Internet Engineering Task Force (IETF)
Request for Comments: 8562
Updates: 5880
Category: Standards Track
ISSN: 2070-1721

D. Katz
Juniper Networks
D. Ward
Cisco Systems
S. Pallagatti, Ed.
VMware
G. Mirsky, Ed.
ZTE Corp.

April 2019

Bidirectional Forwarding Detection (BFD) for Multipoint Networks

Abstract

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

This document updates RFC 5880.

Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 7841.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at https://www.rfc-editor.org/info/rfc8562.

Copyright Notice

Table of Contents

1. Introduction

The Bidirectional Forwarding Detection (BFD) protocol [RFC5880] specifies a method for verifying unicast connectivity between a pair of systems. This document updates [RFC5880] by defining a new method for using BFD. This new method provides verification of multipoint or multicast connectivity between a multipoint sender (the "head") and a set of one or more multipoint receivers (the "tails").

As multipoint transmissions are inherently unidirectional, this mechanism purports only to verify this unidirectional connectivity. Although this seems in conflict with the "Bidirectional" in BFD, the protocol is capable of supporting this use case. Use of BFD in Demand mode allows a tail to monitor the availability of a multipoint path even without the existence of some kind of a return path to the head. As an option, if a return path from a tail to the head exists, the tail may notify the head of the lack of multipoint connectivity. Details of tail notification to the head are outside the scope of this document and are discussed in [RFC8563].

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

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

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

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

2. Keywords

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

3. Goals

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

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

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

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

4. Overview

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

The head of a multipoint BFD session may wish to be alerted to the tails' connectivity (or lack thereof). Details of how the head keeps track of tails and how tails alert their connectivity to the head are outside the scope of this document and are discussed in [RFC8563].

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

5. Protocol Details

This section describes the operation of Multipoint BFD in detail.

5.1. Multipoint BFD Control Packets

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

5.2. Session Model

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

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

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

5.3. Session-Failure Semantics

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

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

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

5.4. State Variables

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

5.4.1. New State Variable Values

A number of new values of the state variable bfd.SessionType are added to the base BFD [RFC5880] and base Seamless Bidirectional Forwarding Detection (S-BFD) [RFC7880] specifications in support of multipoint BFD.

bfd.SessionType

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

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

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

MultipointTail: A multipoint session on a tail.

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

5.4.2. State Variable Initialization and Maintenance

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

bfd.RequiredMinRxInterval

This variable MUST be initialized to zero for session type MultipointHead.

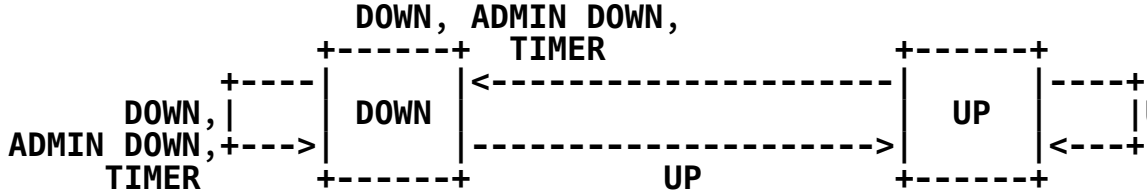
bfd.DemandMode

This variable MUST be initialized to 1 for session type MultipointHead and MUST be initialized to zero for session type MultipointTail.

5.5. State Machine

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

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



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

5.6. Session Establishment

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

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

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

5.7. Discriminators and Packet Demultiplexing

The use of discriminators is somewhat different in multipoint BFD than in point-to-point BFD.

The head sends multipoint BFD Control packets over the multipoint path via the MultipointHead session with My Discriminator set to a value bound to the multipoint path and with Your Discriminator set to zero.

IP and MPLS multipoint tails MUST demultiplex BFD packets based on a combination of the source address, My Discriminator, and the identity of the multipoint path that the multipoint BFD Control packet was received from. Together they uniquely identify the head of the

multipoint path. Bootstrapping a BFD session to multipoint MPLS Label Switched Path (LSP) may use the control plane, e.g., as described in [MVPN-FAILOVER], and is outside the scope of this document.

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

5.8. Packet Consumption on Tails

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

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

5.9. Bringing Up and Shutting Down Multipoint BFD Service

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

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

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

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

5.10. Timer Manipulation

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

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

Change in the value of bfd.RequiredMinRxInterval is outside the scope of this document and is discussed in [RFC8563].

5.11. Detection Times

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

Since MultipointHead sessions never receive packets, they do not calculate a Detection Time.

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

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

5.12. State Maintenance for Down/AdminDown Sessions

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

5.12.1. MultipointHead Sessions

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

5.12.2. MultipointTail Sessions

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

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

5.13. Base Specification Text Replacement

The following sections are meant to replace the corresponding sections in the base specification [RFC5880] to support BFD for multipoint networks while not changing processing for point-to-point BFD.

5.13.1. Reception of BFD Control Packets

The following procedure replaces Section 6.8.6 of [RFC5880] entirely.

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

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

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

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

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

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

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

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

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

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

Set bfd.RemoteDiscr to the value of My Discriminator.

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

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

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

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

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

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

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

      Else

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

      If bfd.SessionState is AdminDown

         Discard the packet

      If the received State is AdminDown

         If bfd.SessionState is not Down

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

            Set bfd.SessionState to Down

      Else

         If bfd.SessionState is Down

            If bfd.SessionType is PointToPoint

               If received State is Down

                  Set bfd.SessionState to Init

               Else if received State is Init

                  Set bfd.SessionState to Up
```

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

               If received State is Up

                  Set bfd.SessionState to Up

         Else if bfd.SessionState is Init

            If received State is Init or Up

               Set bfd.SessionState to Up

         Else (bfd.SessionState is Up)

            If received State is Down

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

               Set bfd.SessionState to Down
```

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

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

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

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

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

5.13.2. Demultiplexing BFD Control Packets

This section is part of the replacement for Section 6.8.6 of [RFC5880]; it is separated for clarity.

If the Multipoint (M) bit is set

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

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

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

Else

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

Else (Multipoint (M) bit is clear)

If the Your Discriminator field is nonzero

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

Else (Your Discriminator is zero)

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

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

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

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

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

5.13.3. Transmitting BFD Control Packets

The following procedure replaces Section 6.8.7 of [RFC5880] entirely.

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

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

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

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

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

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

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

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

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

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

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

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

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

Version

Set to the current version number (1).

Diagnostic (Diag)

Set to bfd.LocalDiag.

State (Sta)

Set to the value indicated by bfd.SessionState.

Poll (P)

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

Final (F)

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

Control Plane Independent (C)

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

Authentication Present (A)

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

Demand (D)

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

Multipoint (M)

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

Detect Mult

Set to bfd.DetectMult.

Length

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

My Discriminator

Set to bfd.LocalDiscr.

Your Discriminator

Set to bfd.RemoteDiscr.

Desired Min TX Interval

Set to bfd.DesiredMinTxInterval.

Required Min RX Interval

Set to bfd.RequiredMinRxInterval.

Required Min Echo RX Interval

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

Authentication Section

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

6. Congestion Considerations

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

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

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

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

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

Additional considerations apply to BFD in multipoint networks, as specified in this document. Indeed, because a tail does not transmit any BFD Control packets to the head of the BFD session, such a head node has no BFD-based mechanism and thus is not aware of the state of the session at the tail. In the absence of any other mechanism, the head of the session could thus continue to send packets towards the tail(s) even though a link failure has happened. In such a scenario, when it is required for the head of the session to be aware of the state of the tail of the session, it is RECOMMENDED to implement the extension described in [RFC8563].

7. IANA Considerations

This document has no IANA actions.

8. Security Considerations

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

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

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

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

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

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

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

9. References

9.1. Normative References

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

[RFC5880] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD)", RFC 5880, DOI 10.17487/RFC5880, June 2010, <https://www.rfc-editor.org/info/rfc5880>.

[RFC7880] Pignataro, C., Ward, D., Akiya, N., Bhatia, M., and S. Pallagatti, "Seamless Bidirectional Forwarding Detection (S-BFD)", RFC 7880, DOI 10.17487/RFC7880, July 2016, <https://www.rfc-editor.org/info/rfc7880>.

[RFC8029] Kompella, K., Swallow, G., Pignataro, C., Ed., Kumar, N., Aldrin, S., and M. Chen, "Detecting Multiprotocol Label Switched (MPLS) Data-Plane Failures", RFC 8029, DOI 10.17487/RFC8029, March 2017, <https://www.rfc-editor.org/info/rfc8029>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

9.2. Informative References

[MVPN-FAILOVER]
Morin, T., Ed., Kebler, R., Ed., and G. Mirsky, Ed., "Multicast VPN fast upstream failover", Work in Progress, draft-ietf-bess-mvpn-fast-failover-05, February 2019.

[RFC4082] Perrig, A., Song, D., Canetti, R., Tygar, J., and B. Briscoe, "Timed Efficient Stream Loss-Tolerant Authentication (TESLA): Multicast Source Authentication Transform Introduction", RFC 4082, DOI 10.17487/RFC4082, June 2005, <https://www.rfc-editor.org/info/rfc4082>.

[RFC5883] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD) for Multihop Paths", RFC 5883, DOI 10.17487/RFC5883, June 2010, <https://www.rfc-editor.org/info/rfc5883>.

[RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <https://www.rfc-editor.org/info/rfc6241>.

[RFC8563] Katz, D., Ward, D., Pallagatti, S., Ed., and G. Mirsky, Ed., "Bidirectional Forwarding Detection (BFD) Multipoint Active Tails", RFC 8563, DOI 10.17487/RFC8563, April 2019, <https://www.rfc-editor.org/info/rfc8563>.

Acknowledgments

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

Contributors

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

Authors' Addresses

Dave Katz
Juniper Networks
1194 N. Mathilda Ave.
Sunnyvale, California 94089-1206
United States of America

Email: dkatz@juniper.net

Dave Ward
Cisco Systems
170 West Tasman Dr.
San Jose, California 95134
United States of America

Email: wardd@cisco.com

Santosh Pallagatti (editor)
VMware

Email: santosh.pallagatti@gmail.com

Greg Mirsky (editor)
ZTE Corp.

Email: gregimirsky@gmail.com