Internet Engineering Task Force (IETF)
Request for Comments: 7880
Updates: 5880
Category: Standards Track
ISSN: 2070-1721

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

July 2016

Seamless Bidirectional Forwarding Detection (S-BFD)

Abstract

This document defines Seamless Bidirectional Forwarding Detection (S-BFD), a simplified mechanism for using BFD with a large proportion of negotiation aspects eliminated, thus providing benefits such as quick provisioning, as well as improved control and flexibility for network nodes initiating path monitoring.

This document updates RFC 5880.

Status of This Memo

This is an Internet Standards Track document.

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

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at http://www.rfc-editor.org/info/rfc7880.

Copyright Notice

Table of Contents

1. Introduction

Bidirectional Forwarding Detection (BFD), as described in [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, expand failure detection coverage, and allow BFD usage for wider scenarios. Additional use cases are listed in [RFC7882].

Specifically, this document defines Seamless Bidirectional Forwarding Detection (S-BFD), a simplified mechanism for using BFD with a large proportion of negotiation aspects eliminated, thus providing benefits such as quick provisioning, as well as improved control and flexibility for network nodes initiating 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 a smooth and continuous operational experience for applications. Therefore, "Seamless BFD" (S-BFD) has been chosen as the name for this mechanism.

2. Terminology

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

The reader is expected to be familiar with the BFD [RFC5880], IP [RFC791] [RFC2460], and MPLS [RFC3031] terms and protocol constructs. The remainder of this section describes several new terms 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 to which the S-BFD mechanism allows remote network nodes to perform continuity tests. 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. An SBFDReflector listens for S-BFD Discriminators.

- 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 terms.

```
    +---------------------+                +------------------------+
    |      Initiator      |                |         Responder      |
    | +-----------------+ |                |    +-----------------+ |
    | |  SBFDInitiator  |---S-BFD Ctrl pkt----->|  SBFDReflector  | |
    | | +-------------+ |<--S-BFD Ctrl pkt------| +-------------+ | |
    | | | BFD Discrim | | |                |    | |S-BFD Discrim| | |
    | | |             | |---S-BFD Echo pkt---+  | |             | | |
    | | +-------------+ | |                | |  | +----------^--+ | |
    | +-----------------+<-------------------+  +------------|----+ |
    |                     |                |                 |      |
    |                     |                |             +---v----+ |
    |                     |                |             | Entity | |
    |                     |                |             +--------+ |
    +---------------------+                +------------------------+
```

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). The required result is that applications on other network nodes will know about the S-BFD Discriminators allocated by a remote node to remote entities. The Reflector BFD session, upon receiving an S-BFD Control packet targeted to one of the local S-BFD Discriminator values, is to transmit a response S-BFD Control packet back to the initiator.

Once the above setup is complete, any network node that knows about the S-BFD Discriminator allocated by a remote node to a remote entity or entities can quickly perform a continuity test to the remote entity by simply sending S-BFD Control packets with a corresponding S-BFD Discriminator value in the Your Discriminator field.

This is exemplified in Figure 2.

```
                   <------- IS-IS Network ------->

                             +---------+
                             |         |
                   A---------B---------C---------D
                   ^                             ^
                   |                             |
               System-ID                     System-ID
                  xxx                           yyy
              BFD Discrim                   BFD Discrim
                  123                           456
```

Figure 2: S-BFD for IS-IS Network

An S-BFD module in a system with IS-IS System-ID xxx (Node A) allocates an S-BFD Discriminator 123, and IS-IS advertises the S-BFD Discriminator 123 in an IS-IS TLV. An S-BFD module in a system with IS-IS System-ID 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 Node A wants to check the reachability of Node D, Node A can send an S-BFD Control packet destined to Node D with the 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 allocate 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 generating a response back because of a matching Your Discriminator value. This is clearly not desirable.

4.2. Discriminator Pools

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

- An SBFDInitiator is to allocate a discriminator from the BFD Discriminator pool. If the system also supports classical BFD (i.e., implements [RFC5880]), then the BFD Discriminator pool SHOULD be shared by SBFDInitiator sessions and classical BFD sessions.

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

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

Locally allocated S-BFD Discriminator values for entities that SBFDReflector sessions are listening for may be arbitrarily 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, a collision is still possible between different S-BFD applications that may be using different values and algorithms to derive S-BFD Discriminator values. If 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 the Your Discriminator field 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).

- 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 a few Reflector BFD sessions may each be responsible for a 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 based on IPv4, IPv6, or MPLS [RFC7881]. Note also that other options are possible and may 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 a particular 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.

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

6.2. State Variable Initialization and Maintenance

State variables (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

An S-BFD packet MUST be demultiplexed with lower-layer information (e.g., dedicated destination UDP port [RFC7881], associated Channel Type [RFC7885]). The following procedure SHOULD be executed on both initiator and reflector:

```
If the packet is an S-BFD packet

   If the S-BFD packet is for an SBFDReflector

      The 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

      The 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 the
      received packet MUST be discarded.

      If the session is an SBFDInitiator session

         The destination of the packet (i.e., the destination IP
         address) SHOULD be verified as being for itself.

      Else

         The packet MUST be discarded.

Else

   The procedure described in Section 6.8.6 of [RFC5880] MUST be
   applied.
```

More details on S-BFD Control packet demultiplexing are provided 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 to as the responder. The responder, upon reception of S-BFD Control packets, is to verify the validity of the packets, as described in [RFC5880].

7.2.1. Responder Demultiplexing

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

```
If the Your Discriminator field is not one of the entries
allocated for local entities

   The packet MUST be discarded.

Else

   The packet is determined to be handled by a Reflector BFD
   session responsible for that S-BFD Discriminator.

   If allowable per local policy (e.g., administrative, security,
   rate-limiter)

      The chosen Reflector BFD session SHOULD transmit a response
      BFD Control packet using the procedures described in
      Section 7.2.2.
```

7.2.2. Transmission of S-BFD Control Packet by SBFDReflector

The 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 need 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 indicate that the S-BFD packet is sent by the SBFDReflector.

Detect Mult

Value to be copied from the Detection Multiplier field of the received BFD packet.

My Discriminator

Value to be copied from the Your Discriminator field of the received BFD packet.

Your Discriminator

Value to be copied from the My Discriminator field of the received BFD packet.

Desired Min TX Interval

Value to be copied from the Desired Min TX Interval field of the received BFD packet.

Required Min RX Interval

Set to bfd.RequiredMinRxInterval. Value indicating the minimum interval, in microseconds, between received S-BFD Control packets. Further details are provided in Section 7.2.3.

Required Min Echo RX Interval

If the device supports looping back S-BFD Echo packets

Set to the minimum required S-BFD 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 that this SBFDReflector can handle. The SBFDReflector can control how fast SBFDInitiators will be sending S-BFD Control packets to themselves by ensuring that 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 the 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 the Your Discriminator field to an S-BFD Discriminator corresponding to the remote entity.

Every SBFDInitiator MUST have a locally unique My Discriminator value allocated from the BFD Discriminator pool.

Figure 3 describes the high-level concept of continuity testing using S-BFD. R2 allocates XX as the S-BFD Discriminator for network reachability purposes and advertises XX to neighbors. Figure 3 shows R1 and R4 performing a continuity test to R2.

```
          +--- md=50/yd=XX (ping) ----+
          |                           |
          |+-- md=XX/yd=50 (pong) --+ |
          ||                        | |
          |v                        | v
          R1 ==================== R2[*] ========= R3 ========= R4
                                    | ^                        |^
                                    | |                        ||
                                    | +-- md=60/yd=XX (ping) --+|
                                    |                           |
                                    +---- md=XX/yd=60 (pong) ---+

         [*] 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 the SBFDReflector session only sending the UP and AdminDown states (i.e., the SBFDReflector session does not send the 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.

```
                    +--+
       ADMIN DOWN,  |  |
       TIMER        |  V
                  +------+   UP                +------+
                  |      |-------------------->|      |----+
                  | DOWN |                     |  UP  |    | UP
                  |      |<--------------------|      |<---+
                  +------+   ADMIN DOWN,       +------+
                             TIMER
```

Figure 4: SBFDInitiator Finite State Machine

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 an UP state setting is received by the SBFDInitiator. The definitions of the states and events have the same meanings as those defined in the base BFD specification [RFC5880].

7.3.2. Transmission of S-BFD Control Packet by SBFDInitiator

The contents of S-BFD Control packets sent by an SBFDInitiator MUST be set as per Section 6.8.7 of [RFC5880]. There are a few fields that need to be set differently from [RFC5880], as follows:

Demand (D)

Used to indicate that the S-BFD packet originated from the SBFDInitiator. Always set to 1.

Your Discriminator

Set to bfd.RemoteDiscr. bfd.RemoteDiscr is set to the Discriminator value of the 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 a transmitted S-BFD Control packet to a remote entity, then the SBFDInitiator SHOULD conclude that the 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 that the reachability of the corresponding remote entity is lost and MUST back off the 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.

- Regarding the third 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 the very first S-BFD packet transmitted by the SBFDInitiator, an implementation MUST NOT expect a response S-BFD packet to be received for a time equivalent to the sum of the latencies: initiator to responder and responder back to initiator.

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

7.4. Diagnostic Values

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

7.5. The Poll Sequence

The 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 the rate at which S-BFD Control packets are generated from an SBFDInitiator. This is done by the SBFDReflector, using the 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 the P bit set, then the SBFDReflector MUST respond with an S-BFD Control packet with the P bit cleared and the 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 checking and connection verification. This is achieved by providing a simplified mechanism with a large proportion of negotiation aspects eliminated, resulting in 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, a deployment SHOULD ensure that S-BFD Control packets do not propagate for an extended period of time outside of the administrative domain that uses it. Further, operational measures SHOULD be taken to determine if responses to S-BFD packets are not sent for an extended period of time and then 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 the scaling aspect: the number of SBFDReflectors. This specification eliminates the need for egress nodes to have fully active BFD sessions when only one side desires to perform continuity tests. With the introduction of the Reflector BFD concept, egress is no longer required to create any active BFD sessions on a per-path/LSP/function basis. Because of this, the total number of BFD sessions in a network is reduced.

8.2. Congestion Considerations

When S-BFD performs failure detection, it consumes resources, including bandwidth and CPU processing. To avoid congestion, it is therefore imperative that operators correctly provision the rates at which S-BFD packets are transmitted. 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 the detection of lost BFD Control packets or other means. The SBFDReflector can limit the rate at which SBFDInitiators will be sending S-BFD Control packets by utilizing Required Min RX Interval, but at the expense of detection time (i.e., detection time will increase).

9. Co-existence with Classical BFD Sessions

Demultiplexing requirements for the initial packet are described in Section 7.1. Because of this, the 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 of the intended target node; this 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., by receiving 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 Sections 7.2.2 and 7.3.2. Because of the stateless nature of SBFDReflector sessions, a value specified in the Required Min Echo RX Interval field is not very meaningful to the SBFDReflector. Thus, it is RECOMMENDED that the Required Min Echo RX Interval field simply be set to zero by the SBFDInitiator. The SBFDReflector MAY set the Required Min Echo RX Interval field to an appropriate value to control the rate at which it wants to receive S-BFD 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

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

- The SBFDInitiator MAY pick a sequence number to be set in "sequence number" in the Authentication Section, based on the configured authentication mode.
- The 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.
- The SBFDReflector MAY use the Auth Key ID in the incoming packet to verify the Authentication Data.
- The SBFDReflector MUST accept the packet if authentication is successful.
- The SBFDReflector MUST compute the Authentication Data and MUST use the same sequence number that it received in the S-BFD Control packet to which it is responding.
- The SBFDInitiator SHOULD accept an S-BFD Control packet with a sequence number within the permissible range. One potential approach is the procedure explained in [BFD-GEN-AUTH].

Using the above method,

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

- o 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 [RFC5880] potentially puts other services at risk if S-BFD packets can be intercepted and those password values are reused for other services.

Considerations related to loop problems are covered in Appendix A.

12. References

12.1. 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>.

12.2. Informative References

[BFD-GEN-AUTH] Bhatia, M., Manral, V., Zhang, D., and M. Jethanandani, "BFD Generic Cryptographic Authentication", Work in Progress, draft-ietf-bfd-generic-crypto-auth-06, April 2014.

[RFC791] Postel, J., "Internet Protocol", STD 5, RFC 791, DOI 10.17487/RFC791, 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>.

[RFC7881] Pignataro, C., Ward, D., and N. Akiya, "Seamless Bidirectional Forwarding Detection (S-BFD) for IPv4, IPv6, and MPLS", RFC 7881, DOI 10.17487/RFC7881, July 2016, <http://www.rfc-editor.org/info/rfc7881>.

[RFC7882] Aldrin, S., Pignataro, C., Mirsky, G., and N. Kumar, "Seamless Bidirectional Forwarding Detection (S-BFD) Use Cases", RFC 7882, DOI 10.17487/RFC7882, July 2016, <http://www.rfc-editor.org/info/rfc7882>.

[RFC7885] Govindan, V. and C. Pignataro, "Seamless Bidirectional Forwarding Detection (S-BFD) for Virtual Circuit Connectivity Verification (VCCV)", RFC 7885, DOI 10.17487/RFC7885, July 2016, <http://www.rfc-editor.org/info/rfc7885>.

Appendix A. Loop Problem and Solution

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

```
 Node A (IP 2001:db8::1) ----------------- Node B (IP 2001:db8::2)
                                  |
                                  |
                      Man in the Middle (MITM)
```

Assume that 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 that a MITM sends a spoofed packet with My Discriminator = 0x01010101, Your Discriminator = 0x02020202, source IP as 2001:db8::1, and destination 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 the Your Discriminator value of the received packet matches the reserved discriminator of Node B. The reflected packet that reached Node A will have My Discriminator = 0x02020202 and Your Discriminator = 0x01010101. Since the Your Discriminator value of the received packet matches the reserved discriminator of Node A, Node A will swap the discriminators and reflect the packet back to Node B. Since the reflectors must set the TTL of the reflected packets to 255, the above scenario will result in an infinite loop because of just one malicious packet injected from the MITM.

The solution is to avoid the loop problem by using the D bit (Demand mode bit). The initiator always sets the D bit, and the reflector always clears it. This way, we can determine if a received packet was a reflected packet and avoid reflecting it back.

Acknowledgements

The authors would like to thank Jeffrey Haas, Greg Mirsky, Marc Binderberger, and Alvaro Retana for performing thorough reviews and providing a 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.

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

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