Internet Engineering Task Force
Internet-Draft
Updates: 5880 (if approved)
Intended status: Standards Track
Expires: November 7, 2016

C. Pignataro
D. Ward
Cisco
N. Akiya
Big Switch Networks
M. Bhatia
Ionos Networks
S. Pallagatti

May 6, 2016

Seamless Bidirectional Forwarding Detection (S-BFD) draft-ietf-bfd-seamless-base-11

Abstract

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

This document updates RFC5880.

Requirements Language

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

Status of This Memo

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

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

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

This Internet-Draft will expire on November 7, 2016.

Table of Contents

1. Introduction

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

Specifically, this document defines Seamless Bidirectional Forwarding Detection (S-BFD) a simplified mechanism to use Bidirectional Forwarding Detection (BFD) with large portions of negotiation aspects eliminated, thus providing benefits such as quick provisioning as well as improved control and flexibility to network nodes initiating the path monitoring. S-BFD enables cases benefiting from the use of core BFD technologies in a fashion that leverages existing implementations and protocol machinery while providing a rather simplified and largely stateless infrastructure for continuity testing.

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

2. Terminology

The reader is expected to be familiar with the BFD [RFC5880], IP [RFC0791] [RFC2460] and MPLS [RFC3031] terminologies and protocol constructs. This section describes several new terminologies introduced by S-BFD.

- Classical BFD - BFD session types based on [RFC5880].

- S-BFD - Seamless BFD.

- S-BFD control packet - a BFD control packet for the S-BFD mechanism.

- S-BFD echo packet - a BFD echo packet for the S-BFD mechanism.

- S-BFD packet - a BFD control packet or a BFD echo packet.

- Entity - a function on a network node that S-BFD mechanism allows remote network nodes to perform continuity test to. An entity can be abstract (e.g., reachability) or specific (e.g., IP addresses, router-IDs, functions).

- SBFDInitiator - an S-BFD session on a network node that performs a continuity test to a remote entity by sending S-BFD packets.

- SBFDReflector - an S-BFD session on a network node that listens for incoming S-BFD control packets to local entities and generates response S-BFD control packets.

- Reflector BFD session - synonymous with SBFDReflector.

- S-BFD discriminator - a BFD discriminator allocated for a local entity and is being listened by an SBFDReflector.

- BFD discriminator - a BFD discriminator allocated for an SBFDInitiator.

- Initiator - a network node hosting an SBFDInitiator.

- Responder - a network node hosting an SBFDReflector.

Figure 1 describes the relationship between S-BFD terminologies.

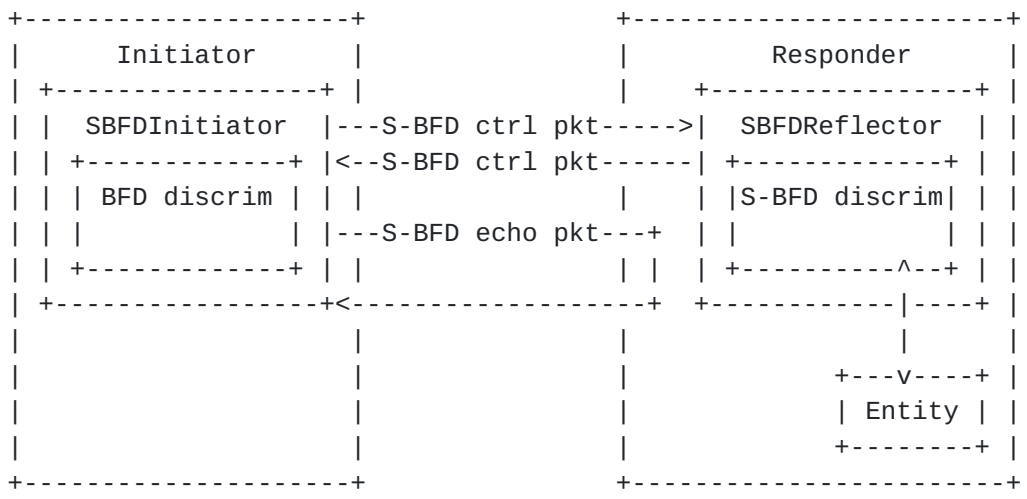


Figure 1: S-BFD Terminology Relationship

3. Seamless BFD Overview

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

Once the above setup is complete, any network node, having the knowledge of the S-BFD discriminator allocated by a remote node to remote entity/entities, can quickly perform a continuity test to the remote entity by simply sending S-BFD control packets with corresponding S-BFD discriminator value in the "your discriminator" field.

This is exemplified in Figure 2.

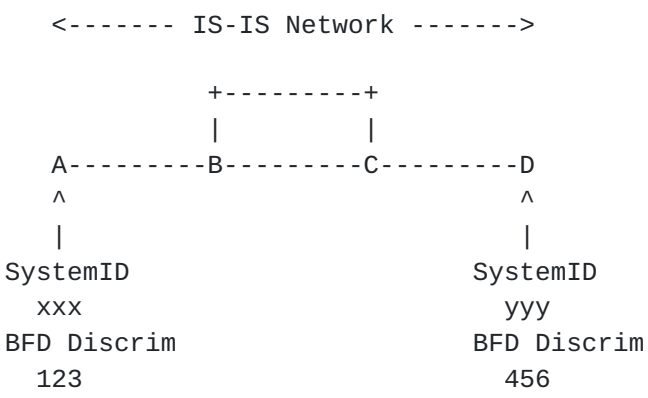


Figure 2: S-BFD for IS-IS Network

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

When a node allocates multiple S-BFD discriminators, how remote nodes determine which of the discriminators is associated with a specific entity is currently unspecified. The use of multiple S-BFD discriminators by a single network node is therefore discouraged until a means of learning the mapping is defined.

4. S-BFD Discriminators

4.1. S-BFD Discriminator Uniqueness

One important characteristic of an S-BFD discriminator is that it MUST be unique within an administrative domain. If multiple network nodes allocated the same S-BFD discriminator value, then S-BFD control packets falsely terminating on a wrong network node can result in a reflector BFD session to generate a response back, due to "your discriminator" matching. This is clearly not desirable.

4.2. Discriminator Pools

This subsection describes a discriminator pool implementation technique to minimize S-BFD discriminator collisions. The result will allow an implementation to better satisfy the S-BFD discriminator uniqueness requirement defined in Section 4.1.

- SBFDInitiator is to allocate a discriminator from the BFD discriminator pool. If the system also supports classical BFD that runs on [RFC5880], then the BFD discriminator pool SHOULD be shared by SBFDInitiator sessions and classical BFD sessions.

- SBFDReflector is to allocate a discriminator from the S-BFD discriminator pool. The S-BFD discriminator pool SHOULD be a separate pool than the BFD discriminator pool.

The remainder of this subsection describes the reasons for the suggestions above.

Locally allocated S-BFD discriminator values for entities, listened by SBFDReflector sessions, may be arbitrary allocated or derived from values provided by applications. These values may be protocol IDs (e.g., System-ID, Router-ID) or network targets (e.g., IP address). To avoid derived S-BFD discriminator values already being assigned to other BFD sessions (i.e., SBFDInitiator sessions and classical BFD sessions), it is RECOMMENDED that the discriminator pool for SBFDReflector sessions be separate from other BFD sessions.

Even when following the separate discriminator pool approach, collision is still possible between one S-BFD application to another S-BFD application, that may be using different values and algorithms to derive S-BFD discriminator values. If the two applications are using S-BFD for the same purpose (e.g., network reachability), then the colliding S-BFD discriminator value can be shared. If the two applications are using S-BFD for a different purpose, then the collision must be addressed. The use of multiple S-BFD discriminators by a single network node, however, is discouraged (see Section 3).

5. Reflector BFD Session

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

- MUST NOT transmit any S-BFD packets based on local timer expiry.

- MUST transmit an S-BFD control packet in response to a received S-BFD control packet having a valid S-BFD discriminator in the "your discriminator" field, unless prohibited by local policies (e.g., administrative, security, rate-limiter, etc.)

- MUST be capable of sending only two states: UP and ADMINDOWN.

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

Note that incoming S-BFD control packets may be IPv4, IPv6 or MPLS based [I-D.ietf-bfd-seamless-ip], and other options are possible and can be defined in future documents. How such S-BFD control packets reach an appropriate reflector BFD session is also a local matter, and is outside the scope of this document.

6. State Variables

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

6.1. New State Variables

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

- bfd.SessionType: This is a new state variable that describes the type of this session. Allowable values for S-BFD sessions are:

  * SBFDInitiator - an S-BFD session on a network node that performs a continuity test to a target entity by sending S-BFD packets.

  * SBFDReflector - an S-BFD session on a network node that listens for incoming S-BFD control packets to local entities and generates response S-BFD control packets.

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

6.2. State Variable Initialization and Maintenance

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

- bfd.DemandMode: This variable MUST be initialized to 1 for session type SBFDInitiator, and MUST be initialized to 0 for session type SBFDReflector. This is done to prevent loops (see Appendix A).

7. S-BFD Procedures

7.1. Demultiplexing of S-BFD Control Packet

S-BFD packet MUST be demultiplexed with lower layer information (e.g., dedicated destination UDP port [I-D.ietf-bfd-seamless-ip], associated channel type [I-D.ietf-pals-seamless-vccv]). The following procedure SHOULD be executed on both initiator and reflector.

```
If S-BFD packet

   If S-BFD packet is for SBFDReflector

      Packet MUST be looked up to locate a corresponding
      SBFDReflector session based on the value from the "your
      discriminator" field in the table describing S-BFD
      discriminators.

   Else

      Packet MUST be looked up to locate a corresponding
      SBFDInitiator session or classical BFD session based on the
      value from the "your discriminator" field in the table
      describing BFD discriminators.  If no match then received
      packet MUST be discarded.

      If session is SBFDInitiator

         Destination of the packet (i.e., destination IP address)
         SHOULD be validated to be for self.

      Else

         Packet MUST be discarded

Else

   Procedure described in [RFC5880] MUST be applied.
```

More details on S-BFD control packet demultiplexing are described in relevant S-BFD data plane documents.

7.2. Responder Procedures

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

7.2.1. Responder Demultiplexing

S-BFD packet MUST be demultiplexed with lower layer information. The following procedure SHOULD be executed by the responder:

If "your discriminator" not one of the entry allocated for local entities

Packet MUST be discarded.

Else

Packet is determined to be handled by a reflector BFD session responsible for that S-BFD discriminator.

If local policy allows (e.g., administrative, security, rate-limiter, etc.)

Chosen reflector BFD session SHOULD transmit a response BFD control packet using procedures described in Section 7.2.2.

7.2.2. Transmission of S-BFD Control Packet by SBFDReflector

Contents of S-BFD control packets sent by an SBFDReflector MUST be set as per Section 6.8.7 of [RFC5880]. There are a few fields that needs to be set differently from [RFC5880] as follows:

State (Sta)

Set to bfd.SessionState (either UP or ADMINDOWN only). Clarification of reflector BFD session state is described in Section 7.2.3.

Demand (D)

Set to 0, to identify the S-BFD packet is sent by the SBFDReflector.

Detect Mult

Value to be copied from "Detection Multiplier" filed of received BFD packet.

My Discriminator

Value be copied from "your discriminator" filed of received BFD packet.

Your Discriminator

Value be copied from "my discriminator" filed of received BFD packet.

Desired Min TX Interval

Value be copied from "Desired Min TX Interval" filed of received BFD packet.

Required Min RX Interval

Set to a bfd.RequiredMinRxInterval, value describing minimum interval, in microseconds between received SBFD Control packets. Further details are described in Section 7.2.3.

Required Min Echo RX Interval

If device supports looping back S-BFD echo packets

Set to the minimum required Echo packet receive interval for this session.

Else

Set to 0.

7.2.3. Additional SBFDReflector Behaviors

- S-BFD control packets transmitted by the SBFDReflector MUST have "Required Min RX Interval" set to a value that expresses, in microseconds, the minimum interval between incoming S-BFD control packets this SBFDReflector can handle. The SBFDReflector can control how fast SBFInitiators will be sending S-BFD control packets to self by ensuring "Required Min RX Interval" indicates a value based on the current load.

- When the SBFDReflector receives an S-BFD control packet from an SBFDInitiator, then the SBFDReflector needs to determine what "state" to send in the response S-BFD control packet. If the monitored local entity is in service, then the "state" MUST be set to UP. If the monitored local entity is "temporarily out of service", then the "state" SHOULD be set to ADMINDOWN.

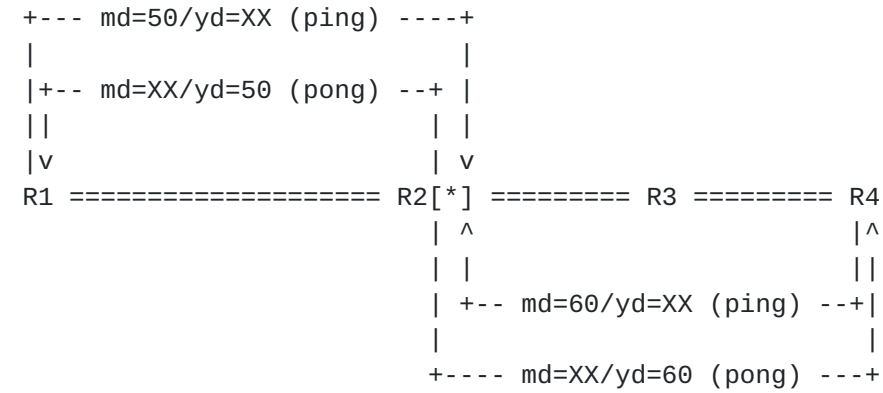
- If an SBFDReflector receives an S-BFD control packet with Demand (D) bit cleared, the packet MUST be discarded (see Appendix A).

7.3. Initiator Procedures

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

Every SBFDInitiator MUST have a locally unique "my discriminator" allocated from the BFD discriminator pool.

Figure 3 describes the high-level concept of continuity test using S-BFD. R2 allocates XX as the S-BFD discriminator for its network reachability purpose, and advertises XX to neighbors. ASCII art shows R1 and R4 performing a continuity test to R2.



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

Figure 3: S-BFD Continuity Test

7.3.1. SBFDInitiator State Machine

An SBFDInitiator may be a persistent session on the initiator with a timer for S-BFD control packet transmissions (stateful SBFDInitiator). An SBFDInitiator may also be a module, a script or a tool on the initiator that transmits one or more S-BFD control

packets "when needed" (stateless SBFDInitiator). For stateless SBFDInitiators, a complete BFD state machine may not be applicable. For stateful SBFDInitiators, the states and the state machine described in [RFC5880] will not function due to SBFDReflector session only sending UP and ADMINDOWN states (i.e., SBFDReflector session does not send INIT state). The following diagram provides the RECOMMENDED state machine for stateful SBFDInitiators. The notation on each arc represents the state of the SBFDInitiator (as received in the State field in the S-BFD control packet) or indicates the expiration of the Detection Timer. See Figure 4.

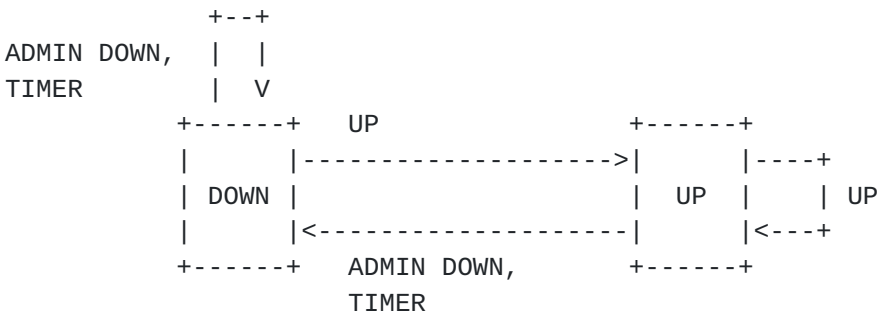


Figure 4: SBFDInitiator FSM

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

7.3.2. Transmission of S-BFD Control Packet by SBFDInitiator

Contents of S-BFD control packets sent by an SBFDInitiator MUST be set as per Section 6.8.7 of [RFC5880]. There are few fields which needs to be set differently from [RFC5880] as follows:

Demand (D)

D bit is used to identify S-BFD packet originated from SBFDInitiator and is always set to 1.

Your Discriminator

Set to bfd.RemoteDiscr. bfd.RemoteDiscr is set to discriminator value of remote entity. It MAY be learnt from routing protocols or configured locally.

Required Min RX Interval

Set to 0.

Required Min Echo RX Interval

Set to 0.

7.3.3. Additional SBFDInitiator Behaviors

- If the SBFDInitiator receives a valid S-BFD control packet in response to transmitted S-BFD control packet to a remote entity, then the SBFDInitiator SHOULD conclude that S-BFD control packet reached the intended remote entity.

- When an SBFDInitiator receives a response S-BFD control packet, if the state specified is ADMINDOWN, the SBFDInitiator MUST NOT conclude loss of reachability to the corresponding remote entity, and MUST back off packet transmission interval for the remote entity to an interval no faster than 1 second.

- When a sufficient number of S-BFD packets have not arrived as they should, the SBFDInitiator SHOULD declare loss of reachability to the remote entity. The criteria for declaring loss of reachability and the action that would be triggered as a result are outside the scope of this document; the action MAY include logging an error.

- Relating to above bullet item, it is critical for an implementation to understand the latency to/from the reflector BFD session on the responder. In other words, for very first S-BFD packet transmitted by the SBFDInitiator, an implementation MUST NOT expect response S-BFD packet to be received for time equivalent to sum of latencies: initiator to responder and responder back to initiator.

- If the SBFDInitiator receives an S-BFD control packet with Demand (D) bit set, the packet MUST be discarded (see Appendix A).

7.4. Diagnostic Values

Diagnostic value in both directions MAY be set to a certain value, to attempt to communicate further information to both ends. Implementation MAY use already existing diagnostic values defined in Section 4.1 of [RFC5880]. However, details of such are outside the scope of this specification.

7.5. The Poll Sequence

Poll sequence MAY be used in both directions. The Poll sequence MUST operate in accordance with [RFC5880]. An SBFDReflector MAY use the Poll sequence to slow down that rate at which S-BFD control packets are generated from an SBFDInitiator. This is done by the SBFDReflector using procedures described in Section 7.2.3 and setting the Poll (P) bit in the reflected S-BFD control packet. The SBFDInitiator is to then send the next S-BFD control packet with the Final (F) bit set. If an SBFDReflector receives an S-BFD control packet with Poll (P) bit set, then the SBFDReflector MUST respond with an S-BFD control packet with Poll (P) bit cleared and Final (F) bit set.

8. Operational Considerations

S-BFD provides a smooth and continuous (i.e., seamless) operational experience as an Operations, Administration, and Maintenance (OAM) mechanism for connectivity check and connection verification. This is achieved by providing a simplified mechanism with large portions of negotiation aspects eliminated, resulting in a faster and simpler provisioning.

Because of this simplified mechanism, due to a misconfiguration, an SBFDInitiator could send S-BFD control packets to a target that does not exist or that is outside the S-BFD administrative domain. As explained in Section 7.3.1, an SBFDInitiator can be a "persistent" initiator or a "when needed" one. When an S-BFD "persistent" SBFDInitiator is used, it SHOULD be controlled that S-BFD control packet do not propagate for an extended period of time outside of the administrative domain that uses it. Further, operational measures SHOULD be taken to identify if S-BFD packets are not responded to for an extended period of time, and remediate the situation. These potential concerns are largely mitigated by dynamic advertisement mechanisms for S-BFD, and with automation checks before applying configurations.

8.1. Scaling Aspect

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

8.2. Congestion Considerations

S-BFD performs failure detection by consuming resources, including bandwidth and CPU processing. It is therefore imperative that operators correctly provision the rates at which S-BFD is transmitted to avoid congestion. 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 exact mechanism used to detect congestion is outside the scope of this specification, but may include detection of lost BFD control packets or other means. The SBFDReflector can limit the rate at which an SBFInitiators will be sending S-BFD control packets utilizing the "Required Min RX Interval", at the expense of increasing the detection time.

9. Co-existence with Classical BFD Sessions

Initial packet demultiplexing requirement is described in Section 7.1. Because of this, S-BFD mechanism can co-exist with classical BFD sessions.

10. S-BFD Echo Function

The concept of the S-BFD Echo function is similar to the BFD Echo function described in [RFC5880]. S-BFD echo packets have the destination of self, thus S-BFD echo packets are self-generated and self-terminated after traversing a link/path. S-BFD echo packets are expected to u-turn on the target node in the data plane and MUST NOT be processed by any reflector BFD sessions on the target node.

When using the S-BFD Echo function, it is RECOMMENDED that:

- Both S-BFD control packets and S-BFD echo packets be sent.
- Both S-BFD control packets and S-BFD echo packets have the same semantics in the forward direction to reach the target node.

In other words, it is not preferable to send just S-BFD echo packets without also sending S-BFD control packets. There are two reasons behind this suggestion:

- S-BFD control packets can verify the reachability to intended target node, which allows one to have confidence that S-BFD echo packets are u-turning on the expected target node.
- S-BFD control packets can detect when the target node is going out of service (i.e., via receiving back ADMINDOWN state).

S-BFD Echo packets can be spoofed, and can u-turn in a transit node before reaching the expected target node. When the S-BFD Echo function is used, it is RECOMMENDED in this specification that both S-BFD control packets and S-BFD echo packets be sent. While the additional use of S-BFD control packets alleviates these two concerns, some form of authentication MAY still be included.

The usage of the "Required Min Echo RX Interval" field is described in Section 7.3.2 and Section 7.2.2. Because of the stateless nature of SBFDReflector sessions, a value specified the "Required Min Echo RX Interval" field is not very meaningful at SBFDReflector. Thus it is RECOMMENDED that the "Required Min Echo RX Interval" field simply be set to zero from SBFDInitiator. SBFDReflector MAY set to appropriate value to control the rate at which it wants to receives SBFD echo packets.

The following aspects of S-BFD Echo functions are left as implementation details, and are outside the scope of this document:

- Format of the S-BFD echo packet (e.g., data beyond UDP header).

- Procedures on when and how to use the S-BFD Echo function.

11. Security Considerations

Same security considerations as [RFC5880] apply to this document. Additionally, implementing the following measures will strengthen security aspects of the mechanism described by this document:

- SBFDInitiator MAY pick a sequence number to be set in "sequence Number" in authentication section based on authentication mode configured.

- SBFDReflector MUST NOT use the crypto sequence number to make a decision about accepting the packet. This is because the SBFDReflector does not maintain S-BFD peer state, and because the SBFDReflector can receive S-BFD packets from multiple SBFDInitiators. Consequently, BFD authentication can be used but not the sequence number.

- SBFDReflector MAY use the Auth Key ID in the incoming packet to verify the authentication data.

- SBFDReflector MUST accept the packet if authentication is successful.

- SBFDReflector MUST compute the Authentication data and MUST use the same sequence number that it received in the S-BFD control packet that it is responding to.

- SBFDInitiator SHOULD accept S-BFD control packet with sequence number within permissible window. One potential approach is the procedure explained in [I-D.ietf-bfd-generic-crypto-auth].

Using the above method,

- SBFDReflector continue to remain stateless despite using security.

- SBFDReflector are not susceptible to replay attacks as they always respond to S-BFD control packets irrespective of the sequence number carried.

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

Additionally, the use of strong forms of authentication is strongly encouraged for S-BFD. The use of Simple Password authentication potentially puts other services at risk, if S-BFD packets can be intercepted and if those password values are reused for other services.

Considerations about loop problems are covered in Appendix A.

12. IANA Considerations

No action is required by IANA for this document.

13. Acknowledgements

The authors would like to thank Jeffrey Haas, Greg Mirsky, Marc Binderberger, and Alvaro Retana for performing thorough reviews and providing number of suggestions. The authors would also like to thank Girija Raghavendra Rao, Les Ginsberg, Srihari Raghavan, Vanitha Neelamegam, and Vengada Prasad Govindan from Cisco Systems for providing valuable comments. Finally, the authors would also like to thank John E. Drake and Pablo Frank for providing comments and suggestions.

14. Contributors

The following are key contributors to this document:

Tarek Saad, Cisco Systems, Inc.
Siva Sivabalan, Cisco Systems, Inc.
Nagendra Kumar, Cisco Systems, Inc.
Mallik Mudigonda, Cisco Systems, Inc.
Sam Aldrin, Google

15. References

15.1. 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, <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>.

15.2. Informative References

[I-D.ietf-bfd-generic-crypto-auth]
Bhatia, M., Manral, V., Zhang, D., and M. Jethanandani, "BFD Generic Cryptographic Authentication", draft-ietf-bfd-generic-crypto-auth-06 (work in progress), April 2014.

[I-D.ietf-bfd-seamless-ip]
Akiya, N., Pignataro, C., and D. Ward, "Seamless Bidirectional Forwarding Detection (S-BFD) for IPv4, IPv6 and MPLS", draft-ietf-bfd-seamless-ip-04 (work in progress), April 2016.

[I-D.ietf-bfd-seamless-use-case]
Aldrin, S., Pignataro, C., Mirsky, G., and N. Kumar, "Seamless Bidirectional Forwarding Detection (S-BFD) Use Cases", draft-ietf-bfd-seamless-use-case-06 (work in progress), April 2016.

[I-D.ietf-pals-seamless-vccv]
Govindan, V. and C. Pignataro, "Seamless BFD for VCCV", draft-ietf-pals-seamless-vccv-03 (work in progress), April 2016.

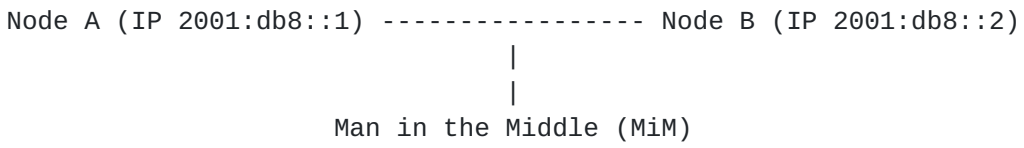
[RFC0791] Postel, J., "Internet Protocol", STD 5, RFC 791, DOI 10.17487/RFC0791, September 1981, <http://www.rfc-editor.org/info/rfc791>.

[RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, DOI 10.17487/RFC2460, December 1998, <http://www.rfc-editor.org/info/rfc2460>.

[RFC3031] Rosen, E., Viswanathan, A., and R. Callon, "Multiprotocol Label Switching Architecture", RFC 3031, DOI 10.17487/RFC3031, January 2001, <http://www.rfc-editor.org/info/rfc3031>.

Appendix A. Loop Problem and Solution

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



Assume node A reserved a discriminator 0x01010101 for target identifier 2001:db8::1 and has a reflector session in listening mode. Similarly node B reserved a discriminator 0x02020202 for its target identifier 2001:db8::2 and also has a reflector session in listening mode.

Suppose MiM sends a spoofed packet with MyDisc = 0x01010101, YourDisc = 0x02020202, source IP as 2001:db8::1 and dest IP as 2001:db8::2. When this packet reaches Node B, the reflector session on Node B will swap the discriminators and IP addresses of the received packet and reflect it back, since YourDisc of the received packet matched with reserved discriminator of Node B. The reflected packet that reached Node A will have MyDdisc=0x02020202 and YourDisc=0x01010101. Since YourDisc of the received packet matched the reserved discriminator of Node A, Node A will swap the discriminators and reflects the packet back to Node B. Since reflectors must set the TTL of the reflected packets to 255, the above scenario will result in an infinite loop with just one malicious packet injected from MiM.

The solution to avoid the loop problem uses the "D" bit (Demand mode bit). The Initiator always sets the 'D' bit and the reflector always clears it. This way we can identify if a received packet was a reflected packet and avoid reflecting it back.

Authors' Addresses

Carlos Pignataro
Cisco Systems, Inc.

Email: cpignata@cisco.com

Dave Ward
Cisco Systems, Inc.

Email: wardd@cisco.com

Nobo Akiya
Big Switch Networks

Email: nobo.akiya.dev@gmail.com

Manav Bhatia
Ionos Networks

Email: manav@ionosnetworks.com

Santosh Pallagatti

Email: santosh.pallagatti@gmail.com