Network Working Group
Internet Draft

D. Katz
Juniper Networks
D. Ward
Cisco Systems

Expires: January, 2006

July, 2005

Bidirectional Forwarding Detection
draft-ietf-bfd-base-03.txt

Status of this Memo

Copyright Notice

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. Comments on this draft should be directed to rtg-bfd@ietf.org.

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].

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. Currently, detection can come fairly quickly in certain circumstances when data link hardware comes into play (such as 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 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 will be 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.

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 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 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 to not 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 which 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 each system has an independent way of verifying that it has connectivity to the other system, so once a BFD session is established, the systems stop sending BFD Control packets, except when either system feels the need to verify connectivity explicitly, in which case a short sequence of BFD Control packets is sent, and then the protocol quiesces.

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 in a row 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 modes. 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, which 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 also may not be used when the path round trip time is greater than the desired detection time. See section 6.5 for more details.

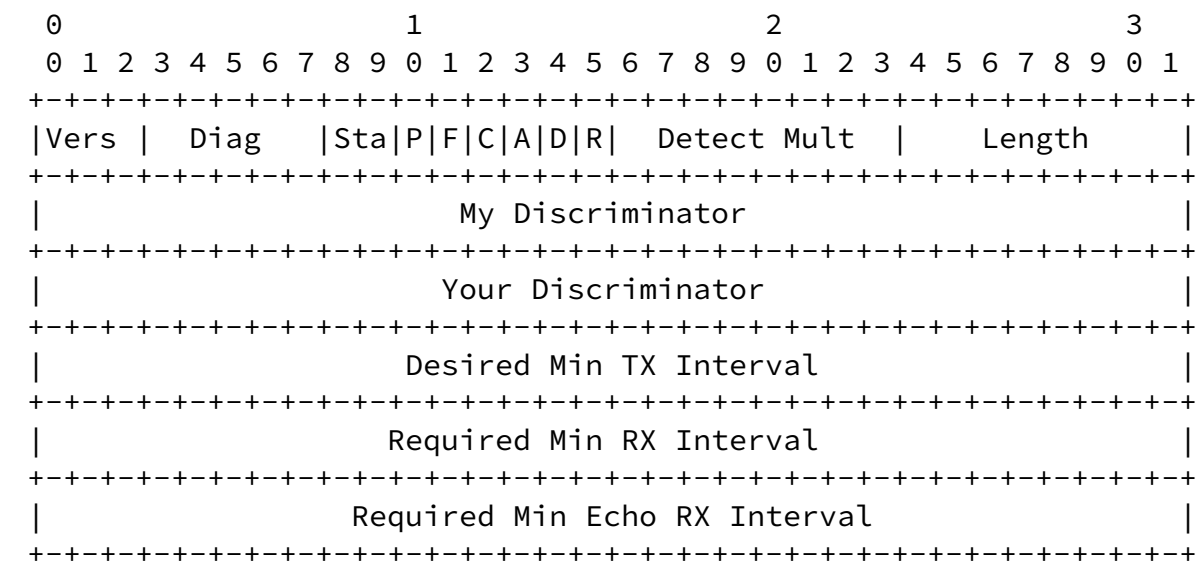
4. BFD Control Packet Format

4.1. Generic BFD Control Packet Format

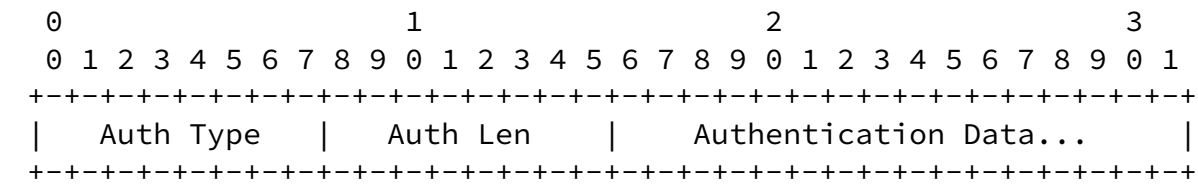
BFD Control packets are sent in an encapsulation appropriate to the environment, which is outside of the scope of this document. See the appropriate application document for encapsulation details.

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 session state change. 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. 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.

Demand (D)

If set, the transmitting system wishes to operate in Demand Mode. If clear, the transmitting system does not wish to or is not capable of operating in Demand Mode.

Reserved (R)

This bit must be zero on transmit, and ignored on receipt.

Detect Mult

Detect time multiplier. The negotiated transmit interval, multiplied by this value, provides the detection time for the transmitting 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.

Required Min RX Interval

This is the minimum interval, in microseconds, between received BFD Control packets that this system is capable of supporting.

Required Min Echo RX Interval

This is the minimum interval, in microseconds, between received BFD Echo packets that this system is capable of supporting. 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

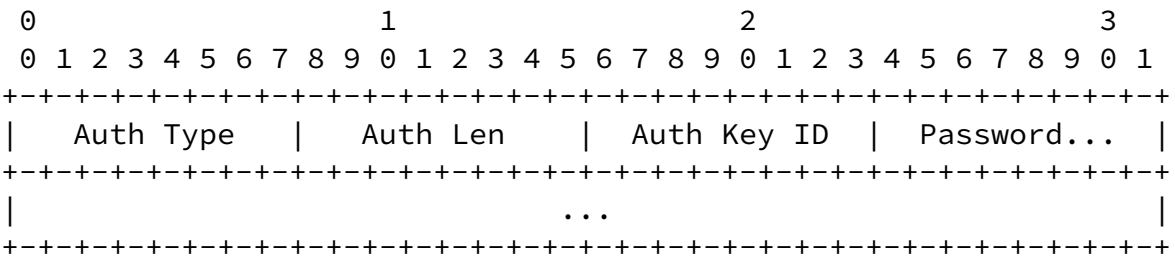
4 - Keyed SHA1
5 - Meticulous Keyed SHA1
6-255 - Reserved for future use

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 Autentication 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

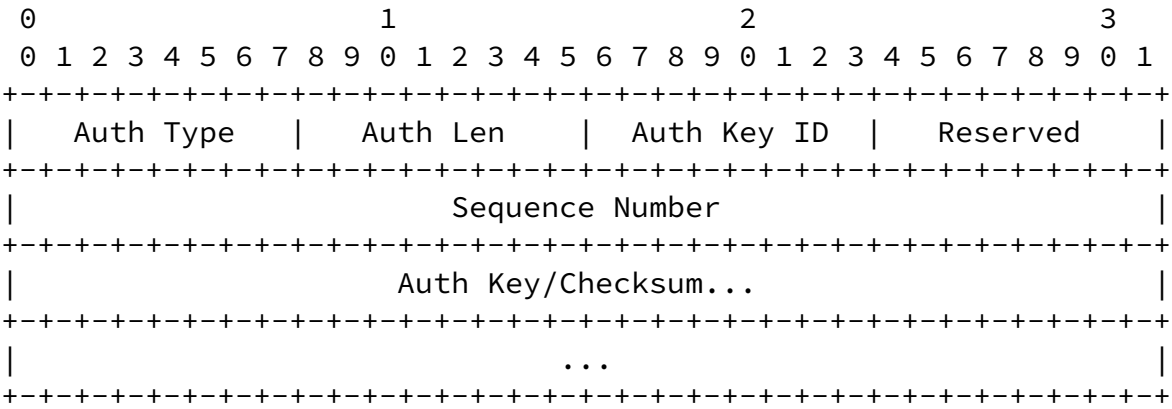
multiple keys to be active simultaneously.

Password

The simple password in use on this session. The password MUST be from 1 to 16 bytes in length.

4.3. Keyed MD5 and Meticulous Keyed MD5 Authentication Section Format

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

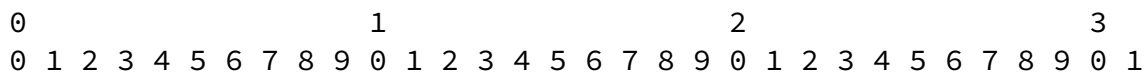
protection against replay attacks.

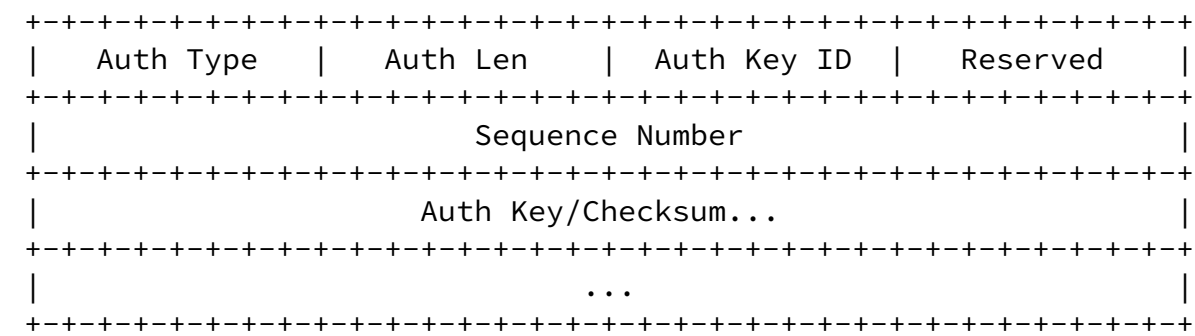
Auth Key/Checksum

This field carries the 16 byte MD5 checksum for the packet. When the checksum is calculated, the shared MD5 key is stored in this field. (See section 6.6.3 for details.)

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/Checksum

This field carries the 20 byte SHA1 checksum for the packet. When the checksum is calculated, the shared SHA1 key is stored in this field. (See section 6.6.4 for details.)

5. BFD Echo Packet Format

BFD Echo packets are sent in an encapsulation appropriate to the environment. See the appropriate application document 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.

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 adversely affect interoperability.

Remember that all references of the form "bfd.Xx" refer to internal state variables (defined in section 6.7.1), 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 comes up.

Once the BFD session is Up, a system can choose to start the Echo function if it desires to 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.

If both systems signal that they want to use Demand mode, the transmission of BFD Control packets ceases once the session is Up. Other means of implying connectivity are used to keep the session alive. If one of the systems wishes to verify connectivity, it can initiate a short exchange (a "Poll Sequence") of BFD Control packets to verify this.

If Demand mode is not active, and no Control packets are received in

the calculated detection time (see section 6.7.4), the session is declared down, and signalled 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.

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.

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.

Init state means that the remote system is communicating, and the

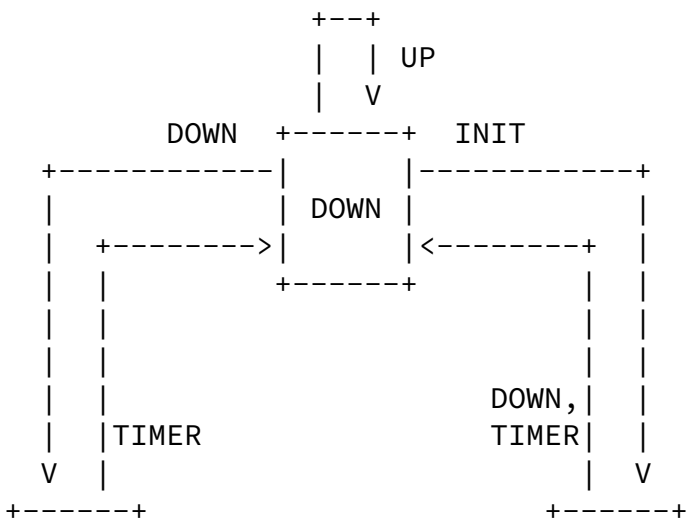
local system desires to bring the session up, but the remote system

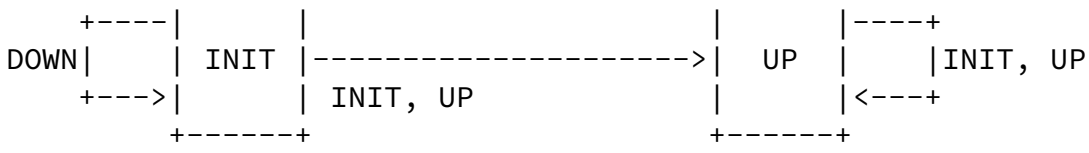
does not yet realize it. A session will remain in Init state until either a BFD Control Packet is received that is signalling Init or Up state (in which case the session advances to Up state) or until the detection time expires, meaning that communication with the remote system has been lost (in which 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.

The following diagram provides an overview of the state machine. Transitions involving AdminDown state are deleted for clarity (but are fully specified in section 6.7.6.) 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 Time.





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 signalled to the system looping them back.

When a system is using the Echo function, it is advantageous to choose a sedate transmission rate for Control packets, since liveness detection is being handled by the Echo packets. This can be controlled by manipulating the Desired Min TX Interval field (see section 6.7.3.)

If the Echo function is only being run in one direction, the system not running the Echo function will more likely wish to send 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 always 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. Demand Mode

Demand mode is negotiated by virtue of both systems setting the Demand (D) bit in its BFD Control packets. Both systems must request Demand mode for it to become active.

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.

Once a BFD session comes up, if Demand mode is active, both systems stop sending periodic BFD Control packets, and depend on the

alternative mechanism for maintaining ongoing connectivity.

When a system wishes to verify connectivity, it initiates a Poll Sequence. It starts periodically sending BFD Control packets with the Poll (P) bit set, at the negotiated transmission rate. When a system receives such a packet, it immediately replies with a BFD Control packet of its own, with the Poll (P) bit clear, and the Final (F) bit set. The receipt of a reply to a Poll terminates the Poll Sequence. 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.

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 6.7.8 for more details.

Note that this mechanism requires that the detection time negotiated is greater than the round trip time between the two systems, or the Poll mechanism will always fail. Enforcement of this requirement is outside the scope of this specification.

Demand mode MAY be enabled or disabled at any time by setting or clearing the Demand (D) bit in the BFD Control packet, without affecting the BFD session state.

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.

6.6. 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.

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 6.7.6.

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

6.6.1. 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 detect 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 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.)

6.6.2. 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.)

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.

6.6.3. 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] checksum 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 checksum 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 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 current authentication key value MUST be placed into the Auth Key/Checksum field. An MD5 checksum MUST be calculated over the entire BFD control packet. The resulting checksum MUST be stored in the Auth Key/Checksum 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 entitled "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.

Replace the contents of the Auth Key/Checksum field with the authentication key selected by the received Auth Key ID field. If the MD5 checksum of the entire BFD Control packet is not equal to the received value of the Auth Key/Checksum field, the received 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, bfd.RcvAuthSeq MUST be set to the value of the received Sequence Number field, and the received packet MUST be accepted.

6.6.4. 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] checksum 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 checksum 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 current authentication key value MUST be placed into the Auth Key/Checksum field. A SHA1 checksum MUST be calculated over the entire BFD control packet. The resulting checksum MUST be stored in the Auth Key/Checksum 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 entitled "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.

Replace the contents of the Auth Key/Checksum field with the

authentication key selected by the received Auth Key ID field. If the SHA1 checksum of the entire BFD Control packet is not equal to the received value of the Auth Key/Checksum field, the received 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.

6.7. 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 signalled its willingness to loop back Echo packets.

When a system is said to have "Demand mode active," it means that bfd.DemandModeDesired is 1 in the local system (see State Variables below), the remote system is signalling with the Demand (D) bit set, and that the session is Up.

6.7.1. 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.

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.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.

bfd.LocalDiag

The diagnostic code specifying the reason for the most recent local session state chage. 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. 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 6.7.3. 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. The setting of this variable is outside the scope of this specification.

bfd.DemandModeDesired

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

bfd.DetectMult

The desired detect time multiplier for BFD Control packets.

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 6.7.4 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 next sequence number for keyed MD5 or SHA1 authentication expected to be 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.

6.7.2. 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. Packets are always periodically transmitted in Asynchronous mode, and are periodically transmitted during Poll Sequences when in Demand mode.

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. With the exceptions listed in the remainder of this section, a system MUST NOT transmit BFD Control packets with an interval less than the larger of bfd.DesiredMinTxInterval and the received Required Min RX Interval field. In other words, the system reporting the slower rate determines the transmission rate.

The periodic transmission of BFD Control packets SHOULD be jittered by up to 25%, that is, the interval SHOULD be reduced by a random value of 0 to 25%, in order to avoid self-synchronization. Thus, the average interval between packets may be up to 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 DetectTime does not pass prior to the receipt of the next BFD Control packet.

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.

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.

6.7.3. 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 Demand mode is active, and either bfd.DesiredMinTxInterval is changed or bfd.RequiredMinRxInterval is changed, a Poll Sequence MUST be initiated (see section 6.7.8).

If Demand mode is not active, bfd.SessionState is Up, and either bfd.DesiredMinTxInterval is changed or bfd.RequiredMinRxInterval is changed, all subsequent transmitted Control packets MUST be sent with the Poll (P) bit set until a packet is received with the Final (F) bit set (except for those packets sent in response to received Polls.)

If bfd.DesiredMinTxInterval is increased and bfd.SessionState is Up, the actual transmission interval used MUST NOT change until a Control packet is received with the Final (F) bit set. This is to ensure that the remote system updates its Detect Time before the transmission interval increases.

If bfd.RequiredMinRxInterval is reduced and bfd.SessionState is Up, the calculated detection time for the remote system MUST NOT change until a Control packet is received with the Final (F) bit set. 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.

When the Echo function is active, a system SHOULD set bfd.DesiredMinTxInterval to a value of not less than one second (1,000,000 microseconds.) This is intended to keep BFD Control

traffic at a negligible level, since the actual detection function is being performed using BFD Echo packets.

6.7.4. 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, in Asynchronous mode, 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 the last received Required Min RX Interval.) 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 6.7.6.)

6.7.5. 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 which will detect a communication failure is acceptable.

6.7.6. 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 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 6.6, 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.

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

If Demand mode is active, 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.

If Demand mode is not active, the Final (F) bit in the received packet is set, and the local system has been transmitting packets with the Poll (P) bit set, the Poll (P) bit MUST be set to zero in subsequent transmitted packets.

Update the Detection Time as described in section 6.7.4.

Update the transmit interval as described in section 6.7.2.

```
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
```

If the Demand (D) bit is set and bfd.DemandModeDesired is 1, and bfd.SessionState is Up, Demand mode is active.

If the Demand (D) bit is clear or bfd.DemandModeDesired is 0, or bfd.SessionState is not Up, Demand mode is not active.

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.

If the packet was not discarded, it has been received for purposes of the Detection Time expiration rules in section 6.7.4.

6.7.7. Transmitting BFD Control Packets

BFD Control packets MUST be transmitted periodically at the rate determined according to section 6.7.2, except as specified in this section.

The transmit interval MUST be recalculated whenever bfd.DesiredMinTxInterval changes, or whenever the received Required Min RX Interval changes, and is equal to the greater of those two values. See sections 6.7.2 and 6.7.3 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 Demand mode is active and a Poll Sequence is not being transmitted.

A system MUST send a BFD Control packet in response to a received BFD Control Packet with the Poll (P) bit set, regardless of the BFD session state. The packet sent in response MUST NOT have the Poll (P) bit set, and MUST have the Final (F) bit set. 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 BFD Control packet SHOULD be transmitted between normally scheduled transmissions when the contents of that packet would differ from those 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 is required to do so according to the requirements of section 6.7.3, 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.DemandModeDesired.

Reserved (R)

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 6.6 if authentication is in use (bfd.AuthType is nonzero.) Otherwise this section is not present.

6.7.8. Initiation of a Poll Sequence

If Demand mode is active, 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.

If Demand mode is active, a Poll Sequence SHOULD be initiated whenever the system feels the need to verify connectivity with the remote system. The conditions under which this is desirable are outside the scope of this specification.

If a Poll Sequence is being sent, and a new Poll Sequence is initiated due to one of the above conditions, the detection interval MUST be restarted in order to ensure that a full Poll Sequence is transmitted under the new conditions.

6.7.9. 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.

6.7.10. 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.

6.7.11. 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 sections 6.7.3 and 6.7.8 for further requirements.

6.7.12. 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 sections 6.7.3 and 6.7.8 apply.

6.7.13. 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. See section 6.7.8 for additional requirements.

6.7.14. 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 6.7.10.)

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 RX Interval to zero or nonzero in outgoing BFD Control packets.

6.7.15. 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.DemandModeDesired to the proper value. If Demand mode is no longer active, the system MUST begin transmitting periodic BFD Control packets as described in section 6.7.7.

6.7.16. 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.

6.7.17. 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 signalling is received from outside BFD that the underlying path has failed, an implementation MAY adminstratively disable the session with the diagnostic Path Down.

Other scenarios MAY use the diagnostic Administratively Down.

6.7.18. Concatenated Paths

If the path being monitored by BFD is concatenated with other paths, 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, 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, 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.

Backward Compatibility (Non-Normative)

Although Version 0 of this document 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.)

Contributors

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

Acknowledgments

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

Demand Mode was inspired by draft-ietf-ipsec-dpd-03.txt, by G. Huang et al.

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

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. This ultimately has denial-of-service effects, as links may be declared to be down (or falsely declared to be up.)

When BFD is run over network layer protocols, a significant denial-of-service risk is created, as BFD packets may be trivial to spoof. When the 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 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 rather than MD5 is believed to have stronger security properties. All comments about MD5 in this section also apply to SHA1.

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.

IANA Considerations

This document has no actions for IANA.

Normative References

[GTSM] Gill, V., et al, "The Generalized TTL Security Mechanism (GTSM)", RFC 3682, February 2004.

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

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

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

[SHA1] "Secure Hash Standard", United States of America, National Institute of Science and Technology, Federal Information Processing Standard (FIPS) 180-1, April 1993.

Authors' Addresses

Dave Katz
Juniper Networks
1194 N. Mathilda Ave.
Sunnyvale, California 94089-1206 USA
Phone: +1-408-745-2000
Email: dkatz@juniper.net

Dave Ward
Cisco Systems
170 W. Tasman Dr.
San Jose, CA 95134 USA
Phone: +1-408-526-4000
Email: dward@cisco.com

Changes from the previous draft

The primary technical change was the addition of a suggested (non-normative) backward compatibility mechanism with version 0 of BFD. A minor tweak to the state machine to more explicitly spell out the action to be taken in AdminDown state was done.

Otherwise, the changes in this draft from the previous version are cosmetic and/or editorial.

IPR Notice

The IETF has been notified of intellectual property rights claimed in regard to some or all of the specification contained in this document. For more information consult the online list of claimed rights.

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in BCP 78 and BCP 79.

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at http://www.ietf.org/ipr.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.

Full Copyright Notice

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.

This document expires in January, 2006.