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

D. Katz
D. Ward
Juniper Networks

June 2010

Bidirectional Forwarding Detection (BFD)

Abstract

This document describes a protocol intended to detect faults in the bidirectional path between two forwarding engines, including interfaces, data link(s), and to the extent possible the forwarding engines themselves, with potentially very low latency. It operates independently of media, data protocols, and routing protocols.

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 5741.

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/rfc5880.

Table of Contents

1. Introduction

An increasingly important feature of networking equipment is the rapid detection of communication failures between adjacent systems, in order to more quickly establish alternative paths. Detection can come fairly quickly in certain circumstances when data link hardware comes into play (such as Synchronous Optical Network (SONET) alarms). However, there are media that do not provide this kind of signaling (such as Ethernet), and some media may not detect certain kinds of failures in the path, for example, failing interfaces or forwarding engine components.

Networks use relatively slow "Hello" mechanisms, usually in routing protocols, to detect failures when there is no hardware signaling to help out. The time to detect failures ("Detection Times") available in the existing protocols are no better than a second, which is far too long for some applications and represents a great deal of lost data at gigabit rates. Furthermore, routing protocol Hellos are of no help when those routing protocols are not in use, and the semantics of detection are subtly different -- they detect a failure in the path between the two routing protocol engines.

The goal of Bidirectional Forwarding Detection (BFD) is to provide low-overhead, short-duration detection of failures in the path between adjacent forwarding engines, including the interfaces, data link(s), and, to the extent possible, the forwarding engines themselves.

An additional goal is to provide a single mechanism that can be used for liveness detection over any media, at any protocol layer, with a wide range of Detection Times and overhead, to avoid a proliferation of different methods.

This document specifies the details of the base protocol. The use of some mechanisms are application dependent and are specified in a separate series of application documents. These issues are so noted.

Note that many of the exact mechanisms are implementation dependent and will not affect interoperability, and are thus outside the scope of this specification. Those issues are so noted.

1.1. Conventions Used in This Document

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 [KEYWORDS].

2. Design

BFD is designed to detect failures in communication with a forwarding plane next hop. It is intended to be implemented in some component of the forwarding engine of a system, in cases where the forwarding and control engines are separated. This not only binds the protocol more to the forwarding plane, but decouples the protocol from the fate of the routing protocol engine, making it useful in concert with various "graceful restart" mechanisms for those protocols. BFD may also be implemented in the control engine, though doing so may preclude the detection of some kinds of failures.

BFD operates on top of any data protocol (network layer, link layer, tunnels, etc.) being forwarded between two systems. It is always run in a unicast, point-to-point mode. BFD packets are carried as the payload of whatever encapsulating protocol is appropriate for the medium and network. BFD may be running at multiple layers in a system. The context of the operation of any particular BFD session is bound to its encapsulation.

BFD can provide failure detection on any kind of path between systems, including direct physical links, virtual circuits, tunnels, MPLS Label Switched Paths (LSPs), multihop routed paths, and unidirectional links (so long as there is some return path, of course). Multiple BFD sessions can be established between the same pair of systems when multiple paths between them are present in at least one direction, even if a lesser number of paths are available in the other direction (multiple parallel unidirectional links or MPLS LSPs, for example).

The BFD state machine implements a three-way handshake, both when establishing a BFD session and when tearing it down for any reason, to ensure that both systems are aware of the state change.

BFD can be abstracted as a simple service. The service primitives provided by BFD are to create, destroy, and modify a session, given the destination address and other parameters. BFD in return provides a signal to its clients indicating when the BFD session goes up or down.

3. Protocol Overview

BFD is a simple Hello protocol that, in many respects, is similar to the detection components of well-known routing protocols. A pair of systems transmit BFD packets periodically over each path between the two systems, and if a system stops receiving BFD packets for long enough, some component in that particular bidirectional path to the neighboring system is assumed to have failed. Under some conditions, systems may negotiate not to send periodic BFD packets in order to reduce overhead.

A path is only declared to be operational when two-way communication has been established between systems, though this does not preclude the use of unidirectional links.

A separate BFD session is created for each communications path and data protocol in use between two systems.

Each system estimates how quickly it can send and receive BFD packets in order to come to an agreement with its neighbor about how rapidly detection of failure will take place. These estimates can be modified in real time in order to adapt to unusual situations. This design also allows for fast systems on a shared medium with a slow system to be able to more rapidly detect failures between the fast systems while allowing the slow system to participate to the best of its ability.

3.1. Addressing and Session Establishment

A BFD session is established based on the needs of the application that will be making use of it. It is up to the application to determine the need for BFD, and the addresses to use -- there is no discovery mechanism in BFD. For example, an OSPF [OSPF] implementation may request a BFD session to be established to a neighbor discovered using the OSPF Hello protocol.

3.2. Operating Modes

BFD has two operating modes that may be selected, as well as an additional function that can be used in combination with the two modes.

The primary mode is known as Asynchronous mode. In this mode, the systems periodically send BFD Control packets to one another, and if a number of those packets in a row are not received by the other system, the session is declared to be down.

The second mode is known as Demand mode. In this mode, it is assumed that a system has an independent way of verifying that it has connectivity to the other system. Once a BFD session is established, such a system may ask the other system to stop sending BFD Control packets, except when the system feels the need to verify connectivity explicitly, in which case a short sequence of BFD Control packets is exchanged, and then the far system quiesces. Demand mode may operate independently in each direction, or simultaneously.

An adjunct to both modes is the Echo function. When the Echo function is active, a stream of BFD Echo packets is transmitted in such a way as to have the other system loop them back through its forwarding path. If a number of packets of the echoed data stream are not received, the session is declared to be down. The Echo function may be used with either Asynchronous or Demand mode. Since the Echo function is handling the task of detection, the rate of periodic transmission of Control packets may be reduced (in the case of Asynchronous mode) or eliminated completely (in the case of Demand mode).

Pure Asynchronous mode is advantageous in that it requires half as many packets to achieve a particular Detection Time as does the Echo function. It is also used when the Echo function cannot be supported for some reason.

The Echo function has the advantage of truly testing only the forwarding path on the remote system. This may reduce round-trip jitter and thus allow more aggressive Detection Times, as well as potentially detecting some classes of failure that might not otherwise be detected.

The Echo function may be enabled individually in each direction. It is enabled in a particular direction only when the system that loops the Echo packets back signals that it will allow it, and when the system that sends the Echo packets decides it wishes to.

Demand mode is useful in situations where the overhead of a periodic protocol might prove onerous, such as a system with a very large number of BFD sessions. It is also useful when the Echo function is being used symmetrically. Demand mode has the disadvantage that Detection Times are essentially driven by the heuristics of the system implementation and are not known to the BFD protocol. Demand

mode may not be used when the path round-trip time is greater than the desired Detection Time, or the protocol will fail to work properly. See section 6.6 for more details.

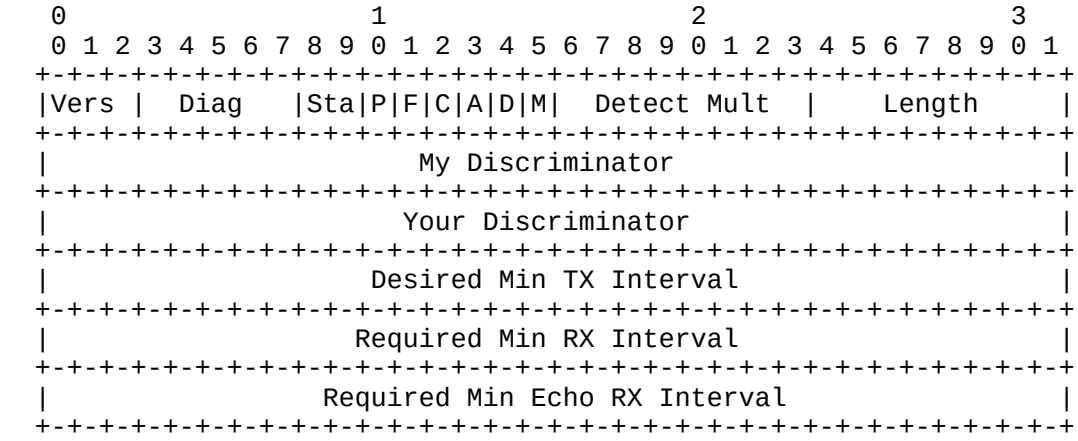
4. BFD Control Packet Format

4.1. Generic BFD Control Packet Format

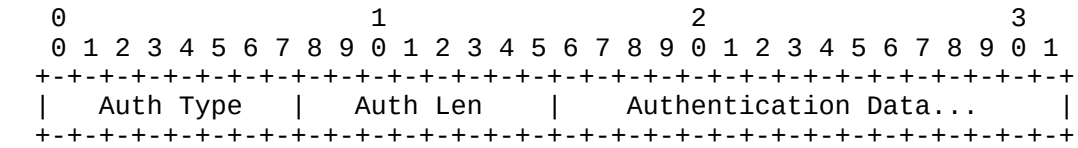
BFD Control packets are sent in an encapsulation appropriate to the environment. The specific encapsulation is outside of the scope of this specification. See the appropriate application document for encapsulation details.

The BFD Control packet has a Mandatory Section and an optional Authentication Section. The format of the Authentication Section, if present, is dependent on the type of authentication in use.

The Mandatory Section of a BFD Control packet has the following format:



An optional Authentication Section MAY be present:



Version (Vers)

The version number of the protocol. This document defines protocol version 1.

Diagnostic (Diag)

A diagnostic code specifying the local system's reason for the last change in session state. Values are:

0 -- No Diagnostic
1 -- Control Detection Time Expired
2 -- Echo Function Failed
3 -- Neighbor Signaled Session Down
4 -- Forwarding Plane Reset
5 -- Path Down
6 -- Concatenated Path Down
7 -- Administratively Down
8 -- Reverse Concatenated Path Down
9-31 -- Reserved for future use

This field allows remote systems to determine the reason that the previous session failed, for example.

State (Sta)

The current BFD session state as seen by the transmitting system. Values are:

0 -- AdminDown
1 -- Down
2 -- Init
3 -- Up

Poll (P)

If set, the transmitting system is requesting verification of connectivity, or of a parameter change, and is expecting a packet with the Final (F) bit in reply. If clear, the transmitting system is not requesting verification.

Final (F)

If set, the transmitting system is responding to a received BFD Control packet that had the Poll (P) bit set. If clear, the transmitting system is not responding to a Poll.

Control Plane Independent (C)

If set, the transmitting system's BFD implementation does not share fate with its control plane (in other words, BFD is implemented in the forwarding plane and can continue to function through disruptions in the control plane). If clear, the transmitting system's BFD implementation shares fate with its control plane.

The use of this bit is application dependent and is outside the scope of this specification. See specific application specifications for details.

Authentication Present (A)

If set, the Authentication Section is present and the session is to be authenticated (see section 6.7 for details).

Demand (D)

If set, Demand mode is active in the transmitting system (the system wishes to operate in Demand mode, knows that the session is Up in both directions, and is directing the remote system to cease the periodic transmission of BFD Control packets). If clear, Demand mode is not active in the transmitting system.

Multipoint (M)

This bit is reserved for future point-to-multipoint extensions to BFD. It MUST be zero on both transmit and receipt.

Detect Mult

Detection time multiplier. The negotiated transmit interval, multiplied by this value, provides the Detection Time for the receiving system in Asynchronous mode.

Length

Length of the BFD Control packet, in bytes.

My Discriminator

A unique, nonzero discriminator value generated by the transmitting system, used to demultiplex multiple BFD sessions between the same pair of systems.

Your Discriminator

The discriminator received from the corresponding remote system. This field reflects back the received value of My Discriminator, or is zero if that value is unknown.

Desired Min TX Interval

This is the minimum interval, in microseconds, that the local system would like to use when transmitting BFD Control packets, less any jitter applied (see section 6.8.2). The value zero is reserved.

Required Min RX Interval

This is the minimum interval, in microseconds, between received BFD Control packets that this system is capable of supporting, less any jitter applied by the sender (see section 6.8.2). If this value is zero, the transmitting system does not want the remote system to send any periodic BFD Control packets.

Required Min Echo RX Interval

This is the minimum interval, in microseconds, between received BFD Echo packets that this system is capable of supporting, less any jitter applied by the sender (see section 6.8.9). If this value is zero, the transmitting system does not support the receipt of BFD Echo packets.

Auth Type

The authentication type in use, if the Authentication Present (A) bit is set.

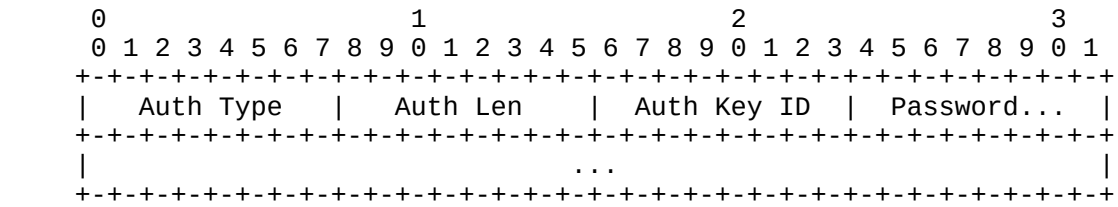
```
   0 - Reserved
   1 - Simple Password
   2 - Keyed MD5
   3 - Meticulous Keyed MD5
   4 - Keyed SHA1
   5 - Meticulous Keyed SHA1
6-255 - Reserved for future use
```

Auth Len

The length, in bytes, of the authentication section, including the Auth Type and Auth Len fields.

4.2. Simple Password Authentication Section Format

If the Authentication Present (A) bit is set in the header, and the Authentication Type field contains 1 (Simple Password), the Authentication Section has the following format:



Auth Type

The Authentication Type, which in this case is 1 (Simple Password).

Auth Len

The length of the Authentication Section, in bytes. For Simple Password authentication, the length is equal to the password length plus three.

Auth Key ID

The authentication key ID in use for this packet. This allows multiple keys to be active simultaneously.

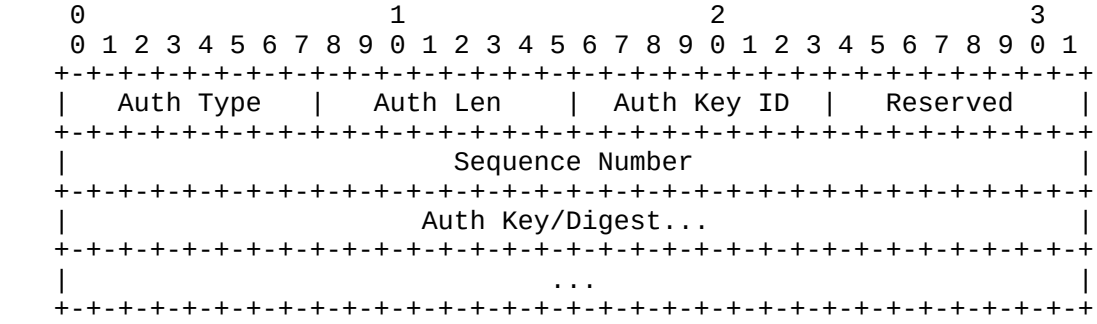
Password

The simple password in use on this session. The password is a binary string, and MUST be from 1 to 16 bytes in length. The password MUST be encoded and configured according to section 6.7.2.

4.3. Keyed MD5 and Meticulous Keyed MD5 Authentication Section Format

The use of MD5-based authentication is strongly discouraged. However, it is documented here for compatibility with existing implementations.

If the Authentication Present (A) bit is set in the header, and the Authentication Type field contains 2 (Keyed MD5) or 3 (Meticulous Keyed MD5), the Authentication Section has the following format:



Auth Type

The Authentication Type, which in this case is 2 (Keyed MD5) or 3 (Meticulous Keyed MD5).

Auth Len

The length of the Authentication Section, in bytes. For Keyed MD5 and Meticulous Keyed MD5 authentication, the length is 24.

Auth Key ID

The authentication key ID in use for this packet. This allows multiple keys to be active simultaneously.

Reserved

This byte MUST be set to zero on transmit, and ignored on receipt.

Sequence Number

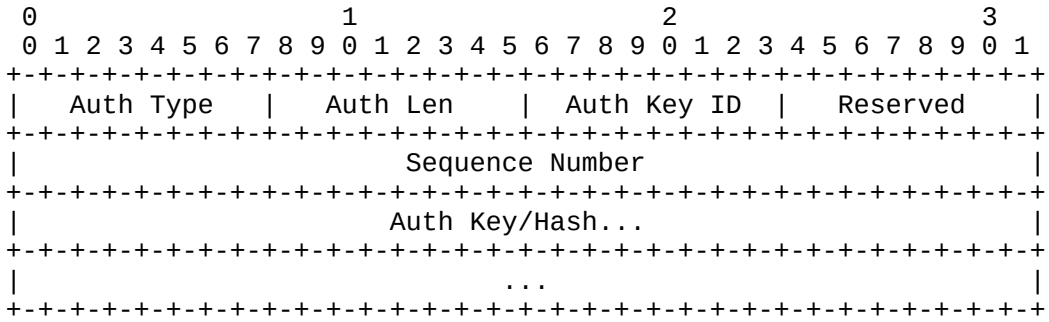
The sequence number for this packet. For Keyed MD5 Authentication, this value is incremented occasionally. For Meticulous Keyed MD5 Authentication, this value is incremented for each successive packet transmitted for a session. This provides protection against replay attacks.

Auth Key/Digest

This field carries the 16-byte MD5 digest for the packet. When the digest is calculated, the shared MD5 key is stored in this field, padded to 16 bytes with trailing zero bytes if needed. The shared key MUST be encoded and configured to section 6.7.3.

4.4. Keyed SHA1 and Meticulous Keyed SHA1 Authentication Section Format

If the Authentication Present (A) bit is set in the header, and the Authentication Type field contains 4 (Keyed SHA1) or 5 (Meticulous Keyed SHA1), the Authentication Section has the following format:



Auth Type

The Authentication Type, which in this case is 4 (Keyed SHA1) or 5 (Meticulous Keyed SHA1).

Auth Len

The length of the Authentication Section, in bytes. For Keyed SHA1 and Meticulous Keyed SHA1 authentication, the length is 28.

Auth Key ID

The authentication key ID in use for this packet. This allows multiple keys to be active simultaneously.

Reserved

This byte MUST be set to zero on transmit and ignored on receipt.

Sequence Number

The sequence number for this packet. For Keyed SHA1 Authentication, this value is incremented occasionally. For Meticulous Keyed SHA1 Authentication, this value is incremented for each successive packet transmitted for a session. This provides protection against replay attacks.

Auth Key/Hash

This field carries the 20-byte SHA1 hash for the packet. When the hash is calculated, the shared SHA1 key is stored in this field, padded to a length of 20 bytes with trailing zero bytes if needed. The shared key MUST be encoded and configured to section 6.7.4.

5. BFD Echo Packet Format

BFD Echo packets are sent in an encapsulation appropriate to the environment. See the appropriate application documents for the specifics of particular environments.

The payload of a BFD Echo packet is a local matter, since only the sending system ever processes the content. The only requirement is that sufficient information is included to demultiplex the received packet to the correct BFD session after it is looped back to the sender. The contents are otherwise outside the scope of this specification.

Some form of authentication SHOULD be included, since Echo packets may be spoofed.

6. Elements of Procedure

This section discusses the normative requirements of the protocol in order to achieve interoperability. It is important for implementors to enforce only the requirements specified in this section, as misguided pedantry has been proven by experience to affect interoperability adversely.

Remember that all references of the form "bfd.Xx" refer to internal state variables (defined in section 6.8.1), whereas all references to "the Xxx field" refer to fields in the protocol packets themselves (defined in section 4).

6.1. Overview

A system may take either an Active role or a Passive role in session initialization. A system taking the Active role MUST send BFD Control packets for a particular session, regardless of whether it has received any BFD packets for that session. A system taking the Passive role MUST NOT begin sending BFD packets for a particular session until it has received a BFD packet for that session, and thus has learned the remote system’s discriminator value. At least one system MUST take the Active role (possibly both). The role that a system takes is specific to the application of BFD, and is outside the scope of this specification.

A session begins with the periodic, slow transmission of BFD Control packets. When bidirectional communication is achieved, the BFD session becomes Up.

Once the BFD session is Up, a system can choose to start the Echo function if it desires and the other system signals that it will allow it. The rate of transmission of Control packets is typically kept low when the Echo function is active.

If the Echo function is not active, the transmission rate of Control packets may be increased to a level necessary to achieve the Detection Time requirements for the session.

Once the session is Up, a system may signal that it has entered Demand mode, and the transmission of BFD Control packets by the remote system ceases. Other means of implying connectivity are used to keep the session alive. If either system wishes to verify bidirectional connectivity, it can initiate a short exchange of BFD Control packets (a "Poll Sequence"; see section 6.5) to do so.

If Demand mode is not active, and no Control packets are received in the calculated Detection Time (see section 6.8.4), the session is declared Down. This is signaled to the remote end via the State (Sta) field in outgoing packets.

If sufficient Echo packets are lost, the session is declared Down in the same manner. See section 6.8.5.

If Demand mode is active and no appropriate Control packets are received in response to a Poll Sequence, the session is declared Down in the same manner. See section 6.6.

If the session goes Down, the transmission of Echo packets (if any) ceases, and the transmission of Control packets goes back to the slow rate.

Once a session has been declared Down, it cannot come back up until the remote end first signals that it is down (by leaving the Up state), thus implementing a three-way handshake.

A session MAY be kept administratively down by entering the AdminDown state and sending an explanatory diagnostic code in the Diagnostic field.

6.2. BFD State Machine

The BFD state machine is quite straightforward. There are three states through which a session normally proceeds: two for establishing a session (Init and Up) and one for tearing down a session (Down). This allows a three-way handshake for both session establishment and session teardown (assuring that both systems are aware of all session state changes). A fourth state (AdminDown) exists so that a session can be administratively put down indefinitely.

Each system communicates its session state in the State (Sta) field in the BFD Control packet, and that received state, in combination with the local session state, drives the state machine.

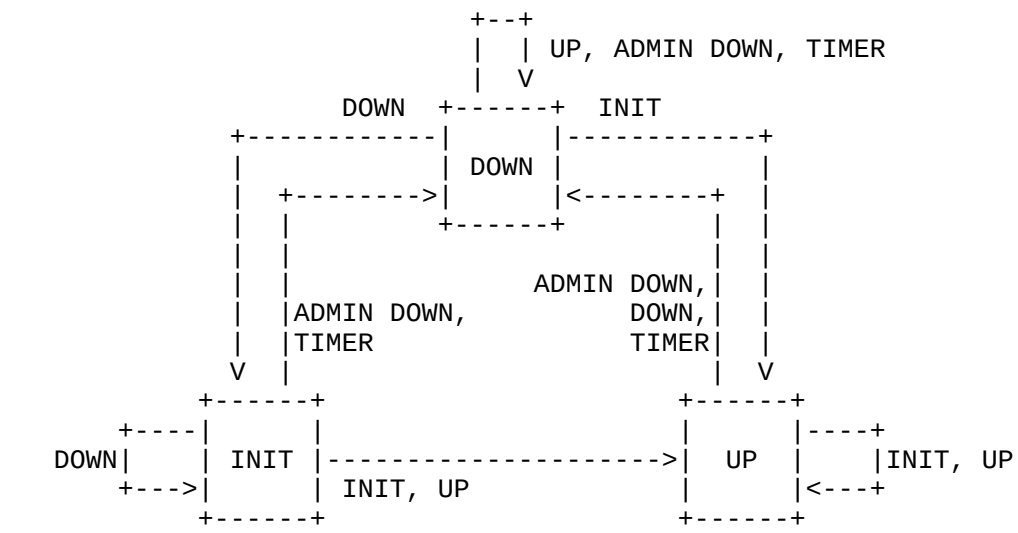
Down state means that the session is down (or has just been created). A session remains in Down state until the remote system indicates that it agrees that the session is down by sending a BFD Control packet with the State field set to anything other than Up. If that packet signals Down state, the session advances to Init state; if that packet signals Init state, the session advances to Up state. Semantically, Down state indicates that the forwarding path is unavailable, and that appropriate actions should be taken by the applications monitoring the state of the BFD session. A system MAY hold a session in Down state indefinitely (by simply refusing to advance the session state). This may be done for operational or administrative reasons, among others.

Init state means that the remote system is communicating, and the local system desires to bring the session up, but the remote system does not yet realize it. A session will remain in Init state until either a BFD Control Packet is received that is signaling Init or Up state (in which case the session advances to Up state) or the Detection Time expires, meaning that communication with the remote system has been lost (in which case the session advances to Down state).

Up state means that the BFD session has successfully been established, and implies that connectivity between the systems is working. The session will remain in the Up state until either connectivity fails or the session is taken down administratively. If either the remote system signals Down state or the Detection Time expires, the session advances to Down state.

AdminDown state means that the session is being held administratively down. This causes the remote system to enter Down state, and remain there until the local system exits AdminDown state. AdminDown state has no semantic implications for the availability of the forwarding path.

The following diagram provides an overview of the state machine. Transitions involving AdminDown state are deleted for clarity (but are fully specified in sections 6.8.6 and 6.8.16). The notation on each arc represents the state of the remote system (as received in the State field in the BFD Control packet) or indicates the expiration of the Detection Timer.



6.3. Demultiplexing and the Discriminator Fields

Since multiple BFD sessions may be running between two systems, there needs to be a mechanism for demultiplexing received BFD packets to the proper session.

Each system MUST choose an opaque discriminator value that identifies each session, and which MUST be unique among all BFD sessions on the system. The local discriminator is sent in the My Discriminator field in the BFD Control packet, and is echoed back in the Your Discriminator field of packets sent from the remote end.

Once the remote end echoes back the local discriminator, all further received packets are demultiplexed based on the Your Discriminator field only (which means that, among other things, the source address field can change or the interface over which the packets are received can change, but the packets will still be associated with the proper session).

The method of demultiplexing the initial packets (in which Your Discriminator is zero) is application dependent, and is thus outside the scope of this specification.

Note that it is permissible for a system to change its discriminator during a session without affecting the session state, since only that system uses its discriminator for demultiplexing purposes (by having the other system reflect it back). The implications on an implementation for changing the discriminator value is outside the scope of this specification.

6.4. The Echo Function and Asymmetry

The Echo function can be run independently in each direction between a pair of systems. For whatever reason, a system may advertise that it is willing to receive (and loop back) Echo packets, but may not wish to ever send any. The fact that a system is sending Echo packets is not directly signaled to the system looping them back.

When a system is using the Echo function, it is advantageous to choose a sedate reception rate for Control packets, since liveness detection is being handled by the Echo packets. This can be controlled by manipulating the Required Min RX Interval field (see section 6.8.3).

If the Echo function is only being run in one direction, the system not running the Echo function will more likely wish to receive fairly rapid Control packets in order to achieve its desired Detection Time. Since BFD allows independent transmission rates in each direction, this is easily accomplished.

A system SHOULD otherwise advertise the lowest value of Required Min RX Interval and Required Min Echo RX Interval that it can under the circumstances, to give the other system more freedom in choosing its transmission rate. Note that a system is committing to be able to receive both streams of packets at the rate it advertises, so this should be taken into account when choosing the values to advertise.

6.5. The Poll Sequence

A Poll Sequence is an exchange of BFD Control packets that is used in some circumstances to ensure that the remote system is aware of parameter changes. It is also used in Demand mode (see section 6.6) to validate bidirectional connectivity.

A Poll Sequence consists of a system sending periodic BFD Control packets with the Poll (P) bit set. When the other system receives a Poll, it immediately transmits a BFD Control packet with the Final (F) bit set, independent of any periodic BFD Control packets it may be sending (see section 6.8.7). When the system sending the Poll sequence receives a packet with Final, the Poll Sequence is terminated, and any subsequent BFD Control packets are sent with the Poll bit cleared. A BFD Control packet MUST NOT have both the Poll (P) and Final (F) bits set.

If periodic BFD Control packets are already being sent (the remote system is not in Demand mode), the Poll Sequence MUST be performed by setting the Poll (P) bit on those scheduled periodic transmissions; additional packets MUST NOT be sent.

After a Poll Sequence is terminated, the system requesting the Poll Sequence will cease the periodic transmission of BFD Control packets if the remote end is in Demand mode; otherwise, it will return to the periodic transmission of BFD Control packets with the Poll (P) bit clear.

Typically, the entire sequence consists of a single packet in each direction, though packet losses or relatively long packet latencies may result in multiple Poll packets to be sent before the sequence terminates.

6.6. Demand Mode

Demand mode is requested independently in each direction by virtue of a system setting the Demand (D) bit in its BFD Control packets. The system receiving the Demand bit ceases the periodic transmission of BFD Control packets. If both systems are operating in Demand mode, no periodic BFD Control packets will flow in either direction.

Demand mode requires that some other mechanism is used to imply continuing connectivity between the two systems. The mechanism used does not have to be the same in both directions, and is outside of the scope of this specification. One possible mechanism is the receipt of traffic from the remote system; another is the use of the Echo function.

When a system in Demand mode wishes to verify bidirectional connectivity, it initiates a Poll Sequence (see section 6.5). If no response is received to a Poll, the Poll is repeated until the Detection Time expires, at which point the session is declared to be Down. Note that if Demand mode is operating only on the local system, the Poll Sequence is performed by simply setting the Poll (P) bit in regular periodic BFD Control packets, as required by section 6.5.

The Detection Time in Demand mode is calculated differently than in Asynchronous mode; it is based on the transmit rate of the local system, rather than the transmit rate of the remote system. This ensures that the Poll Sequence mechanism works properly. See section 6.8.4 for more details.

Note that the Poll mechanism will always fail unless the negotiated Detection Time is greater than the round-trip time between the two systems. Enforcement of this constraint is outside the scope of this specification.

Demand mode MAY be enabled or disabled at any time, independently in each direction, by setting or clearing the Demand (D) bit in the BFD Control packet, without affecting the BFD session state. Note that the Demand bit MUST NOT be set unless both systems perceive the session to be Up (the local system thinks the session is Up, and the remote system last reported Up state in the State (Sta) field of the BFD Control packet).

When the transmitted value of the Demand (D) bit is to be changed, the transmitting system MUST initiate a Poll Sequence in conjunction with changing the bit in order to ensure that both systems are aware of the change.

If Demand mode is active on either or both systems, a Poll Sequence MUST be initiated whenever the contents of the next BFD Control packet to be sent would be different than the contents of the previous packet, with the exception of the Poll (P) and Final (F) bits. This ensures that parameter changes are transmitted to the remote system and that the remote system acknowledges these changes.

Because the underlying detection mechanism is unspecified, and may differ between the two systems, the overall Detection Time characteristics of the path will not be fully known to either system. The total Detection Time for a particular system is the sum of the time prior to the initiation of the Poll Sequence, plus the calculated Detection Time.

Note that if Demand mode is enabled in only one direction, continuous bidirectional connectivity verification is lost (only connectivity in the direction from the system in Demand mode to the other system will be verified). Resolving the issue of one system requesting Demand mode while the other requires continuous bidirectional connectivity verification is outside the scope of this specification.

6.7. Authentication

An optional Authentication Section MAY be present in the BFD Control packet. In its generic form, the purpose of the Authentication Section is to carry all necessary information, based on the authentication type in use, to allow the receiving system to determine the validity of the received packet. The exact mechanism depends on the authentication type in use, but in general the transmitting system will put information in the Authentication

Section that vouches for the packet’s validity, and the receiving system will examine the Authentication Section and either accept the packet for further processing or discard it.

The same authentication type, and any keys or other necessary information, obviously must be in use by the two systems. The negotiation of authentication type, key exchange, etc., are all outside the scope of this specification and are expected to be performed by means outside of the protocol.

Note that in the subsections below, to "accept" a packet means only that the packet has passed authentication; it may in fact be discarded for other reasons as described in the general packet reception rules described in section 6.8.6.

Implementations supporting authentication MUST support both types of SHA1 authentication. Other forms of authentication are optional.

6.7.1. Enabling and Disabling Authentication

It may be desirable to enable or disable authentication on a session without disturbing the session state. The exact mechanism for doing so is outside the scope of this specification. However, it is useful to point out some issues in supporting this mechanism.

In a simple implementation, a BFD session will fail when authentication is either turned on or turned off, because the packet acceptance rules essentially require the local and remote machines to

do so in a more or less synchronized fashion (within the Detection Time) -- a packet with authentication will only be accepted if authentication is "in use" (and likewise packets without authentication).

One possible approach is to build an implementation such that authentication is configured, but not considered "in use" until the first packet containing a matching authentication section is received (providing the necessary synchronization). Likewise, authentication could be configured off, but still considered "in use" until the receipt of the first packet without the authentication section.

In order to avoid security risks, implementations using this method SHOULD only allow the authentication state to be changed at most once without some form of intervention (so that authentication cannot be turned on and off repeatedly simply based on the receipt of BFD Control packets from remote systems). Unless it is desired to enable or disable authentication, an implementation SHOULD NOT allow the authentication state to change based on the receipt of BFD Control packets.

6.7.2. Simple Password Authentication

The most straightforward (and weakest) form of authentication is Simple Password Authentication. In this method of authentication, one or more Passwords (with corresponding Key IDs) are configured in each system and one of these Password/ID pairs is carried in each BFD Control packet. The receiving system accepts the packet if the Password and Key ID matches one of the Password/ID pairs configured in that system.

Transmission Using Simple Password Authentication

The currently selected password and Key ID for the session MUST be stored in the Authentication Section of each outgoing BFD Control packet. The Auth Type field MUST be set to 1 (Simple Password). The Auth Len field MUST be set to the proper length (4 to 19 bytes).

The password is a binary string, and MUST be 1 to 16 bytes in length. For interoperability, the management interface by which the password is configured MUST accept ASCII strings, and SHOULD also allow for the configuration of any arbitrary binary string in hexadecimal form. Other configuration methods MAY be supported.

Reception Using Simple Password Authentication

If the received BFD Control packet does not contain an Authentication Section, or the Auth Type is not 1 (Simple Password), then the received packet MUST be discarded.

If the Auth Key ID field does not match the ID of a configured password, the received packet MUST be discarded.

If the Auth Len field is not equal to the length of the password selected by the key ID, plus three, the packet MUST be discarded.

If the Password field does not match the password selected by the key ID, the packet MUST be discarded.

Otherwise, the packet MUST be accepted.

6.7.3. Keyed MD5 and Meticulous Keyed MD5 Authentication

The Keyed MD5 and Meticulous Keyed MD5 Authentication mechanisms are very similar to those used in other protocols. In these methods of authentication, one or more secret keys (with corresponding key IDs) are configured in each system. One of the keys is included in an MD5 [MD5] digest calculated over the outgoing BFD Control packet, but the Key itself is not carried in the packet. To help avoid replay attacks, a sequence number is also carried in each packet. For Keyed MD5, the sequence number is occasionally incremented. For Meticulous Keyed MD5, the sequence number is incremented on every packet.

The receiving system accepts the packet if the key ID matches one of the configured Keys, an MD5 digest including the selected key matches that carried in the packet, and the sequence number is greater than or equal to the last sequence number received (for Keyed MD5), or strictly greater than the last sequence number received (for Meticulous Keyed MD5).

Transmission Using Keyed MD5 and Meticulous Keyed MD5 Authentication

The Auth Type field MUST be set to 2 (Keyed MD5) or 3 (Meticulous Keyed MD5). The Auth Len field MUST be set to 24. The Auth Key ID field MUST be set to the ID of the current authentication key. The Sequence Number field MUST be set to bfd.XmitAuthSeq.

The authentication key value is a binary string of up to 16 bytes, and MUST be placed into the Auth Key/Digest field, padded with trailing zero bytes as necessary. For interoperability, the management interface by which the key is configured MUST accept

ASCII strings, and SHOULD also allow for the configuration of any arbitrary binary string in hexadecimal form. Other configuration methods MAY be supported.

An MD5 digest MUST be calculated over the entire BFD Control packet. The resulting digest MUST be stored in the Auth Key/Digest field prior to transmission (replacing the secret key, which MUST NOT be carried in the packet).

For Keyed MD5, bfd.XmitAuthSeq MAY be incremented in a circular fashion (when treated as an unsigned 32-bit value). bfd.XmitAuthSeq SHOULD be incremented when the session state changes, or when the transmitted BFD Control packet carries different contents than the previously transmitted packet. The decision as to when to increment bfd.XmitAuthSeq is outside the scope of this specification. See the section titled "Security Considerations" below for a discussion.

For Meticulous Keyed MD5, bfd.XmitAuthSeq MUST be incremented in a circular fashion (when treated as an unsigned 32-bit value).

Receipt Using Keyed MD5 and Meticulous Keyed MD5 Authentication

If the received BFD Control packet does not contain an Authentication Section, or the Auth Type is not correct (2 for Keyed MD5 or 3 for Meticulous Keyed MD5), then the received packet MUST be discarded.

If the Auth Key ID field does not match the ID of a configured authentication key, the received packet MUST be discarded.

If the Auth Len field is not equal to 24, the packet MUST be discarded.

If bfd.AuthSeqKnown is 1, examine the Sequence Number field. For Keyed MD5, if the sequence number lies outside of the range of bfd.RcvAuthSeq to bfd.RcvAuthSeq+(3*Detect Mult) inclusive (when treated as an unsigned 32-bit circular number space), the received packet MUST be discarded. For Meticulous Keyed MD5, if the sequence number lies outside of the range of bfd.RcvAuthSeq+1 to bfd.RcvAuthSeq+(3*Detect Mult) inclusive (when treated as an unsigned 32-bit circular number space) the received packet MUST be discarded.

Otherwise (bfd.AuthSeqKnown is 0), bfd.AuthSeqKnown MUST be set to 1, and bfd.RcvAuthSeq MUST be set to the value of the received Sequence Number field.

Replace the contents of the Auth Key/Digest field with the authentication key selected by the received Auth Key ID field. If the MD5 digest of the entire BFD Control packet is equal to the received value of the Auth Key/Digest field, the received packet MUST be accepted. Otherwise (the digest does not match the Auth Key/Digest field), the received packet MUST be discarded.

6.7.4. Keyed SHA1 and Meticulous Keyed SHA1 Authentication

The Keyed SHA1 and Meticulous Keyed SHA1 Authentication mechanisms are very similar to those used in other protocols. In these methods of authentication, one or more secret keys (with corresponding key IDs) are configured in each system. One of the keys is included in a SHA1 [SHA1] hash calculated over the outgoing BFD Control packet, but the key itself is not carried in the packet. To help avoid replay attacks, a sequence number is also carried in each packet. For Keyed SHA1, the sequence number is occasionally incremented. For Meticulous Keyed SHA1, the sequence number is incremented on every packet.

The receiving system accepts the packet if the key ID matches one of the configured keys, a SHA1 hash including the selected key matches that carried in the packet, and if the sequence number is greater than or equal to the last sequence number received (for Keyed SHA1), or strictly greater than the last sequence number received (for Meticulous Keyed SHA1).

Transmission Using Keyed SHA1 and Meticulous Keyed SHA1 Authentication

The Auth Type field MUST be set to 4 (Keyed SHA1) or 5 (Meticulous Keyed SHA1). The Auth Len field MUST be set to 28. The Auth Key ID field MUST be set to the ID of the current authentication key. The Sequence Number field MUST be set to bfd.XmitAuthSeq.

The authentication key value is a binary string of up to 20 bytes, and MUST be placed into the Auth Key/Hash field, padding with trailing zero bytes as necessary. For interoperability, the management interface by which the key is configured MUST accept ASCII strings, and SHOULD also allow for the configuration of any arbitrary binary string in hexadecimal form. Other configuration methods MAY be supported.

A SHA1 hash MUST be calculated over the entire BFD control packet. The resulting hash MUST be stored in the Auth Key/Hash field prior to transmission (replacing the secret key, which MUST NOT be carried in the packet).

For Keyed SHA1, bfd.XmitAuthSeq MAY be incremented in a circular fashion (when treated as an unsigned 32-bit value). bfd.XmitAuthSeq SHOULD be incremented when the session state changes, or when the transmitted BFD Control packet carries different contents than the previously transmitted packet. The decision as to when to increment bfd.XmitAuthSeq is outside the scope of this specification. See the section titled "Security Considerations" below for a discussion.

For Meticulous Keyed SHA1, bfd.XmitAuthSeq MUST be incremented in a circular fashion (when treated as an unsigned 32-bit value).

Receipt Using Keyed SHA1 and Meticulous Keyed SHA1 Authentication

If the received BFD Control packet does not contain an Authentication Section, or the Auth Type is not correct (4 for Keyed SHA1 or 5 for Meticulous Keyed SHA1), then the received packet MUST be discarded.

If the Auth Key ID field does not match the ID of a configured authentication key, the received packet MUST be discarded.

If the Auth Len field is not equal to 28, the packet MUST be discarded.

If bfd.AuthSeqKnown is 1, examine the Sequence Number field. For Keyed SHA1, if the sequence number lies outside of the range of bfd.RcvAuthSeq to bfd.RcvAuthSeq+(3*Detect Mult) inclusive (when treated as an unsigned 32-bit circular number space), the received packet MUST be discarded. For Meticulous Keyed SHA1, if the sequence number lies outside of the range of bfd.RcvAuthSeq+1 to bfd.RcvAuthSeq+(3*Detect Mult) inclusive (when treated as an unsigned 32-bit circular number space, the received packet MUST be discarded.

Otherwise (bfd.AuthSeqKnown is 0), bfd.AuthSeqKnown MUST be set to 1, bfd.RcvAuthSeq MUST be set to the value of the received Sequence Number field, and the received packet MUST be accepted.

Replace the contents of the Auth Key/Hash field with the authentication key selected by the received Auth Key ID field. If the SHA1 hash of the entire BFD Control packet is equal to the received value of the Auth Key/Hash field, the received packet MUST be accepted. Otherwise (the hash does not match the Auth Key/Hash field), the received packet MUST be discarded.

6.8. Functional Specifics

The following section of this specification is normative. The means by which this specification is achieved is outside the scope of this specification.

When a system is said to have "the Echo function active" it means that the system is sending BFD Echo packets, implying that the session is Up and the other system has signaled its willingness to loop back Echo packets.

When the local system is said to have "Demand mode active," it means that bfd.DemandMode is 1 in the local system (see section 6.8.1), the session is Up, and the remote system is signaling that the session is in state Up.

When the remote system is said to have "Demand mode active," it means that bfd.RemoteDemandMode is 1 (the remote system set the Demand (D) bit in the last received BFD Control packet), the session is Up, and the remote system is signaling that the session is in state Up.

6.8.1. State Variables

A minimum amount of information about a session needs to be tracked in order to achieve the elements of procedure described here. The following is a set of state variables that are helpful in describing the mechanisms of BFD. Any means of tracking this state may be used so long as the protocol behaves as described.

When the text refers to initializing a state variable, this takes place only at the time that the session (and the corresponding state variables) is created. The state variables are subsequently manipulated by the state machine and are never reinitialized, even if the session fails and is reestablished.

Once session state is created, and at least one BFD Control packet is received from the remote end, it MUST be preserved for at least one Detection Time (see section 6.8.4) subsequent to the receipt of the last BFD Control packet, regardless of the session state. This preserves timing parameters in case the session flaps. A system MAY preserve session state longer than this. The preservation or destruction of session state when no BFD Control packets for this session have been received from the remote system is outside the scope of this specification.

All state variables in this specification are of the form "bfd.Xx" and should not be confused with fields carried in the protocol packets, which are always spelled out to match the names in section 4.

bfd.SessionState

The perceived state of the session (Init, Up, Down, or AdminDown). The exact action taken when the session state changes is outside the scope of this specification, though it is expected that this state change (particularly, to and from Up state) is reported to other components of the system. This variable MUST be initialized to Down.

bfd.RemoteSessionState

The session state last reported by the remote system in the State (Sta) field of the BFD Control packet. This variable MUST be initialized to Down.

bfd.LocalDiscr

The local discriminator for this BFD session, used to uniquely identify it. It MUST be unique across all BFD sessions on this system, and nonzero. It SHOULD be set to a random (but still unique) value to improve security. The value is otherwise outside the scope of this specification.

bfd.RemoteDiscr

The remote discriminator for this BFD session. This is the discriminator chosen by the remote system, and is totally opaque to the local system. This MUST be initialized to zero. If a period of a Detection Time passes without the receipt of a valid, authenticated BFD packet from the remote system, this variable MUST be set to zero.

bfd.LocalDiag

The diagnostic code specifying the reason for the most recent change in the local session state. This MUST be initialized to zero (No Diagnostic).

bfd.DesiredMinTxInterval

The minimum interval, in microseconds, between transmitted BFD Control packets that this system would like to use at the current time, less any jitter applied (see section 6.8.2). The actual interval is negotiated between the two systems. This MUST be initialized to a value of at least one second (1,000,000 microseconds) according to the rules described in section 6.8.3. The setting of this variable is otherwise outside the scope of this specification.

bfd.RequiredMinRxInterval

The minimum interval, in microseconds, between received BFD Control packets that this system requires, less any jitter applied by the sender (see section 6.8.2). The setting of this variable is outside the scope of this specification. A value of zero means that this system does not want to receive any periodic BFD Control packets. See section 6.8.18 for details.

bfd.RemoteMinRxInterval

The last value of Required Min RX Interval received from the remote system in a BFD Control packet. This variable MUST be initialized to 1.

bfd.DemandMode

Set to 1 if the local system wishes to use Demand mode, or 0 if not.

bfd.RemoteDemandMode

Set to 1 if the remote system wishes to use Demand mode, or 0 if not. This is the value of the Demand (D) bit in the last received BFD Control packet. This variable MUST be initialized to zero.

bfd.DetectMult

The desired Detection Time multiplier for BFD Control packets on the local system. The negotiated Control packet transmission interval, multiplied by this variable, will be the Detection Time for this session (as seen by the remote system). This variable MUST be a nonzero integer, and is otherwise outside the scope of this specification. See section 6.8.4 for further information.

bfd.AuthType

The authentication type in use for this session, as defined in section 4.1, or zero if no authentication is in use.

bfd.RcvAuthSeq

A 32-bit unsigned integer containing the last sequence number for Keyed MD5 or SHA1 Authentication that was received. The initial value is unimportant.

bfd.XmitAuthSeq

A 32-bit unsigned integer containing the next sequence number for Keyed MD5 or SHA1 Authentication to be transmitted. This variable MUST be initialized to a random 32-bit value.

bfd.AuthSeqKnown

Set to 1 if the next sequence number for Keyed MD5 or SHA1 authentication expected to be received is known, or 0 if it is not known. This variable MUST be initialized to zero.

This variable MUST be set to zero after no packets have been received on this session for at least twice the Detection Time. This ensures that the sequence number can be resynchronized if the remote system restarts.

6.8.2. Timer Negotiation

The time values used to determine BFD packet transmission intervals and the session Detection Time are continuously negotiated, and thus may be changed at any time. The negotiation and time values are independent in each direction for each session.

Each system reports in the BFD Control packet how rapidly it would like to transmit BFD packets, as well as how rapidly it is prepared to receive them. This allows either system to unilaterally determine the maximum packet rate (minimum interval) in both directions.

See section 6.8.7 for the details of packet transmission timing and negotiation.

6.8.3. Timer Manipulation

The time values used to determine BFD packet transmission intervals and the session Detection Time may be modified at any time without affecting the state of the session. When the timer parameters are changed for any reason, the requirements of this section apply.

If either bfd.DesiredMinTxInterval is changed or bfd.RequiredMinRxInterval is changed, a Poll Sequence MUST be initiated (see section 6.5). If the timing is such that a system receiving a Poll Sequence wishes to change the parameters described in this paragraph, the new parameter values MAY be carried in packets with the Final (F) bit set, even if the Poll Sequence has not yet been sent.

If bfd.DesiredMinTxInterval is increased and bfd.SessionState is Up, the actual transmission interval used MUST NOT change until the Poll Sequence described above has terminated. This is to ensure that the remote system updates its Detection Time before the transmission interval increases.

If bfd.RequiredMinRxInterval is reduced and bfd.SessionState is Up, the previous value of bfd.RequiredMinRxInterval MUST be used when calculating the Detection Time for the remote system until the Poll Sequence described above has terminated. This is to ensure that the remote system is transmitting packets at the higher rate (and those packets are being received) prior to the Detection Time being reduced.

When bfd.SessionState is not Up, the system MUST set bfd.DesiredMinTxInterval to a value of not less than one second (1,000,000 microseconds). This is intended to ensure that the bandwidth consumed by BFD sessions that are not Up is negligible, particularly in the case where a neighbor may not be running BFD.

If the local system reduces its transmit interval due to bfd.RemoteMinRxInterval being reduced (the remote system has advertised a reduced value in Required Min RX Interval), and the remote system is not in Demand mode, the local system MUST honor the new interval immediately. In other words, the local system cannot wait longer than the new interval between the previous packet transmission and the next one. If this interval has already passed since the last transmission (because the new interval is significantly shorter), the local system MUST send the next periodic BFD Control packet as soon as practicable.

When the Echo function is active, a system SHOULD set bfd.RequiredMinRxInterval to a value of not less than one second (1,000,000 microseconds). This is intended to keep received BFD Control traffic at a negligible level, since the actual detection function is being performed using BFD Echo packets.

In any case other than those explicitly called out above, timing parameter changes MUST be effected immediately (changing the transmission rate and/or the Detection Time).

Note that the Poll Sequence mechanism is ambiguous if more than one parameter change is made that would require its use, and those multiple changes are spread across multiple packets (since the semantics of the returning Final are unclear). Therefore, if multiple changes are made that require the use of a Poll Sequence, there are three choices: 1) they MUST be communicated in a single BFD Control packet (so the semantics of the Final reply are clear), or 2) sufficient time must have transpired since the Poll Sequence was completed to disambiguate the situation (at least a round trip time since the last Poll was transmitted) prior to the initiation of another Poll Sequence, or 3) an additional BFD Control packet with the Final (F) bit *clear* MUST be received after the Poll Sequence has completed prior to the initiation of another Poll Sequence (this option is not available when Demand mode is active).

6.8.4. Calculating the Detection Time

The Detection Time (the period of time without receiving BFD packets after which the session is determined to have failed) is not carried explicitly in the protocol. Rather, it is calculated independently in each direction by the receiving system based on the negotiated transmit interval and the detection multiplier. Note that there may be different Detection Times in each direction.

The calculation of the Detection Time is slightly different when in Demand mode versus Asynchronous mode.

In Asynchronous mode, the Detection Time calculated in the local system is equal to the value of Detect Mult received from the remote system, multiplied by the agreed transmit interval of the remote system (the greater of bfd.RequiredMinRxInterval and the last received Desired Min TX Interval). The Detect Mult value is (roughly speaking, due to jitter) the number of packets that have to be missed in a row to declare the session to be down.

If Demand mode is not active, and a period of time equal to the Detection Time passes without receiving a BFD Control packet from the remote system, and bfd.SessionState is Init or Up, the session has gone down -- the local system MUST set bfd.SessionState to Down and bfd.LocalDiag to 1 (Control Detection Time Expired).

In Demand mode, the Detection Time calculated in the local system is equal to bfd.DetectMult, multiplied by the agreed transmit interval of the local system (the greater of bfd.DesiredMinTxInterval and bfd.RemoteMinRxInterval). bfd.DetectMult is (roughly speaking, due to jitter) the number of packets that have to be missed in a row to declare the session to be down.

If Demand mode is active, and a period of time equal to the Detection Time passes after the initiation of a Poll Sequence (the transmission of the first BFD Control packet with the Poll bit set), the session has gone down -- the local system MUST set bfd.SessionState to Down, and bfd.LocalDiag to 1 (Control Detection Time Expired).

(Note that a packet is considered to have been received, for the purposes of Detection Time expiration, only if it has not been "discarded" according to the rules of section 6.8.6).

6.8.5. Detecting Failures with the Echo Function

When the Echo function is active and a sufficient number of Echo packets have not arrived as they should, the session has gone down -- the local system MUST set bfd.SessionState to Down and bfd.LocalDiag to 2 (Echo Function Failed).

The means by which the Echo function failures are detected is outside of the scope of this specification. Any means that will detect a communication failure are acceptable.

6.8.6. Reception of BFD Control Packets

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

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

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

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

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

If the Multipoint (M) bit is nonzero, the packet MUST be discarded.

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

If the Your Discriminator field is nonzero, it MUST be used to select the session with which this BFD packet is associated. If no session is found, the packet MUST be discarded.

If the Your Discriminator field is zero and the State field is not Down or AdminDown, the packet MUST be discarded.

If the Your Discriminator field is zero, the session MUST be selected based on some combination of other fields, possibly including source addressing information, the My Discriminator field, and the interface over which the packet was received. The exact method of selection is application specific and is thus outside the scope of this specification. If a matching session is not found, a new session MAY be created, or the packet MAY be discarded. This choice is outside the scope of this specification.

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

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

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

Set bfd.RemoteDiscr to the value of My Discriminator.

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

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

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

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

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

Update the transmit interval as described in section 6.8.2.

Update the Detection Time as described in section 6.8.4.

If bfd.SessionState is AdminDown

    Discard the packet

If received state is AdminDown
    If bfd.SessionState is not Down
        Set bfd.LocalDiag to 3 (Neighbor signaled
            session down)
        Set bfd.SessionState to Down

Else

    If bfd.SessionState is Down
        If received State is Down
            Set bfd.SessionState to Init
        Else if received State is Init
            Set bfd.SessionState to Up

    Else if bfd.SessionState is Init
        If received State is Init or Up
            Set bfd.SessionState to Up

    Else (bfd.SessionState is Up)
        If received State is Down
            Set bfd.LocalDiag to 3 (Neighbor signaled
                session down)
            Set bfd.SessionState to Down

Check to see if Demand mode should become active or not (see
section 6.6).

If bfd.RemoteDemandMode is 1, bfd.SessionState is Up, and
bfd.RemoteSessionState is Up, Demand mode is active on the remote
system and the local system MUST cease the periodic transmission
of BFD Control packets (see section 6.8.7).
```

If bfd.RemoteDemandMode is 0, or bfd.SessionState is not Up, or bfd.RemoteSessionState is not Up, Demand mode is not active on the remote system and the local system MUST send periodic BFD Control packets (see section 6.8.7).

If the Poll (P) bit is set, send a BFD Control packet to the remote system with the Poll (P) bit clear, and the Final (F) bit set (see section 6.8.7).

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

6.8.7. Transmitting BFD Control Packets

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

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

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

The transmit interval MUST be recalculated whenever bfd.DesiredMinTxInterval changes, or whenever bfd.RemoteMinRxInterval changes, and is equal to the greater of those two values. See sections 6.8.2 and 6.8.3 for details on transmit timers.

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

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

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

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 set as soon as practicable, 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.

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

A BFD Control packet SHOULD be transmitted during the interval between periodic Control packet transmissions when the contents of that packet would differ from that in the previously transmitted packet (other than the Poll and Final bits) in order to more rapidly communicate a change in state.

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

Version

Set to the current version number (1).

Diagnostic (Diag)

Set to bfd.LocalDiag.

State (Sta)

Set to the value indicated by bfd.SessionState.

Poll (P)

Set to 1 if the local system is sending a Poll Sequence, or 0 if not.

Final (F)

Set to 1 if the local system is responding to a Control packet received with the Poll (P) bit set, or 0 if not.

Control Plane Independent (C)

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

Authentication Present (A)

Set to 1 if authentication is in use on this session (bfd.AuthType is nonzero), or 0 if not.

Demand (D)

Set to bfd.DemandMode if bfd.SessionState is Up and bfd.RemoteSessionState is Up. Otherwise, it is set to 0.

Multipoint (M)

Set to 0.

Detect Mult

Set to bfd.DetectMult.

Length

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

My Discriminator

Set to bfd.LocalDiscr.

Your Discriminator

Set to bfd.RemoteDiscr.

Desired Min TX Interval

Set to bfd.DesiredMinTxInterval.

Required Min RX Interval

Set to bfd.RequiredMinRxInterval.

Required Min Echo RX Interval

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

Authentication Section

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

6.8.8. Reception of BFD Echo Packets

A received BFD Echo packet MUST be demultiplexed to the appropriate session for processing. A means of detecting missing Echo packets MUST be implemented, which most likely involves processing of the Echo packets that are received. The processing of received Echo packets is otherwise outside the scope of this specification.

6.8.9. Transmission of BFD Echo Packets

BFD Echo packets MUST NOT be transmitted when bfd.SessionState is not Up. BFD Echo packets MUST NOT be transmitted unless the last BFD Control packet received from the remote system contains a nonzero value in Required Min Echo RX Interval.

BFD Echo packets MAY be transmitted when bfd.SessionState is Up. The interval between transmitted BFD Echo packets MUST NOT be less than the value advertised by the remote system in Required Min Echo RX Interval, except as follows:

A 25% jitter MAY be applied to the rate of transmission, such that the actual interval MAY be between 75% and 100% of the advertised value. A single BFD Echo packet MAY be transmitted between normally scheduled Echo transmission intervals.

The transmission of BFD Echo packets is otherwise outside the scope of this specification.

6.8.10. Min Rx Interval Change

When it is desired to change the rate at which BFD Control packets arrive from the remote system, bfd.RequiredMinRxInterval can be changed at any time to any value. The new value will be transmitted in the next outgoing Control packet, and the remote system will adjust accordingly. See section 6.8.3 for further requirements.

6.8.11. Min Tx Interval Change

When it is desired to change the rate at which BFD Control packets are transmitted to the remote system (subject to the requirements of the neighboring system), bfd.DesiredMinTxInterval can be changed at any time to any value. The rules in section 6.8.3 apply.

6.8.12. Detect Multiplier Change

When it is desired to change the detect multiplier, the value of bfd.DetectMult can be changed to any nonzero value. The new value will be transmitted with the next BFD Control packet, and the use of a Poll Sequence is not necessary. See section 6.6 for additional requirements.

6.8.13. Enabling or Disabling The Echo Function

If it is desired to start or stop the transmission of BFD Echo packets, this MAY be done at any time (subject to the transmission requirements detailed in section 6.8.9).

If it is desired to enable or disable the looping back of received BFD Echo packets, this MAY be done at any time by changing the value of Required Min Echo RX Interval to zero or nonzero in outgoing BFD Control packets.

6.8.14. Enabling or Disabling Demand Mode

If it is desired to start or stop Demand mode, this MAY be done at any time by setting bfd.DemandMode to the proper value. Demand mode will subsequently become active under the rules described in section 6.6.

If Demand mode is no longer active on the remote system, the local system MUST begin transmitting periodic BFD Control packets as described in section 6.8.7.

6.8.15. Forwarding Plane Reset

When the forwarding plane in the local system is reset for some reason, such that the remote system can no longer rely on the local forwarding state, the local system MUST set bfd.LocalDiag to 4 (Forwarding Plane Reset), and set bfd.SessionState to Down.

6.8.16. Administrative Control

There may be circumstances where it is desirable to administratively enable or disable a BFD session. When this is desired, the following procedure MUST be followed:

```
If enabling session
   Set bfd.SessionState to Down

Else
   Set bfd.SessionState to AdminDown
   Set bfd.LocalDiag to an appropriate value
   Cease the transmission of BFD Echo packets
```

If signaling is received from outside BFD that the underlying path has failed, an implementation MAY administratively disable the session with the diagnostic Path Down.

Other scenarios MAY use the diagnostic Administratively Down.

BFD Control packets SHOULD be transmitted for at least a Detection Time after transitioning to AdminDown state in order to ensure that the remote system is aware of the state change. BFD Control packets MAY be transmitted indefinitely after transitioning to AdminDown state in order to maintain session state in each system (see section 6.8.18 below).

6.8.17. Concatenated Paths

If the path being monitored by BFD is concatenated with other paths (connected end-to-end in series), it may be desirable to propagate the indication of a failure of one of those paths across the BFD session (providing an interworking function for liveness monitoring between BFD and other technologies).

Two diagnostic codes are defined for this purpose: Concatenated Path Down and Reverse Concatenated Path Down. The first propagates forward path failures (in which the concatenated path fails in the direction toward the interworking system), and the second propagates

reverse path failures (in which the concatenated path fails in the direction away from the interworking system, assuming a bidirectional link).

A system MAY signal one of these failure states by simply setting bfd.LocalDiag to the appropriate diagnostic code. Note that the BFD session is not taken down. If Demand mode is not active on the remote system, no other action is necessary, as the diagnostic code will be carried via the periodic transmission of BFD Control packets. If Demand mode is active on the remote system (the local system is not transmitting periodic BFD Control packets), a Poll Sequence MUST be initiated to ensure that the diagnostic code is transmitted. Note that if the BFD session subsequently fails, the diagnostic code will be overwritten with a code detailing the cause of the failure. It is up to the interworking agent to perform the above procedure again, once the BFD session reaches Up state, if the propagation of the concatenated path failure is to resume.

6.8.18. Holding Down Sessions

A system MAY choose to prevent a BFD session from being established. One possible reason might be to manage the rate at which sessions are established. This can be done by holding the session in Down or AdminDown state, as appropriate.

There are two related mechanisms that are available to help with this task. First, a system is REQUIRED to maintain session state (including timing parameters), even when a session is down, until a Detection Time has passed without the receipt of any BFD Control packets. This means that a system may take down a session and transmit an arbitrarily large value in the Required Min RX Interval field to control the rate at which it receives packets.

Additionally, a system MAY transmit a value of zero for Required Min RX Interval to indicate that the remote system should send no packets whatsoever.

So long as the local system continues to transmit BFD Control packets, the remote system is obligated to obey the value carried in Required Min RX Interval. If the remote system does not receive any BFD Control packets for a Detection Time, it SHOULD reset bfd.RemoteMinRxInterval to its initial value of 1 (per section 6.8.1, since it is no longer required to maintain previous session state) and then can transmit at its own rate.

7. Operational Considerations

BFD is likely to be deployed as a critical part of network infrastructure. As such, care should be taken to avoid disruption.

Obviously, any mechanism that blocks BFD packets, such as firewalls or other policy processes, will cause BFD to fail.

Mechanisms that control packet scheduling, such as policers, traffic shapers, priority queueing, etc., have the potential of impacting BFD operations if the Detection Time is similar in scale to the scheduled packet transmit or receive rate. The delivery of BFD packets is time-critical, relative to the magnitude of the Detection Time, so this may need to be taken into account in implementation and deployment, particularly when very short Detection Times are to be used.

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 is outside the scope of this specification, and the requirements of this mechanism may differ depending on how BFD is deployed, and how it interacts with other parts of the system (for example, exponential backoff may not be appropriate in cases where routing protocols are interacting closely with BFD).

Note that "congestion" is not only a traffic phenomenon, but also a computational one. It is possible for systems with a large number of BFD sessions and/or very short packet intervals to become CPU-bound. As such, a congestion control algorithm SHOULD be used even across single hops in order to avoid the possibility of catastrophic system collapse, as such failures have been seen repeatedly in other periodic Hello-based protocols.

The mechanisms for detecting congestion are outside the scope of this specification, but may include the detection of lost BFD Control packets (by virtue of holes in the authentication sequence number space, or by BFD session failure) or other means.

The mechanisms for reducing BFD’s traffic load are the control of the local and remote packet transmission rate via the Min RX Interval and Min TX Interval fields.

Note that any mechanism that increases the transmit or receive intervals will increase the Detection Time for the session.

It is worth noting that a single BFD session does not consume a large amount of bandwidth. An aggressive session that achieves a detection time of 50 milliseconds, by using a transmit interval of 16.7 milliseconds and a detect multiplier of 3, will generate 60 packets per second. The maximum length of each packet on the wire is on the order of 100 bytes, for a total of around 48 kilobits per second of bandwidth consumption in each direction.

8. IANA Considerations

This document defines two registries administered by IANA. The first is titled "BFD Diagnostic Codes" (see section 4.1). Initial values for the BFD Diagnostic Code registry are given below. Further assignments are to be made through Expert Review [IANA-CONSIDERATIONS]. Assignments consist of a BFD Diagnostic Code name and its associated value.

Value	BFD Diagnostic Code Name
0	No Diagnostic
1	Control Detection Time Expired
2	Echo Function Failed
3	Neighbor Signaled Session Down
4	Forwarding Plane Reset
5	Path Down
6	Concatenated Path Down
7	Administratively Down
8	Reverse Concatenated Path Down
9-31	Unassigned

The second registry is titled "BFD Authentication Types" (see section 4.1). Initial values for the BFD Authentication Type registry are given below. Further assignments are to be made through Expert Review [IANA-CONSIDERATIONS]. Assignments consist of a BFD Authentication Type Code name and its associated value.

Value	BFD Authentication Type Name
0	Reserved
1	Simple Password
2	Keyed MD5
3	Meticulous Keyed MD5
4	Keyed SHA1
5	Meticulous Keyed SHA1
6-255	Unassigned

9. Security Considerations

As BFD may be tied into the stability of the network infrastructure (such as routing protocols), the effects of an attack on a BFD session may be very serious: a link may be falsely declared to be down, or falsely declared to be up; in either case, the effect is denial of service.

An attacker who is in complete control of the link between the systems can easily drop all BFD packets but forward everything else (causing the link to be falsely declared down), or forward only the BFD packets but nothing else (causing the link to be falsely declared up). This attack cannot be prevented by BFD.

To mitigate threats from less capable attackers, BFD specifies two mechanisms to prevent spoofing of BFD Control packets. The Generalized TTL Security Mechanism [GTSM] uses the time to live (TTL) or Hop Count to prevent off-link attackers from spoofing packets. The Authentication Section authenticates the BFD Control packets. These mechanisms are described in more detail below.

When a BFD session is directly connected across a single link (physical, or a secure tunnel such as IPsec), the TTL or Hop Count MUST be set to the maximum on transmit, and checked to be equal to the maximum value on reception (and the packet dropped if this is not the case). See [GTSM] for more information on this technique. If BFD is run across multiple hops or an insecure tunnel (such as Generic Routing Encapsulation (GRE)), the Authentication Section SHOULD be utilized.

The level of security provided by the Authentication Section varies based on the authentication type used. Simple Password authentication is obviously only as secure as the secrecy of the passwords used, and should be considered only if the BFD session is guaranteed to be run over an infrastructure not subject to packet interception. Its chief advantage is that it minimizes the computational effort required for authentication.

Keyed MD5 Authentication is much stronger than Simple Password Authentication since the keys cannot be discerned by intercepting packets. It is vulnerable to replay attacks in between increments of the sequence number. The sequence number can be incremented as seldom (or as often) as desired, trading off resistance to replay attacks with the computational effort required for authentication.

Meticulous Keyed MD5 authentication is stronger yet, as it requires the sequence number to be incremented for every packet. Replay attack vulnerability is reduced due to the requirement that the

sequence number must be incremented on every packet, the window size of acceptable packets is small, and the initial sequence number is randomized. There is still a window of attack at the beginning of the session while the sequence number is being determined. This authentication scheme requires an MD5 calculation on every packet transmitted and received.

Using SHA1 is believed to have stronger security properties than MD5. All comments about MD5 in this section also apply to SHA1.

Both Keyed MD5/SHA1 and Meticulous Keyed MD5/SHA1 use the "secret suffix" construction (also called "append only") in which the shared secret key is appended to the data before calculating the hash, instead of the more common Hashed Message Authentication Code (HMAC) construction [HMAC]. This construction is believed to be appropriate for BFD, but designers of any additional authentication mechanisms for BFD are encouraged to read [HMAC] and its references.

If both systems randomize their Local Discriminator values at the beginning of a session, replay attacks may be further mitigated, regardless of the authentication type in use. Since the Local Discriminator may be changed at any time during a session, this mechanism may also help mitigate attacks.

The security implications of the use of BFD Echo packets are dependent on how those packets are defined, since their structure is local to the transmitting system and outside the scope of this specification. However, since Echo packets are defined and processed only by the transmitting system, the use of cryptographic authentication does not guarantee that the other system is actually alive; an attacker could loop the Echo packets back (without knowing any secret keys) and cause the link to be falsely declared to be up. This can be mitigated by using a suitable interval for BFD Control packets. [GTSM] could be applied to BFD Echo packets, though the TTL/Hop Count will be decremented by 1 in the course of echoing the packet, so spoofing is possible from one hop away.

10. References

10.1. Normative References

[GTSM] Gill, V., Heasley, J., Meyer, D., Savola, P., Ed., and C. Pignataro, "The Generalized TTL Security Mechanism (GTSM)", RFC 5082, October 2007.

[KEYWORDS] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[MD5] Rivest, R., "The MD5 Message-Digest Algorithm", RFC 1321, April 1992.

[SHA1] Eastlake 3rd, D. and P. Jones, "US Secure Hash Algorithm 1 (SHA1)", RFC 3174, September 2001.

10.2. Informative References

[HMAC] Krawczyk, H., Bellare, M., and R. Canetti, "HMAC: Keyed-Hashing for Message Authentication", RFC 2104, February 1997.

[IANA-CONSIDERATIONS] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008.

[OSPF] Moy, J., "OSPF Version 2", STD 54, RFC 2328, April 1998.

Appendix A. Backward Compatibility (Non-Normative)

Although version 0 of this protocol (as defined in early versions of the Internet-Draft that became this RFC) is unlikely to have been deployed widely, some implementors may wish to have a backward compatibility mechanism. Note that any mechanism may be potentially used that does not alter the protocol definition, so interoperability should not be an issue.

The suggested mechanism described here has the property that it will converge on version 1 if both systems implement it, even if one system is upgraded from version 0 within a Detection Time. It will interoperate with a system that implements only one version (or is configured to support only one version). A system should obviously not perform this function if it is configured to or is only capable of using a single version.

A BFD session will enter a "negotiation holddown" if it is configured for automatic versioning and either has just started up, or the session has been manually cleared. The session is set to AdminDown state and version 1. During the holddown period, which lasts for one Detection Time, the system sends BFD Control packets as usual, but ignores received packets. After the holddown time is complete, the state transitions to Down and normal operation resumes.

When a system is not in holddown, if it doing automatic versioning and is currently using version 1, if any version 0 packet is received for the session, it switches immediately to version 0. If it is currently using version 0 and a version 1 packet is received that indicates that the neighbor is in state AdminDown, it switches to version 1. If using version 0 and a version 1 packet is received indicating a state other than AdminDown, the packet is ignored (per spec).

If the version being used is changed, the session goes down as appropriate for the new version (Down state for version 1 or Failing state for version 0).

Appendix B. Contributors

Kireeti Kompella and Yakov Rekhter of Juniper Networks were also significant contributors to this document.

Appendix C. Acknowledgments

This document was inspired by (and is intended to replace) the Protocol Liveness Protocol document, written by Kireeti Kompella.

Demand mode was inspired by "A Traffic-Based Method of Detecting Dead Internet Key Exchange (IKE) Peers", by G. Huang, et al.

The authors would also like to thank Mike Shand, John Scudder, Stewart Bryant, Pekka Savola, Richard Spencer, and Pasi Eronen for their substantive input.

The authors would also like to thank Owen Wheeler for hosting teleconferences between the authors of this specification and multiple vendors in order address implementation and clarity issues.

Authors' Addresses

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

Phone: +1-408-745-2000
EMail: dkatz@juniper.net

Dave Ward
Juniper Networks
1194 N. Mathilda Ave.
Sunnyvale, CA 94089-1206
USA

Phone: +1-408-745-2000
EMail: dward@juniper.net