledgments	8
8. References	8
8.1. Normative References	8
8.2. Informative References	8
8.3. URIs	9
Authors' Addresses	9

1. Introduction

[I-D.salam-l2vpn-evpn-oam-req-frmwk] outlines the OAM requirements of Ethernet VPN networks [I-D.ietf-l2vpn-evpn]. This document proposes mechanisms for proactive fault detection at the network OAM layer of EVPN. These mechanisms could either be deployed for periodic and proactive monitoring, or be triggered by specific events to aid troubleshooting. EVPN fault detection mechanisms need to consider unicast and BUM traffic separately since they map to different FECs in EVPN. Since BUM traffic can be transported using MP2P or P2MP

tunnels, this document proposes slightly different fault detection mechanisms to suit each type using the principles of BFD over MPLS LSPs [RFC5884] and Point-to-multipoint BFD[I-D.ietf-bfd-multipoint]. Please note that this document uses the term EVPN loosely to include [I-D.ietf-l2vpn-evpn], [I-D.ietf-l2vpn-pbb-evpn] as well as [I-D.ietf-l2vpn-trill-evpn].

1.1. Terminology

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

1.2. Motivation for running BFD at the network layer of EVPN

The choice of running BFD at the network layer of the OAM model for EVPN [I-D.salam-l2vpn-evpn-oam-req-frmwk] was made after considering the following:

- o In addition to detecting link failures in the EVPN network, BFD sessions at the network layer can be used to monitor the successful programming of labels used for setting up MP2P and P2MP EVPN tunnels transporting Unicast and BUM traffic. The scope of reachability detection covers the ingress and the egress EVPN PE nodes and the network connecting them.
- o Monitoring a representative set of path(s) or a particular path among the multiple paths available between two EVPN PE nodes could be done by exercising the entropy labels when they are used. However paths that cannot be realized by entropy variations cannot be monitored. Fault monitoring requirements outlined by [I-D.salam-l2vpn-evpn-oam-req-frmwk] are addressed by the mechanisms proposed by this draft.

Successful establishment and maintenance of BFD sessions between EVPN PE nodes does not fully guarantee that the EVPN service is functioning. For example, an egress EVPN-PE can understand the EVPN label but could switch data to incorrect interface. However, once BFD sessions in the EVPN Network Layer reach UP state, it does provide additional confidence that data transported using those tunnels will reach the expected egress node. When BFD sessions in the EVPN Network Layer exits UP state, it provides additional confidence that data transported using those tunnels will not reach the expected egress node.

2. Scope of fault detection mechanisms proposed in this document

This section proposes proactive fault detection using BFD mechanisms for:

- a. BUM traffic using MP2P tunnels (ingress replication).
- b. BUM traffic using P2MP tunnels (LSM).
- c. Unicast traffic.

This specification describes procedures only for BFD asynchronous mode. BFD demand mode is outside the scope of this specification. Further, the use of the Echo function is outside the scope of this specification. The approach takes advantage of the inclusive multicast route used in EVPN to advertise the multi-destination FEC for bootstrapping the BFD sessions. Earlier approaches for P2MP BFD [I-D.ietf-mpls-mcast-cv] have used periodic MPLS ping requests to bootstrap P2MP BFD sessions over MPLS.

2.1. Fault Detection of BUM traffic using ingress replication (MP2P)

Ingress replication uses separate MP2P tunnels for transporting BUM traffic from the ingress PE (head) to a set of one or more egress PEs (tails). The fault detection mechanism proposed by this document takes advantage of the fact that a unique copy is made by the head for each tail. Another key aspect to be considered in EVPN is the advertisement of the inclusive multicast route. The BUM traffic flows from a head node to a particular tail only after the head receives the inclusive multicast route containing the BUM EVPN label (downstream allocated) corresponding to the MP2P tunnel. Note that once the BFD session for the EVPN BUM label is UP, either end of the BFD session MUST NOT change the local discriminator values of the BFD Control packets it generates, unless it first brings down the session as specified in RFC 5884 [RFC5884].

2.1.1. Bootstrapping BFD sessions at the head of the MP2P tunnel

To simplify BFD session de-multiplexing, we take advantage of the fact that the head replicates a BUM packet for each tail by using unique sets of discriminators in each copy of the (replicated) BFD packet. These discriminators MUST be exchanged out-of-band using MPLS ping RFC 5884 [RFC5884] before the start of the BFD session between the head and the tail node(s). The head PE performing ingress replication MUST initiate an LSP ping using the inclusive multicast FEC [I-D.jain-l2vpn-evpn-lsp-ping] upon receiving an inclusive multicast route from a tail to bootstrap the BFD session. This MPLS ping MUST include the BFD TLV specified in RFC 5884

[RFC5884]. There could exist multiple BFD sessions between a head of the multi-destination tunnel and an individual tail due to the usage of entropy labels RFC 6790 [RFC6790] for an inclusive multicast FEC. For fine-grained fault detection, a BFD session MAY be bootstrapped to monitor all unique path(s) that can be realized using entropy labels between a head and a given tail. However, the path(s) MUST be monitored using at least one or more number of representative BFD session(s) to satisfy the fault monitoring requirements [I-D.salam-l2vpn-evpn-oam-req-frmwk]. The issue of demultiplexing (separate) BFD sessions established to monitor liveness of the multiple paths of a <MPLS LSP, FEC> has not been fully addressed by [RFC5884]. Procedures proposed in [ID.draft-vgovindan-mpls-extended-bfd-disc-tlv] [1] could be used for monitoring connectivity of the path(s) that are realized through entropy labels. The head node MAY initiate separate BFD sessions using different instance identifiers to verify connectivity of the different paths.

2.1.2. Bootstrapping BFD sessions at the tail nodes of the MP2P tunnel

The tail nodes MUST bootstrap a BFD session based on the incoming MPLS ping initiated by the head [I-D.ietf-bfd-multipoint]. At the tail node, a new BFD discriminator MUST be allocated for each unique combination of the source IP and the attributes of the <inclusive multicast FEC, BUM label> when a MPLS ping initiated from the head is received. A tail node MAY include the instance identifier, if present to support monitoring of specific paths or all realizable paths.

2.2. Fault Detection of BUM traffic using P2MP tunnels (LSM)

The case of using P2MP tunnels for distributing BUM traffic presents a different challenge for using BFD. Clearly, the yourDisc of the BFD packet MUST be zero [I-D.ietf-bfd-multipoint] as the packet is multicast from the root unlike ingress replication where individual copies are made from the head. However the MPLS label that identifies the P-Tunnel [I-D.ietf-l2vpn-evpn] used for forwarding the multi-destination traffic provides a convenient method of identifying the source and the FEC (multi-destination tree) being tracked by the BFD session. The tails of the multi-destination tree MUST use the MPLS label identifying the P-tunnel to de-multiplex the BFD packet. In the case of Aggregate Inclusive trees, where the root of the multi-destination tree reuses the same LSP label for traffic of various EVIs, the tail node MUST use the MPLS labels of the P-Tunnel and the upstream assigned label which the PE has bound uniquely to the EVI. The myDisc of the BFD packet is filled with an unique value allocated by the root to identify the multi-path session.

2.2.1. Bootstrapping BFD sessions at the root of the P2MP tunnel

The P2MP BFD sessions MUST be bootstrapped at the head [I-D.ietf-bfd-multipoint] as soon as there is one receiver for the MDT traffic.

2.2.2. Bootstrapping BFD sessions at the tail nodes of the P2MP tunnel

The P2MP BFD sessions MUST be bootstrapped at the tail upon reception of the P2MP BFD packets from the head. The tail MUST use the P2MP MDT label to de-multiplex the incoming BFD packet. The BFD session MAY be destroyed immediately upon leaving Up state.

2.3. Fault Detection of unicast traffic

The mechanisms specified in BFD for MPLS LSPs RFC 5884 [RFC5884] can be applied to bootstrap and maintain BFD sessions for unicast EVPN traffic. The discriminators required for de-multiplexing the BFD sessions MUST be exchanged using MPLS ping specifying the Unicast EVPN FEC [I-D.jain-l2vpn-evpn-lsp-ping] before starting the BFD session. This is needed since the MPLS label stack does not contain enough information to disambiguate the sender of the packet. The usage of MPLS entropy labels take care of addressing the requirement of monitoring faults of the various paths of the multi-path server layer network RFC 6790 [RFC6790]. Each unique realizable path between the participating PE routers MAY be monitored separately when entropy labels are used. Alternately, all paths MUST be tracked by at least one or a fewer number of representative BFD session(s) in which case the granularity of fault-detection would be coarser. The PE node receiving the MPLS ping MUST allocate one BFD discriminator for every unique combination of the source IP address and the tuple of <unicast FEC, EVPN label>. A node MAY include the instance identifier of the entropy label, if present to satisfy the requirement of fault monitoring of specific paths or all realizable paths. Note that once the BFD session for the EVPN label is UP, either end of the BFD session MUST NOT change the local discriminator values of the BFD Control packets it generates, unless it first brings down the session as specified in RFC 5884 [RFC5884].

3. BFD packet encapsulation

3.1. Using GAL/G-ACh encapsulation without IP headers

3.1.1. Ingress replication

The packet contains the following labels: LSP label (transport) when not using PHP, the optional entropy label, the BUM label and the SH label [I-D.ietf-l2vpn-evpn] (where applicable). The G-ACh type is set

to TBD. The discriminator values of BFD are obtained through negotiation through the out-of-band MPLS ping.

3.1.2. LSM

The packet contains the following labels: label identifying the P-Tunnel, upstream label which the PE has bound uniquely to the EVI (for aggregate inclusive trees only). The G-ACh type is set to TBD. The yourDisc value is set to 0 and the myDisc value is uniquely generated by the root.

3.1.3. Unicast

The packet contains the following labels: LSP label (transport) when not using PHP, the optional entropy label and the EVPN Unicast label. The G-ACh type is set to TBD. The discriminator values of BFD are obtained through negotiation through the out-of-band MPLS ping.

3.2. Using IP headers

The encapsulation option using IP headers will not be suited for EVPN, as using different values in the destination IP address for data and OAM (BFD) packets could cause the BFD packets to follow a different path than that of data packets. Hence this option MUST NOT be used for EVPN.

4. Scalability Considerations

The mechanisms proposed by this draft could affect the packet load on the network and its elements especially when supporting configurations involving a large number of EVIs. The option of slowing down or speeding up BFD timer values can be used by an administrator or a network management entity to maintain the overhead incurred due to fault monitoring at an acceptable level.

5. Security Considerations

This document does not introduce any new security issues, the security considerations defined in RFC 5880 [RFC5880] and [I-D.ietf-bfd-multipoint] apply in this document.

6. IANA Considerations

A new G-Ach Type is requested for for GAL encapsulated BFD as the existing type [RFC5885] specifically applies to PW-ACH encapsulation.

7. Acknowledgments

We thank Nobo Akiya, Tina Lam, Jose Liste, Mudigonda Mallik and Gregory Mirsky for their valuable input, discussions and comments.

8. References

8.1. Normative References

- [I-D.ietf-bfd-multipoint]
Katz, D. and D. Ward, "BFD for Multipoint Networks", draft-ietf-bfd-multipoint-03 (work in progress), February 2014.
- [I-D.jain-l2vpn-evpn-lsp-ping]
Jain, P., Boutros, S., and S. Salam, "LSP-Ping Mechanisms for E-VPN and PBB-EVPN", draft-jain-l2vpn-evpn-lsp-ping-03 (work in progress), June 2014.
- [ID.vgovindan-mpls-extended-bfd-disc-tlv]
Govindan, V. and N. Akiya, "Label Switched Path (LSP) Ping Extended Bidirectional Forwarding Detection (BFD) Discriminator TLV", , July 2014, <<http://tools.ietf.org/html/draft-vgovindan-mpls-extended-bfd-disc-tlv-00>>.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [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.
- [RFC5885] Nadeau, T. and C. Pignataro, "Bidirectional Forwarding Detection (BFD) for the Pseudowire Virtual Circuit Connectivity Verification (VCCV)", RFC 5885, June 2010.

8.2. Informative References

- [I-D.ietf-l2vpn-evpn]
Sajassi, A., Aggarwal, R., Bitar, N., Isaac, A., and J. Uttaro, "BGP MPLS Based Ethernet VPN", draft-ietf-l2vpn-evpn-07 (work in progress), May 2014.

- [I-D.ietf-l2vpn-pbb-evpn]
Sajassi, A., Salam, S., Bitar, N., Isaac, A., Henderickx, W., and L. Jin, "PBB-EVPN", draft-ietf-l2vpn-pbb-evpn-07 (work in progress), June 2014.
- [I-D.ietf-l2vpn-trill-evpn]
Sajassi, A., Salam, S., Bitar, N., and S. Aldrin, "TRILL-EVPN", draft-ietf-l2vpn-trill-evpn-01 (work in progress), October 2013.
- [I-D.ietf-mpls-mcast-cv]
Swallow, G., "Connectivity Verification for Multicast Label Switched Paths", draft-ietf-mpls-mcast-cv-00 (work in progress), April 2007.
- [I-D.salam-l2vpn-evpn-oam-req-frmwk]
Salam, S., Sajassi, A., Aldrin, S., and J. Drake, "E-VPN Operations, Administration and Maintenance Requirements and Framework", draft-salam-l2vpn-evpn-oam-req-frmwk-02 (work in progress), January 2014.
- [RFC6790] Kompella, K., Drake, J., Amante, S., Henderickx, W., and L. Yong, "The Use of Entropy Labels in MPLS Forwarding", RFC 6790, November 2012.

8.3. URIs

- [1] <http://tools.ietf.org/html/draft-vgovindan-mpls-extended-bfd-disc-tlv-00>

Authors' Addresses

Vengada Prasad Govindan
Cisco Systems

Email: venggovi@cisco.com

Samer Salam
Cisco Systems

Email: ssalam@cisco.com

Ali Sajassi
Cisco Systems

Email: sajassi@cisco.com

Internet Engineering Task Force
Internet-Draft
Intended status: Standards Track
Expires: January 5, 2015

V. Govindan
N. Akiya
Cisco Systems
July 4, 2014

Label Switched Path (LSP) Ping
Extended Bidirectional Forwarding Detection (BFD) Discriminator TLV
draft-vgovindan-mpls-extended-bfd-disc-tlv-00

Abstract

This document defines an extended Bidirectional Forwarding Detection (BFD) discriminator TLV for the Multiprotocol Label Switching (MPLS) Label Switched Path (LSP) Ping mechanism, to allow bootstrapping of multiple BFD sessions for a given FEC.

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 January 5, 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
2. Overview	3
3. Procedures for BFD session establishment and removal using the Extended BFD TLV	3
3.1. Procedures for establishing BFD sessions	3
3.2. Procedures for removing BFD sessions	3
4. Extended BFD Discriminator TLV	4
5. Mutually Exclusive: BFD TLVs	5
6. Backwards Compatibility	5
7. Encapsulation	5
8. Security Considerations	5
9. IANA Considerations	6
9.1. Extended BFD Discriminator TLV	6
10. Acknowledgements	6
11. Contributing Authors	6
12. Normative References	6
Appendix A. Alternate format for the BFD Extended TLV	6
Authors' Addresses	8

1. Background

Bidirectional Forwarding Detection (BFD) [RFC5880] for Multiprotocol Label Switching (MPLS) Label Switched Paths (LSPs), [RFC5884], describes a mechanism to use BFD to monitor the connectivity in-band on the LSPs. The BFD session on the LSP egress is bootstrapped using the LSP Ping mechanism, defined in [RFC4379], carrying the BFD Discriminator TLV that describes the BFD discriminator of the BFD session on the LSP ingress.

The BFD Discriminator TLV and defined procedures around this TLV only allow one BFD session to be bootstrapped per <MPLS Forwarding Equivalent Class (FEC), LSP>. There are scenarios where an LSP ingress may desire to run multiple BFD sessions to monitor the connectivity on an LSP. To achieve the bootstrapping of multiple BFD sessions per FEC, a new TLV and procedures are required. Two scenarios where this is useful are described below:

- o Entropy labels help achieve load balancing of traffic belonging to the same <MPLS FEC, LSP>. It may be beneficial to track the

individual paths of the multi-path network using separate BFD sessions for each non-congruent path.

- o It may be useful to establish multiple BFD sessions for the same <MPLS FEC, LSP> to achieve BFD session redundancy, i.e. protection against false positives due to equipment or soft failures inside boxes.

2. Overview

An LSR ingress wanting to bootstrap one or more BFD sessions on an LSP is to include the Extended BFD Discriminator TLV, described in Section 4, in the MPLS echo request message for the FEC. The Extended BFD Discriminator TLV is capable of carrying multiple BFD discriminators, and each BFD discriminator is accompanied with an instance identifier. The LSR egress, upon reception of this MPLS echo request, is to create requested number of BFD sessions for the specified FEC. Each BFD session object created on the LSR ingress and the LSR egress MUST be annotated with corresponding instance identifier. BFD session procedures are to follow those described in [RFC5884].

3. Procedures for BFD session establishment and removal using the Extended BFD TLV

3.1. Procedures for establishing BFD sessions

There are at least two options possible here:

1. BFD session establishment MUST follow the procedure specified in [RFC5884].
2. The base procedure for BFD session establishment MUST be the same as that of [RFC5884]. This procedure can be enhanced by specifying additional Operation type field and Operation status field in the proposed Extended BFD Discriminator TLV. See Appendix A for a description of Operation types and Operation status codes.

3.2. Procedures for removing BFD sessions

[RFC5884] does not specify an explicit procedure for deleting BFD sessions. A few options are possible here:

1. Specify an explicit delete procedure for the BFD session using Operation types field and Operation status field through the Extended BFD TLV. See Appendix A for a description of Operation types and Operation status codes.

2. Specify a timer based deletion procedure: A new purge timer field can be introduced within the proposed Extended BFD Discriminator TLV. The ingress specifies the value for the purge timer field. Once the BFD session transitions from up to down state, the egress is to delete the session after the value specified in the purge timer field. Ed Note: This approach is an open topic for discussion.
3. No new procedure to delete a BFD session is introduced. Assumption by the egress is that BFD sessions can be deleted if corresponding FEC is deleted from the system or sometime after BFD sessions go down.

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

4. Extended BFD Discriminator TLV

The Extended BFD Discriminator object is a new TLV that MAY be included in the MPLS echo request message. An MPLS echo request MUST NOT include more than one Extended BFD Discriminator object. The Extended BFD Discriminator object describes one or more BFD discriminators along with each having an instance identifier. An MPLS echo reply MAY include the Extended BFD Discriminator object, but MUST NOT include more than one Extended BFD Discriminator object.

Extended BFD Discriminator TLV Type is TBD1. Length is 8 or multiples of 8. Length of (8 x N) implies that there are N entries in the Value field of the Extended BFD Discriminator TLV. Each entry in the Value field of the Extended BFD Discriminator TLV has following format:

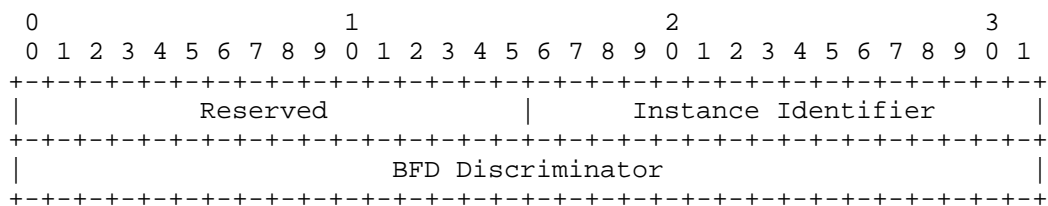


Figure 1: Extended BFD Discriminator TLV

Reserved - This field MUST be set to zero on transmit, and ignored on receipt.

Instance Identifier - An instance identifier of the BFD session. The instance identifier is a value allocated by the LSP ingress

for corresponding BFD Discriminator, and MUST be unique within the FEC on the LSP ingress node. The instance identifier MUST NOT change for the lifetime of the BFD session.

BFD Discriminator - The BFD discriminator allocated for this BFD session by the LSP ingress.

See Appendix A for a discussion on an alternate format for the TLV.

5. Mutually Exclusive: BFD TLVs

The BFD Discriminator TLV and the Extended BFD Discriminator TLV are mutually exclusive. An MPLS echo request/reply message MUST NOT include both the BFD Discriminator TLV and the Extended BFD Discriminator TLV. Reception of an MPLS echo request with both the BFD Discriminator TLV and the Extended BFD Discriminator TLV is to result in the Return Code being set to Malformed echo request received (1).

6. Backwards Compatibility

If an LSP ingress wishes to bootstrap multiple BFD sessions with the Extended BFD Discriminator TLV when an LSP already has a BFD session bootstrapped with the BFD Discriminator TLV, following procedures are RECOMMENDED.

The LSP ingress is to send an MPLS echo request carrying the Extended BFD Discriminator TLV with the same BFD discriminator of the existing BFD session (one bootstrapped previously with the BFD Discriminator TLV), giving it an instance identifier. Once the transition of the existing BFD session is completed, then the LSP ingress can generate further MPLS echo request messages with the Extended BFD Discriminator TLV to bootstrap more BFD sessions.

7. Encapsulation

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

8. Security Considerations

This document defines a mechanism to bootstrap multiple BFD sessions per FEC. 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.

9. IANA Considerations

9.1. Extended BFD Discriminator TLV

The IANA is requested to assign new value TBD1 for Extended BFD Discriminator TLV from the "Multiprotocol Label Switching Architecture (MPLS) Label Switched Paths (LSPs) Ping Parameters - TLVs" registry.

Value	Meaning	Reference
-----	-----	-----
TBD1	Extended BFD Discriminator TLV	this document

10. Acknowledgements

TBD

11. Contributing Authors

Girija Rao
Cisco Systems
Email: giraghav@cisco.com

Mallik Mudigonda
Cisco Systems
Email: mmudigon@cisco.com

12. 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.

Appendix A. Alternate format for the BFD Extended TLV

The BFD Extended TLV can be used to carry the Operation Type and the Operation Status (Op Status) bits that are defined below:

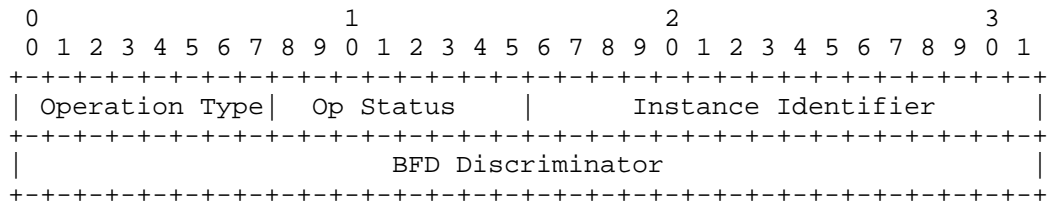


Figure 2: Alternate format of the Extended
BFD Discriminator TLV

Ed Note: The definitions of Operation Type and Operation Status fields are subject to discussion. Additional codes can be defined if this approach is pursued.

Operation Type - Operation to be performed on the corresponding BFD Discriminator. Valid values are:

- 1 - Create: This value MAY be used in the MPLS echo request, but MUST NOT be used in the MPLS echo reply. The operation type 1 indicates that receiver (i.e. LSP egress) is to ensure that BFD session for this FEC, with corresponding BFD Discriminator in "your discriminator" field, exists or is created.
- 2 - Delete: This value MAY be used in the MPLS echo request, but MUST NOT be used in the MPLS echo reply. The operation type 2 indicates that receiver (i.e. LSP egress) is to ensure that BFD session for this FEC, with corresponding BFD Discriminator in "your discriminator" field, does not exist or is deleted.
- 3 - CreateAck: This value MUST NOT be used in the MPLS echo request, but MAY be used in the MPLS echo reply. The operation type 3 indicates that receiver (i.e. LSP egress) is acknowledging received Create(1) request.
- 4 - DeleteAck: This value MUST NOT be used in the MPLS echo request, but MAY be used in the MPLS echo reply. The operation type 4 indicates that receiver (i.e. LSP egress) is acknowledging received Delete(2) request.

Op Status

- 0 - The operation succeeded.
- 1 - Not enough Resources.

BFD Discriminator - When the Extended BFD Discriminator TLV is carried in the MPLS echo request, this field describes the BFD discriminator allocated for this BFD session by the LSP ingress. When the Extended BFD Discriminator TLV is carried in the MPLS echo reply, this field describes the BFD discriminator allocated for this BFD session by the LSP egress.

The Extended BFD Discriminator TLV in an MPLS echo request MUST have either Create(1) or Delete(2) operation type. The Extended BFD Discriminator TLV in an MPLS echo reply MUST have either CreateAck(3) or DeleteACK(4) operation type.

Authors' Addresses

Vengada Prasad Govindan
Cisco Systems

Email: venggovi@cisco.com

Nobo Akiya
Cisco Systems

Email: nobo@cisco.com

INTERNET-DRAFT
Intended Status: Informational

Mingui Zhang
Zuliang Wang
Mach Chen
Huawei
July 4, 2014

Expires: January 5, 2015

Use Cases Requiring New Features of BFD
draft-zhang-bfd-new-use-cases-00.txt

Abstract

This document describes some use cases arising from the deployment of BFD. These use cases are expected by ISPs but not supported by current BFD yet. Requirements are developed on basis of these use cases so that they can be taken into consideration in the future update of the BFD protocol.

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>

Copyright and License 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. Introduction	3
2. Acronyms and Terminology	3
2.1. Acronyms	3
2.2. Terminology	3
3. Use Cases	3
3.1. Use Case #1: Reliable Detection for Multipath	3
3.2. Use Case #2: Application Consistence	4
3.3. Use Case #3: Capability Inquiry and Feedback	5
3.4. Use Case #4: State Relay	6
3.5. Use Case #5: Detection of Asymmetric LSPs	7
4. Security Considerations	7
5. IANA Considerations	7
Acknowledgements	7
6. References	7
6.1. Normative References	7
6.2. Informative References	8
Author's Addresses	9

1. Introduction

BFD is able to detect a network fault with a very low latency. It is designed to be independent of any media, data protocols, and routing protocols [RFC5880]. Today, it has been widely deployed in ISPs' networks and used by various applications.

Requirements for those BFD core use cases used to be generally fulfilled. However, there are also some use cases that do not fit current BFD. This document reveals five use cases arising from the real deployment of BFD but not supported yet. From these use cases, some basic requirements are extracted to be considered in the future when BFD is to be updated. This document aims to provide some information for the discussion on whether these use cases can be handled with a smooth update of the BFD protocol.

2. Acronyms and Terminology

2.1. Acronyms

BFD: Bidirectional Forwarding Detection

2.2. Terminology

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

Familiarity with [RFC5880] is assumed in this document.

3. Use Cases

3.1. Use Case #1: Reliable Detection for Multipath

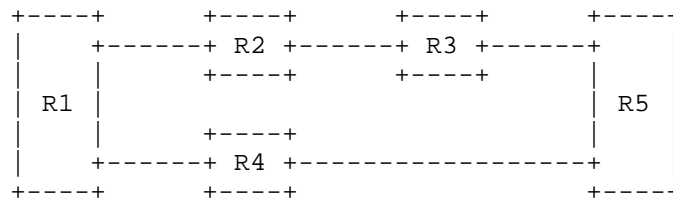


Figure 3.1. An example topology of BFD for OSPF ECMP

Carrier networks widely adopt multipath techniques between two endpoints for the purpose of higher bandwidth and resilience, such as ECMP in OSPF/ISIS, Ethernet Link Aggregation (LAG), and MPLS Link Bundling [RFC4201]. If BFD is deployed on network devices, the


```

+-----+ IGP/PIM/RSVP      IGP/PIM/RSVP +-----+
| R1 +-----+-----+ R2 |
+-----+                               +-----+

```

Figure 3.3. Client applications of the BFD are inconsistent

Endpoint subscribes the BFD detection locally. If the two endpoints subscribing one BFD session with different applications while different applications claim different detection requirements, the BFD may malfunction. Take Figure 3.3 as an example, the pair of interface cards on R1 and R2 are multiply configured with IGP/PIM/RSVP. Assume IGP requires a transmit interval of 10 milliseconds and a detection multiplier of 3 while PIM requires a transmit interval of 100 milliseconds and a multiplier of 5. Finally, the BFD session may achieve a detection time of 500 milliseconds.

The two endpoints are required to negotiate the detection requirements of the applications subscribing the same BFD session. If these requirements are inconsistent, the BFD session SHOULD not be established before the inconsistency is resolved.

3.3. Use Case #3: Capability Inquiry and Feedback

If the local system restarts, it may resume the BFD session. Suppose the link has been failed or the peer has no resources to create the BFD session or the peer had been taken down administratively during the absence of the local system. Since no BFD control packets will be received from the peer system, the BFD will report a Down state. Rather, the real state of the forwarding path can either be Down or Up.

```

+-----+      capability inquiry->      +-----+
| R1 +-----+-----+ R2 |
+-----+      <-able/unable/no response +-----+

```

Figure 3.4. BFD capability inquiry and feedback

The local system is required to inquire the peer's BFD capability when the BFD session is resumed after the system reboots. The peer is required to feedback whether the BFD is able to be created as required. If the peer can establish the BFD session as required, the remote system MUST send a BFD Control packet in the detection time with the State field set to anything other than Up. This is shown in Figure 3.4.

- o If the peer cannot establish the BFD session because it does not support the detection as required or it does not have the resource anymore to establish the BFD session or the BFD has been taken down

administratively, the peer MUST feedback it is unable to establish the session. If the feedback is received, the BFD MUST not report a Down state of the forwarding path. It's up to the application to determine the state of the forwarding path.

- o If no feedback is received from the peer in the detection time, the BFD will continue to report to the application that the forwarding path is in Down state.

It's required that the above update is supported by both peering systems. In other words, this update is not backward compatible.

3.4. Use Case #4: State Relay

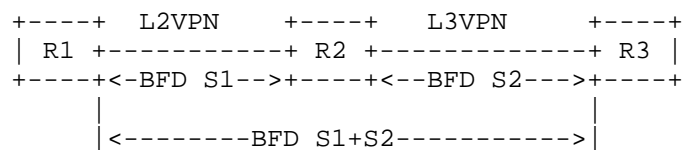


Figure 3.5. BFD session concatenation

End to end forwarding paths mixed with L2VPN and L3VPN tunnels are widely adopted in IPRAN. As shown in Figure 3.5, the tunnel between R1 and R2 and the tunnel between R2 and R3 are connected end-to-end in series. These three endpoints can establish two separate BFD sessions to detect the whole forwarding path. However, it's impossible for R1 and R3 to establish a single BFD session between each other.

When the L2VPN tunnel fails, R1 and R2 are disconnected but R3 is unaware of the failure. R2 has to resort to the control plane of L3VPN to disseminate the failure. For example, R2 can withdraw the VPN route through BGP [RFC4364]. This will trigger the reconvergence of L3VPN. Usually, the reconvergence is slow and traffic being sent from R3 to R1 will suffer from a long time period of interruption.

Section 6.8.17 of [RFC5880] provides the Concatenated Paths mechanism. R2 can propagate the state of the BFD session S1 to S2 through the diagnostic code. However, the indication of the failure requires the participation of the interworking system R2, which may become a heavy overhead when lots of paths need be concatenated. While this happens often in IPRAN.

In this use case, carriers expect the state change of BFD session S1 is relayed to R3 without the participation of the interworking system R2. R3 can immediately sense that R1 is not reachable and stop sending traffic to an obvious black-hole. It's also required that the

relations of the concatenation paths are relayed to R3 by R2 as well. In other words, R2 need transmit the correspondence (mapping) between the concatenated BFD sessions to R3 through the BFD control packet.

3.5. Use Case #5: Detection of Asymmetric LSPs

A bidirectional LSP is probably adopting different forwarding paths for different directions. Suppose the ingress LSR set up the BFD session with Echo function enabled. When the echo packets are looped back, the other system chooses the forwarding path by default according to the IP forwarding path. If this forwarding path is different to the reverse forwarding path of the LSP, the BFD detection will be inaccurate.

The ingress LSR should be able to advertise in the BFD control packets whether the LSP reverse forwarding path should be used as the forwarding path for echo packets. If the ingress LSR is requiring the LSP reverse path as the forwarding path for echo packets, the egress LSR MUST loop back the echo packets according to the reverse path rather than the default IP forwarding path.

4. Security Considerations

This document raises no new security issues.

5. IANA Considerations

This document requires no IANA actions. RFC Editor: please remove this section before publication.

Acknowledgements

Authors would like to thank the comments and suggestions from Marc Binderberger and Xudong Zhang.

6. References

6.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC4201] Kompella, K., Rekhter, Y., and L. Berger, "Link Bundling in MPLS Traffic Engineering (TE)", RFC 4201, October 2005.
- [RFC5880] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD)", RFC 5880, June 2010.

- [RFC5881] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD) for IPv4 and IPv6 (Single Hop)", RFC 5881, June 2010.
- [RFC5882] Katz, D. and D. Ward, "Generic Application of Bidirectional Forwarding Detection (BFD)", RFC 5882, June 2010.
- [RFC7130] M. Bhatia, Ed., M. Chen, Ed., S. Boutros, Ed., M. Binderberger, Ed., J. Haas, Ed., "Bidirectional Forwarding Detection (BFD) on Link Aggregation Group (LAG) Interfaces", RFC 7130, February 2014.

6.2. Informative References

- [RFC5883] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD) for Multihop Paths", RFC 5883, June 2010.

Author's Addresses

Mingui Zhang
Huawei Technologies
No. 156 Beiqing Rd. Haidian District,
Beijing 100095
P.R. China

Email: zhangmingui@huawei.com

Zuliang Wang
Huawei Technologies
No. 156 Beiqing Rd. Haidian District,
Beijing 100095
P.R. China

Email: zuni.wang@huawei.com

Mach(Guoyi) Chen
Huawei Technologies
No. 156 Beiqing Rd. Haidian District,
Beijing 100095
P.R. China

EMail: mach@huawei.com