Internet Engineering Task Force
Internet-Draft
Updates: XXXX (RFC BFD for Multi-point Networks) (if approved)
Intended status: Standards Track
Expires: December 6, 2018

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

June 4, 2018

BFD Multipoint Active Tails. draft-ietf-bfd-multipoint-active-tail-08

Abstract

This document describes active tail extensions to and updates the Bidirectional Forwarding Detection (BFD) protocol for multipoint networks.

Status of This Memo

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

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

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

This Internet-Draft will expire on December 6, 2018.

Copyright Notice

Table of Contents

1. Introduction

This application of BFD is an extension to Multipoint BFD [I-D.ietf-bfd-multipoint], which allows tails to notify the head of the lack of multipoint connectivity. As a further option, heads can request a notification from the tails by means of a polling mechanism. Notification to the head can be enabled for all tails, or for only a subset of the tails.

The goal of this application is for the head to reasonably rapidly have knowledge of tails that have lost connectivity from the head.

Since scaling is a primary concern (particularly state explosion toward the head), it is required that the head be in control of all timing aspects of the mechanism, and that BFD packets from the tails to the head not be synchronized.

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.

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

This document effectively modifies and adds to Sections 5.12 and 5.13 of the base BFD multipoint document [I-D.ietf-bfd-multipoint].

2. Terminology and Acronyms

BFD Bidirectional Forwarding Detection

c-poll Composite Poll

m-poll Multipoint Poll

3. 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.

4. Overview

A head may wish to be alerted to the tails' connectivity (or lack thereof), there are a number of options. First, if all that is needed is a best-effort failure notification, as discussed in Section 5.2.1, the tails can send unsolicited unicast BFD Control packets to the head when the path fails, as described in Section 6.4.

If the head wishes to know of the active tails on the multipoint path, it may send a multipoint BFD Control packet with the Poll (P) bit set, which will induce the tails to return a unicast BFD Control packet with the Final (F) bit set (detailed description in Section 5.2.2). The head can then create BFD session state for each of the tails that have multipoint connectivity. If the head sends such a packet on occasion, it can keep track of which tails answer, thus providing a more deterministic mechanism for detecting which tails fail to respond (implying a loss of multipoint connectivity). In this document, this method referenced to as Multipoint Poll (m-poll).

If the head wishes the definite indication of the tails' connectivity, it may do all of the above, but if it detects that a tail did not answer the previous multipoint poll, it may initiate a Demand mode Poll Sequence as a unicast to that tail (detailed description in Section 5.2.3). This covers the case where either the multipoint poll or the single reply also is lost in transit. If desired, the head may Poll one or more tails proactively to track the tails' connectivity. In this document this method that combines the use of multipoint and unicast polling of tails by the head referenced to as Composite Poll (c-poll).

If the awareness of the state of some nodes is more important for the head, in the sense that the head needs to detect the lack of multipoint connectivity to a subset of tails at a different rate, the head may transmit unicast BFD Polls to that subset of tails. In this case, the timing may be independent on a tail-by-tail basis.

Individual tails may be configured so that they never send BFD control packets to the head. Such tails will never be known to the head, but will still be able to detect multipoint path failures from the head.

5. Operational Scenarios

It is worth analyzing how this protocol reacts to various scenarios. There are three path components present, namely, the multipoint path, the forward unicast path (from head to a particular tail), and the reverse unicast path (from a tail to the head). There are also four

options as to how the head is notified about failures from the tail. For the different modes described below the setting of new state variables are given even if these are only introduced later in the document (see Section 6.3).

5.1. No Head Notification

In this scenario, only the multipoint path is used and none of the others matter. A failure in the multipoint path will result in the tail noticing the failure within a detection time, and the head will remain ignorant of the tail state. This mode emulates the behavior described in [I-D.ietf-bfd-multipoint]. In this mode for bfd.SessionType is MultipointTail, variable bfd.SilentTail (see Section 6.3.1) MUST be set to 1. If bfd.SessionType is MultipointHead or MultipointClient bfd.ReportTailDown MUST be set to 0. The head MAY set bfd.RequiredMinRxInterval to zero and thus suppress tails sending any BFD control packets.

5.2. Head Notification

In these scenarios, the tail sends unsolicited or solicited BFD packets in response to the detection of a multipoint path failure. All these scenarios have common settings:

- if bfd.SessionType is MultipointTail, variable bfd.SilentTail (see Section 6.3.1) MUST be set to 0;
- if bfd.SessionType is MultipointHead or MultipointClient bfd.ReportTailDown MUST be set to 1;
- the head MUST set bfd.RequiredMinRxInterval to non-zero and thus allow tails sending BFD control packets.

5.2.1. Head Notification Without Polling

In this scenario, the tail sends unsolicited BFD packets in response to the detection of a multipoint path failure. It uses the reverse unicast path, but not the forward unicast path.

If the multipoint path fails but the reverse unicast path stays up, the tail will detect the failure within a detection time, and the head will know about it within one reverse packet time (since the notification is delayed).

If both the multipoint path and the reverse unicast paths fail, the tail will detect the failure but the head will remain unaware of it.

5.2.2. Head Notification and Tail Solicitation with Multipoint Polling

In this scenario, the head sends occasional multipoint Polls in addition to (or in lieu of) non-Poll multipoint BFD Control packets, expecting the tails to reply with Final. This also uses the reverse unicast path, but not the forward unicast path.

If the multipoint path fails but the reverse unicast path stays up, the tail will detect the failure within a detection time, and the head will know about it within one reverse packet time (the notification is delayed to avoid synchronization of the tails).

If both the multipoint path and the reverse unicast paths fail, the tail will detect the failure but the head will remain unaware of this fact.

If the reverse unicast path fails but the multipoint path stays up, the head will see the BFD session fail, but the state of the multipoint path will be unknown to the head. The tail will continue to receive multipoint data traffic.

If either the multipoint Poll or the unicast reply is lost in transit, the head will see the BFD session fail, but the state of the multipoint path will be unknown to the head. The tail will continue to receive multipoint data traffic.

5.2.3. Head Notification with Composite Polling

In this scenario, the head sends occasional multipoint Polls in addition to (or in lieu of) non-Poll multipoint BFD control packets, expecting the tails to reply with Final. If a tail that had previously replied to a multipoint Poll fails to reply (or if the head simply wishes to verify tail connectivity), the head issues a unicast Poll Sequence to the tail. This scenario makes use of all three paths. In this mode for bfd.SessionType of MultipointTail, variable bfd.SilentTail (see Section 6.3.1) MUST be set to 0.

If the multipoint path fails but the two unicast paths stay up, the tail will detect the failure within a detection time, and the head will know about it within one reverse packet time (since the notification is delayed). Note that the reverse packet time may be smaller in this case if the head has previously issued a unicast Poll (since the tail will not delay transmission of the notification in this case).

If both the multipoint path and the reverse unicast paths fail (regardless of the state of the forward unicast path), the tail will detect the failure but the head will remain unaware of this fact.

The head will detect a BFD session failure to the tail but cannot make a determination about the state of the tail's multipoint connectivity.

If the forward unicast path fails but the reverse unicast path stays up, the head will detect a BFD session failure to the tail if it happens to send a unicast Poll sequence, but cannot make a determination about the state of the tail's multipoint connectivity. If the multipoint path to the tail fails prior to any unicast Poll being sent, the tail will detect the failure within a detection time, and the head will know about it within one reverse packet time (since the notification is delayed).

If the multipoint path stays up but the reverse unicast path fails, the head will see the particular MultipointClient session fail if it happens to send a Poll Sequence, but the state of the multipoint path will be unknown to the head. The tail will continue to receive multipoint data traffic.

If the multipoint path and the reverse unicast path both stay up but the forward unicast path fails, neither side will notice so long as a unicast Poll Sequence is never sent by the head. If the head sends a unicast Poll Sequence, the head will see the BFD session fail, but the state of the multipoint path will be unknown to the head. The tail will continue to receive multipoint data traffic.

6. Protocol Details

This section describes the operation of BFD Multipoint active tail in detail. This section updates the section 4 of [I-D.ietf-bfd-multipoint] as the following:

- Section 6.3 introduces new state variables and modifies the usage of a few existing ones;
- Section 6.13 replaces the corresponding sections in the base BFD for multipoint networks specification.

6.1. Multipoint Client Session

If the head is keeping track of some or all of the tails, it has a session of type MultipointClient per tail that it cares about. All of the MultipointClient sessions for tails on a particular multipoint path are associated with the MultipointHead session to which the clients are listening. A BFD Poll Sequence may be sent over a MultipointClient session to a tail if the head wishes to verify connectivity. These sessions receive any BFD Control packets sent by the tails, and MUST NOT transmit periodic BFD Control packets other

than Poll Sequences (since periodic transmission is always done by the MultipointHead session).

6.2. Multipoint Client Session Failure

If a MultipointClient session receives a BFD Control packet from the tail with state Down or AdminDown, the head reliably knows that the tail has lost multipoint connectivity. If the Detection Time expires on a MultipointClient session, it is ambiguous as to whether the multipoint connectivity failed or whether there was a unicast path problem in one direction or the other, so the head does not reliably know the tail's state.

6.3. State Variables

BFD Multipoint active tail introduces new state variables and modifies the usage of a few existing ones defined in section 4.4 of [I-D.ietf-bfd-multipoint].

6.3.1. New State Variables

Few state variables are added in support of Multipoint BFD active tail.

bfd.SilentTail

If 0, a tail may send packets to the head according to other parts of this specification. Setting this to 1 allows tails to be provisioned to always be silent, even when the head is soliciting traffic from the tails. This can be useful, for example, in deployments of a large number of tails when the head wishes to track the state of a subset of them. This variable MUST be initialized based on configuration.

This variable is only pertinent when bfd.SessionType is MultipointTail and SHOULD NOT be modified after the MultipointTail session has been created.

bfd.ReportTailDown

Set to 1 if the head wishes tails to notify the head, via periodic BFD Control packets, when they see the BFD session fail. If 0, the tail will never send periodic BFD Control packets, and the head will not be notified of session failures by the tails. This variable MUST be initialized based on configuration.

This variable is only pertinent when bfd.SessionType is MultipointHead or MultipointClient.

bfd.UnicastRcvd

Set to 1 if a tail receives a unicast BFD Control packet from the head. This variable MUST be set to zero if the session transitions from Up state to some other state.

This variable MUST be initialized to zero.

This variable is only pertinent when bfd.SessionType is MultipointTail.

6.3.2. New State Variable Value

A new state variable value being added to:

bfd.SessionType

The type of this session as defined in [RFC7880]. A new value introduced is:

MultipointClient: A session on the head that tracks the state of an individual tail, when desirable.

This variable MUST be initialized to the appropriate type when the session is created, according to the rules in section 4.4 of [I-D.ietf-bfd-multipoint].

6.3.3. 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. The values of some of these variables relate to those of the same variables of a MultipointHead session (see section 4.4.2 of [I-D.ietf-bfd-multipoint]).

bfd.LocalDiscr

For session type MultipointClient, this variable MUST always match the value of bfd.LocalDiscr in the associated MultipointHead session.

bfd.DesiredMinTxInterval

For session type MultipointClient, this variable MUST always match the value of bfd.DesiredMinTxInterval in the associated MultipointHead session.

bfd.RequiredMinRxInterval

It MAY be set to zero at the head BFD system to suppress traffic from the tails. Setting it to zero in the MultipointHead session suppresses traffic from all tails, the setting in a MultipointClient session suppresses traffic from a single tail.

bfd.DemandMode

This variable MUST be initialized to 1 for session types MultipointClient.

bfd.DetectMult

For session type MultipointClient, this variable MUST always match the value of bfd.DetectMult in the associated MultipointHead session.

6.4. Controlling Multipoint BFD Options

The state variables defined above are used to choose which operational options are active.

The most basic form of operation, as explained in [I-D.ietf-bfd-multipoint], in which BFD Control packets flow only from the head and no tracking is desired of tail state at the head, is accomplished by setting bfd.ReportTailDown to 0 in the MultipointHead session (Section 5.1).

If the head wishes to know of active the tails, it sends multipoint Polls as needed. Previously known tails that don't respond to the Polls will be detected (as per Section 5.2.2).

If the head wishes to request a notification from the tails when they lose connectivity, it sets bfd.ReportTailDown to 1 in either the MultipointHead session (if such notification is desired from all tails) or in the MultipointClient session (if notification is desired from a particular tail). Note that the setting of this variable in a MultipointClient session for a particular tail overrides the setting in the MultipointHead session.

If the head wishes to verify the state of a tail on an ongoing basis, it sends a Poll Sequence from the MultipointClient session associated

with that tail as needed. This has the effect of eliminating the initial delay, described in Section 6.13.3, that the tail would otherwise insert prior to transmission of the packet thus the head may have notification of the session failure more quickly when comparing with use of m-poll.

If a tail wishes to operate silently (sending no BFD Control packets to the head) it sets bfd.SilentTail to 1 in the MultipointTail session. This allows a tail to be silent independent of the settings on the head.

6.5. State Machine

Though the state transitions for the state machine, as defined in section 5.5 of [I-D.ietf-bfd-multipoint], for a session type MultipointHead are only administratively driven, the state machine for a session of type MultipointClient is same and the diagram is applicable.

6.6. Session Establishment

If BFD Control packets are received at the head, they are demultiplexed to sessions of type MultipointClient, which represent the set of tails that the head is interested in tracking. These sessions will typically also be established dynamically based on the receipt of BFD Control packets. The head has broad latitude in choosing which tails to track, if any, without affecting the basic operation of the protocol. The head directly controls whether or not tails are allowed to send BFD Control packets back to the head by setting bfd.RequiredMinRxInterval to zero in a MultipointHead or a MultipointClient session.

6.7. Discriminators and Packet Demultiplexing

When the tails send BFD Control packets to the head from the MultipointTail session, the contents of Your Discr (the discriminator received from the head) will not be sufficient for the head to demultiplex the packet, since the same value will be received from all tails on the multicast tree. In this case, the head MUST demultiplex packets based on the source address and the value of Your Discr, which together uniquely identify the tail and the multipoint path.

When the head sends unicast BFD Control packets to a tail from a MultipointClient session, the value of Your Discr will be valid, and the tail MUST demultiplex the packet based solely on Your Discr.

6.8. Controlling Tail Packet Transmission

As the fan-in from the tails to the head may be very large, it is critical that the flow of BFD Control packets from the tails is controlled.

The head always operates in Demand mode. This means that no tail will send an asynchronous BFD Control packet as long as the session is Up.

The value of Required Min Rx Interval received by a tail in a unicast BFD Control packet, if any, always takes precedence over the value received in Multipoint BFD Control packets. This allows the packet rate from individual tails to be controlled separately as desired by sending a BFD Control packet from the corresponding MultipointClient session. This also eliminates the random delay, as discussed in Section 6.13.3, prior to transmission from the tail that would otherwise be inserted, reducing the latency of reporting a failure to the head.

If the head wishes to suppress traffic from the tails when they detect a session failure, it MAY set bfd.RequiredMinRxInterval to zero, which is a reserved value that indicates that the sender wishes to receive no periodic traffic. This can be set in the MultipointHead session (suppressing traffic from all tails) or it can be set in a MultipointClient session (suppressing traffic from only a single tail).

Any tail may be provisioned to never send *any* BFD Control packets to the head by setting bfd.SilentTail to 1. This provides a mechanism by which only a subset of tails reports their session status to the head.

6.9. Soliciting the Tails

If the head wishes to know of the active tails, the MultipointHead session MAY send a BFD Control packet as specified in Section 6.13.3, with the Poll (P) bit set to 1. This will cause all of the tails to reply with a unicast BFD Control Packet, randomized across one packet interval.

The decision as to when to send a multipoint Poll is outside the scope of this specification. However, it must never be sent more often than the regular multipoint BFD Control packet. Since the tail will treat a multipoint Poll like any other multipoint BFD Control packet, Polls may be sent in lieu of non-Poll packets.

Soliciting the tails also starts the Detection Timer for each of the associated MultipointClient sessions, which will cause those sessions to time out if the associated tails do not respond.

Note that for this mechanism to work properly, the Detection Time (which is equal to bfd.DesiredMinTxInterval) MUST be greater than the round trip time of BFD Control packets from the head to the tail (via the multipoint path) and back (via a unicast path). See Section 6.11 for more details.

6.10. Verifying Connectivity to Specific Tails

If the head wishes to verify connectivity to a specific tail, the corresponding MultipointClient session MAY send a BFD Poll Sequence to said tail. This might be done in reaction to the expiration of the Detection Timer (the tail didn't respond to a multipoint Poll), or it might be done on a proactive basis.

The interval between transmitted packets in the Poll Sequence MUST be calculated as specified in the base BFD specification [RFC5880] (the greater of bfd.DesiredMinTxInterval and bfd.RemoteMinRxInterval).

The value transmitted in Required Min RX Interval will be used by the tail (rather than the value received in any multipoint packet) when it transmits BFD Control packets to the head notifying it of a session failure and the transmitted packets will not be delayed. This value can potentially be set much lower than in the multipoint case, in order to speed up a notification to the head, since the value will be used only by the single tail. This value (and the lack of delay) are "sticky", in that once the tail receives it, it will continue to use it indefinitely. Therefore, if the head no longer wishes to single out the tail, it SHOULD reset the timer to the default by sending a Poll Sequence with the same value of Required Min Rx Interval as is carried in the multipoint packets, or it MAY reset the tail session by sending a Poll Sequence with state AdminDown (after the completion of which the session will come back up).

Note that a failure of the head to receive a response to a Poll Sequence does not necessarily mean that the tail has lost multipoint connectivity, though a reply to a Poll Sequence does reliably indicate connectivity or lack thereof (by virtue of the tail's state not being Up in the BFD Control packet).

6.11. Detection Times

MultipointClient sessions at the head are always in the Demand mode, and as such only care about detection time in two cases. First, if a Poll Sequence is being sent on a MultipointClient session, the detection time on this session is calculated according to the base BFD specification [RFC5880], that is, the transmission interval multiplied by bfd.DetectMult. Second, when a multipoint Poll is sent to solicit tail replies, the detection time on all associated MultipointClient sessions that aren't currently sending Poll Sequences is set to a value greater than or equal to bfd.RequiredMinRxInterval (one packet time). This value can be made arbitrarily large in order to ensure that the detection time is greater than the round trip time of a BFD Control packet between the head and the tail with no ill effects, other than delaying the detection of unresponsive tails. Note that a detection time expiration on a MultipointClient session at the head, while indicating a BFD session failure, cannot be construed to mean that the tail is not hearing multipoint packets from the head.

6.12. MultipointClient Down/AdminDown Sessions

If the MultipointHead session is going down (which only happens administratively), all associated MultipointClient sessions SHOULD be destroyed as they are superfluous.

If a MultipointClient session goes down due to the receipt of an unsolicited BFD Control packet from the tail with state Down or AdminDown (not in response to a Poll), and tail connectivity verification is not being done, the session MAY be destroyed. If verification is desired, the session SHOULD send a Poll Sequence and the session SHOULD be maintained.

If the tail replies to a Poll Sequence with state Down or AdminDown, it means that the tail session is definitely down. In this case, the session MAY be destroyed.

If the Detection Time expires on a MultipointClient session (meaning that the tail did not reply to a Poll Sequence) the session MAY be destroyed.

6.13. Base BFD for Multipoint Networks Specification Text Replacement

The following sections are meant to extend the corresponding sections in the base BFD for Multipoint Networks specification [I-D.ietf-bfd-multipoint].

6.13.1. Reception of BFD Control Packets

The following procedure updates parts of Section 5.13.1 of [I-D.ietf-bfd-multipoint].

When a BFD Control packet is received, the procedure defined in Section 5.13.1 of [I-D.ietf-bfd-multipoint] 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. In addition to that, if tail tracking is desired by the head, below procedure MUST be applied.

If bfd.SessionType is MultipointTail

If bfd.UnicastRcvd is 0 or the M bit is clear, set bfd.RemoteMinRxInterval to the value of Required Min RX Interval.

If the M bit is clear, set bfd.UnicastRcvd to 1.

Else (not MultipointTail)

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

If the Poll (P) bit is set, and bfd.SilentTail is zero, send a BFD Control packet to the remote system with the Poll (P) bit clear, and the Final (F) bit set (see Section 6.13.3).

6.13.2. Demultiplexing BFD Control Packets

This section is part of the addition to Section 5.13.2 of [I-D.ietf-bfd-multipoint], separated for clarity.

If 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.

If bfd.SessionType is MultipointHead

Find a MultipointClient session grouped to this MultipointHead session, based on the source address and the value of Your Discriminator. If a session is found and is not MultipointClient, the packet MUST be discarded. If no session is found, a new session of type

MultipointClient MAY be created, or the packet MAY be discarded. This choice is outside the scope of this specification.

If bfd.SessionType is not MultipointClient, the packet MUST be discarded.

6.13.3. Transmitting BFD Control Packets

A system MUST NOT periodically transmit BFD Control packets if bfd.SessionType is MultipointClient and a Poll Sequence is not being transmitted.

If bfd.SessionType value is MultipointTail and periodic the transmission of BFD Control packets is just starting (due to Demand mode not being active on the remote system), the first packet to be transmitted MUST be delayed by a random amount of time between zero and (0.9 * bfd.RemoteMinRxInterval).

If a BFD Control packet is received with the Poll (P) bit set to 1, the receiving system MUST transmit a BFD Control packet with the Poll (P) bit clear and the Final (F) bit, without respect to the transmission timer or any other transmission limitations, without respect to the session state, and without respect to whether Demand mode is active on either system. A system MAY limit the rate at which such packets are transmitted. If rate limiting is in effect, the advertised value of Desired Min TX Interval MUST be greater than or equal to the interval between transmitted packets imposed by the rate limiting function. If the Multipoint (M) bit is set in the received packet, the packet transmission MUST be delayed by a random amount of time between zero and (0.9 * bfd.RemoteMinRxInterval). Otherwise, the packet MUST be transmitted as soon as practicable.

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

Contents of transmitted packet MUST be as explained in section 5.13.3 of [I-D.ietf-bfd-multipoint].

7. Assumptions

If head notification is to be used, it is assumed that a multipoint BFD packet encapsulation contains enough information so that a tail can address a unicast BFD packet to the head.

If head notification is to be used, it is assumed that is that there is bidirectional unicast communication available (at the same

protocol layer within which BFD is being run) between the tail and head.

For the head to know reliably that a tail has lost multipoint connectivity, the unicast paths in both directions between that tail and the head must remain operational when the multipoint path fails. It is thus desirable that unicast paths not share fate with the multipoint path to the extent possible if the head wants more definite knowledge of the tail state.

Since the normal BFD three-way handshake is not used in this application, a tail transitioning from state Up to Down and back to Up again may not be reliably detected at the head.

8. IANA Considerations

This document has no actions for IANA.

9. Security Considerations

The same security considerations as those described in [RFC5880] and [I-D.ietf-bfd-multipoint] apply to this document.

Additionally, implementations that create MultpointClient sessions dynamically upon receipt of BFD Control packet from a tail MUST implement protective measures to prevent an infinite number of MultipointClient sessions being created. Below are listed some points to be considered in such implementations.

- When the number of MultipointClient sessions exceeds the number of expected tails, then the implementation should generate an alarm to users to indicate the anomaly.

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

This specification does not raise any additional security issues beyond those of the specifications referred to in the list of normative references.

10. Contributors

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

11. Acknowledgments

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

12. Normative References

[I-D.ietf-bfd-multipoint]
Katz, D., Ward, D., Networks, J., and G. Mirsky, "BFD for Multipoint Networks", draft-ietf-bfd-multipoint-16 (work in progress), April 2018.

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

[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>.

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

Authors' Addresses

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

Email: dkatz@juniper.net

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

Email: wardd@cisco.com

Santosh Pallagatti (editor)
Individual Contributor
Embassy Business Park
Bangalore, KA 560093
India

Email: santosh.pallagatti@gmail.com

Greg Mirsky (editor)
ZTE Corp.

Email: gregimirsky@gmail.com