Internet Engineering Task Force
Internet-Draft
Intended status: Standards Track
Expires: May 20, 2016

D. Katz
Juniper Networks
D. Ward
Cisco Systems
S. Pallagatti, Ed.
Juniper Networks

November 17, 2015

# BFD Multipoint Active Tails.
# draft-ietf-bfd-multipoint-active-tail-01

## Abstract

This document describes active tail extensions to the Bidirectional Forwarding Detection (BFD) protocol for multipoint and multicast networks. Comments on this draft should be directed to rtg-bfd@ietf.org.

## 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

## Copyright Notice

## Table of Contents

## 1. Introduction

This application of BFD is an extension to Multipoint BFD [I-D.ietf-bfd-multipoint] which allows tail to unreliably notify the head of the lack of multipoint connectivity. As a further option, this notification can be made reliable. Notification to the head can be enabled for all tails, or for only a subset of the tails.

Multipoint BFD base document [I-D.ietf-bfd-multipoint] describes procedures to verify only the head-to-tail connectivity over the multipoint path. Although it may use unicast paths in both directions, Multipoint BFD does not verify those paths (and in fact it is preferable if unicast paths share as little fate with the multipoint path as is feasible.)

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 implosion toward the head), it is a 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.

This document effectively modifies and adds to the base BFD specification and base BFD multipoint document.

## 2. Overview

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 an unreliable failure notification, the head can direct the tails to transmit unicast BFD Control packets back to the head when the path fails.

If the head wishes to know the identity of the tails on the multipoint path, it may solicit membership by sending 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. 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 somewhat reliable mechanism for detecting which tails fail to respond (implying a loss of multipoint connectivity.)

If the head wishes a reliable 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 the tail. This covers the case where

either the multipoint poll or the single reply thereto is lost in transit. If desired, the head may Poll one or more tails proactively to track the tails' connectivity.

If some tails are more equal than others, 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, even when the head wishes notification of path failure from the tail. Such tails will never be known to the head, but will still be able to detect multipoint path failures from the head.

## 3. Protocol Details

This section describes the operation of BFD Multipoint active tail in detail. This section is update to section 4 of [I-D.ietf-bfd-multipoint]

### 3.1. 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 grouped with the MultipointHead session to which the clients are listening. A BFD Poll Sequence may be sent over such a session to a tail if the head wishes to verify connectivity. These sessions receive any BFD Control packets sent by the tails, and never transmit periodic BFD Control packets other than Poll Sequences (since periodic transmission is always done by the MultipointHead session.)

### 3.2. 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 state.

## 3.3. 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].

### 3.3.1. New State Variables

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

bfd.SessionType

The type of this session as defined in [I-D.ietf-bfd-multipoint]. 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 4.13 of [I-D.ietf-bfd-multipoint].

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.

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.

### 3.3.2. State Variable Initialization and Maintenance

Some state variables defined in section 6.8.1 of the [RFC5880] needs to be initialized or manipulated differently depending on the session type (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 should be noted that for sessions of type MultipointTail, this variable only affects the rate of unicast Polls sent by the head; the rate of multipoint packets is necessarily unaffected by it.

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.

## 3.4. 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.

If the head wishes to know the identity of the tails, it sends multipoint Polls as needed.  Previously known tails that don't respond to the Polls will be detected.

If the head wishes to be notified by 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.

If the head wants to more quickly be alerted to a session failure from a particular tail, it sends a BFD Control packet from the MultipointClient session associated with that tail.  This has the effect of eliminating the initial delay that the tail would otherwise insert prior to transmission of the packet.

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.

## 3.5. State Machine

State machine for session of type MultipointClient is same as defined in section 4.5 of [I-D.ietf-bfd-multipoint].

## 3.6. 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.

## 3.7. 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.

## 3.8. 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 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 report their session status to the head.

## 3.9. Soliciting the Tails

If the head wishes to know the identities of the tails, the MultipointHead session MAY send a BFD Control packet as specified in Section 3.13.3, 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 associated MultipointClient session, 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 3.11 for more details.

## 3.10. 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 specification (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 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.)

## 3.11. Detection Times

MultipointClient sessions at the head are always in 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 specification, 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 BFD round trip time 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.

## 3.12. 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.

## 3.13. Base Specification Text Replacement

The following sections are meant to replace the corresponding sections in the base specification.

### 3.13.1. Reception of BFD Control Packets

The following procedure replaces section 6.8.6 of [RFC5880].

When a BFD Control packet is received, procedure defined in section 4.13.1 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 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 3.13.3.)
```

### 3.13.2. Demultiplexing BFD Control Packets

This section is part of the replacement for [RFC5880] section 6.8.6 and addition to section 4.13.2 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 MulticastHead

Find a MultipointClient session grouped to this MulticastHead session, based on the source address and the value of Your Discriminator.  If a session is found and is not MulticastClient, 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 MulticastClient, the packet MUST be discarded.

### 3.13.3.  Transmitting BFD Control Packets

The following procedure replaces section 6.8.7 of [RFC5880].

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

If bfd.SessionType is MulticastTail and periodic 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 4.13.3 of [I-D.ietf-bfd-multipoint].

## 4. 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 reliable knowledge of 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.

## 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.

### 5.1. No Head Notification

Since the only path used in this scenario is the multipoint path, 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.

## 5.2. Unreliable Head Notification

In this scenario, the tail sends back unsolicicted 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.

## 5.3. Semi-reliable Head Notification and Tail Solicitation

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.

## 5.4. Reliable Head Notification

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.

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 BFD 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. IANA Considerations

This document has no actions for IANA.

## 7. Security Considerations

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

## 8. Contributors

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

## 9. Acknowledgements

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.

## 10. Normative References

[I-D.ietf-bfd-multipoint]
Katz, D., Ward, D., and J. Networks, "BFD for Multipoint Networks", draft-ietf-bfd-multipoint-07 (work in progress), August 2015.

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

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

Authors' Addresses

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)
Juniper Networks
Embassy Business Park
Bangalore, KA  560093
India

Email: santoshpk@juniper.net