Network Working Group
Internet Draft

D. Katz
Juniper Networks
D. Ward
Cisco Systems

Expires: September, 2005

March, 2005

BFD for IPv4 and IPv6 (Single Hop)
draft-ietf-bfd-v4v6-1hop-02.txt

Status of this Memo

Copyright Notice

Abstract

This document describes the use of the Bidirectional Forwarding Detection protocol over IPv4 and IPv6 for single IP hops. It further describes the use of BFD with OSPFv2, OSPFv3, and IS-IS. 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].

1. Introduction

One very desirable application for BFD [BFD] is to track IPv4 and IPv6 connectivity between directly-connected systems. This could be used to supplant the detection mechanisms in IS-IS and OSPF, or to monitor router-host connectivity, among other applications.

This document describes the particulars necessary to use BFD in this environment, and describes how BFD can be used in conjunction OSPFv2 [OSPFv2], OSPFv3 [OSPFv3], and IS-IS [ISIS].

2. Applications and Limitations

This application of BFD can be used by any pair of systems communicating via IPv4 and/or IPv6 across a single IP hop that can be associated with an incoming interface. This includes, but is not limited to, physical media, virtual circuits, and tunnels.

Each BFD session between a pair of systems MUST traverse a separate path in both directions.

If BFD is to be used in conjunction with both IPv4 and IPv6 on a particular link, a separate BFD session MUST be established for each protocol (and thus encapsulated by that protocol) over that link.

3. Initialization and Demultiplexing

In this application, there will be only a single BFD session between two systems over a given interface (logical or physical) for a particular protocol. The BFD session must be bound to this interface. As such, both sides of a session MUST take the "Active" role (sending initial BFD Control packets with a zero value of Your Discriminator) and any BFD packet from the remote machine with a zero value of Your Discriminator MUST be associated with the session bound to the remote system, interface, and protocol.

4. Encapsulation

4.1. BFD for IPv4

In the case of IPv4, BFD Control packets MUST be transmitted in UDP packets with destination port 3784, within an IPv4 packet. The source port MUST be in the range 49152 through 65535. The same UDP source port number MUST be used for all BFD Control packets associated with a particular session. The source port number SHOULD be unique among all BFD sessions on the system. If more than 16384 BFD sessions are simultaneously active, UDP source port numbers MAY be reused on multiple sessions, but the number of distinct uses of the same UDP source port number SHOULD be minimized. An implementation MAY use the UDP port source number to aid in demultiplexing incoming BFD Control packets, but ultimately the mechanisms in [BFD] MUST be used to demultiplex incoming packets to the proper session.

BFD Echo packets MUST be transmitted in UDP packets with destination UDP port 3785 in an IPv4 packet. The setting of the UDP source port is outside the scope of this specification. The destination address MUST be chosen in such a way as to cause the remote system to forward the packet back to the local system. The source address MUST be chosen in such a way as to preclude the remote system from generating ICMP Redirect messages (in particular, the source address MUST NOT be part of the subnet bound to the interface over which the BFD Echo packet is being transmitted.)

4.2. BFD for IPv6

In the case of IPv6, BFD Control packets MUST be transmitted in UDP packets with destination port 3784, within an IPv6 packet. The source port MUST be in the range 49152 through 65535. The same UDP source port number MUST be used for all BFD Control packets

associated with a particular session. The source port number SHOULD be unique among all BFD sessions on the system. If more than 16384 BFD sessions are simultaneously active, UDP source port numbers MAY be reused on multiple sessions, but the number of distinct uses of the same UDP source port number SHOULD be minimized. An implementation MAY use the UDP port source number to aid in demultiplexing incoming BFD Control packets, but ultimately the mechanisms in [BFD] MUST be used to demultiplex incoming packets to the proper session.

BFD Echo packets MUST be transmitted in UDP packets with destination UDP port 3785 in an IPv6 packet. The setting of the UDP source port is outside the scope of this specification. The source and destination addresses MUST both be associated with the local system. The destination address MUST be chosen in such a way as to cause the remote system to forward the packet back to the local system.

5. TTL/Hop Count Issues

If BFD authentication is not in use on a session, all BFD Control packets for the session MUST be sent with a TTL or Hop Count value of 255. All received BFD Control packets that are demultiplexed to the session MUST be discarded if the received TTL or Hop Count is not equal to 255. A discussion of this mechanism can be found in [GTSM].

If BFD authentication is in use on a session, all BFD Control packets MUST be sent with a TTL or Hop Count value of 255. All received BFD Control packets that are demultiplexed the session MAY be discarded if the received TTL or Hop Count is not equal to 255.

In the context of this section, "authentication in use" means that the system is sending BFD control packets with the Authentication bit set and with the Authentication Section included, and that all unauthenticated packets demultiplexed to the session are discarded, per the BFD base specification.

6. Addressing Issues

On a subnetted network, BFD Control packets MUST be transmitted with source and destination addresses that are part of the subnet (addressed from and to interfaces on the subnet.)

On an addressed but unsubnetted point-to-point link, BFD Control packets MUST be transmitted with source and destination addresses that match the addresses configured on that link.

On an unnumbered point-to-point link, the source address of a BFD Control packet MUST NOT be used to identify the session. This means that the initial BFD packet MUST be accepted with any source address, and that subsequent BFD packets MUST be demultiplexed solely by the My Discriminator field (as is always the case.) This allows the source address to change if necessary. If the received source address changes, the local system MUST NOT use that address as the destination in outgoing BFD Control packets; rather it MUST continue to use the address configured at session creation. An implementation MAY notify the application that the neighbor's source address has changed, so that the application might choose to change the destination address or take some other action. Note that the TTL/Hop Count check described in section 5 (or the use of authentication) precludes the BFD packets from having come from any source other than the immediate neighbor.

7. BFD for use with OSPFv2, OSPFv3, and IS-IS

The two versions of OSPF, as well as IS-IS, all suffer from an architectural limitation, namely that their Hello protocols are limited in the granularity of failure their detection times. In particular, OSPF has a minimum detection time of two seconds, and IS-IS has a minimum detection time of one second.

BFD MAY be used to achieve arbitrarily small detection times for these protocols by supplanting the Hello protocols used in each case.

It should be noted that the purpose of using BFD in this context is not to replace the adjacency timeout mechanism, nor is it to demonstrate that the network is fully functional for the use of the routing protocol, but is simply to advise the routing protocol that there are problems forwarding the data protocol for which the routing protocol is calculating routes.

7.1. Session Establishment

The mechanism by which a BFD session is established in this environment is outside the scope of this specification. An obvious choice would be to use the discovery mechanism inherent in the Hello protocols in OSPF and IS-IS to bootstrap the establishment of a BFD session.

Any BFD sessions established to support OSPF and IS-IS across a single IP hop MUST operate in accordance with the rest of this document.

If multiple routing protocols wish to establish BFD sessions with the same remote system for the same data protocol, all MUST share a single BFD session.

7.2. Session Parameters

The setting of the various timing parameters and modes in this application are outside the scope of this specification.

Note that all protocols sharing a session will operate using the same parameters. The mechanism for choosing the parameters among those desired by the various protocols are outside the scope of this specification.

7.3. Interactions with OSPF and IS-IS without Graceful Restart

Slightly different mechanisms are used if the routing protocol supports the routing of multiple data protocols, depending on whether the routing protocol supports separate topologies for each data protocol. With a shared topology, if one of the data protocols fails (as signalled by the associated BFD session), it is necessary to consider the path to have failed for all data protocols, since there is otherwise no way for the routing protocol to turn away traffic for the failed protocol (and such traffic would be black holed indefinitely.)

With individual routing topologies for each data protocol, only the failed data protocol needs to be rerouted around the failed path.

Therefore, when a BFD session transitions from Up to Down, action SHOULD be taken in the routing protocol to signal the lack of connectivity for the data protocol (IPv4 or IPv6) over which BFD is running. If only one data protocol is being advertised in the routing protocol Hello, or if multiple protocols are being advertised

but the protocols must share a common topology, a Hello protocol timeout SHOULD be emulated for the associated OSPF neighbors and/or IS-IS adjacencies.

If multiple data protocols are advertised in the routing protocol Hello, and the routing protocol supports different topologies for each data protocol, the failing data protocol SHOULD no longer be advertised in Hello packets in order to signal a lack of connectivity for that protocol.

Note that it is possible in some failure scenarios for the network to be in a state such that the IGP comes up, but the BFD session cannot be established, and, more particularly, data cannot be forwarded. To avoid this situation, it would be beneficial to not allow the IGP to establish a neighbor/adjacency. However, this would preclude the operation of the IGP in an environment in which not all systems support BFD.

Therefore, if a BFD session is not in Up state (possibly because the remote system does not support BFD), it is OPTIONAL to preclude the establishment of an OSPF neighbor or an IS-IS adjacency. The choice of whether to do so SHOULD be controlled by means outside the scope of this specification, such as configuration or other mechanisms.

7.4. Interactions with OSPF and IS-IS with Graceful Restart

The Graceful Restart functions in OSPF [OSPF-GRACE] and IS-IS [ISIS-GRACE] are predicated on the existence of a separate forwarding plane that does not necessarily share fate with the control plane in which the routing protocols operate. In particular, the assumption is that the forwarding plane can continue to function while the protocols restart and sort things out.

BFD implementations announce via the Control Plane Independent (C) bit whether or not BFD shares fate with the control plane. This information is used to determine the actions to be taken in conjunction with Graceful Restart.

If BFD does not share its fate with the control plane on either system, it can be used to determine whether Graceful Restart is NOT viable (the forwarding plane is not operating.) In this situation, if a BFD session fails while graceful restart is taking place, and BFD is independent of the control plane on the local system, and the remote system has been transmitting BFD Control packets with the C bit set, the graceful restart SHOULD be aborted and the topology change made visible to the network as outlined in section 7.3.

If BFD shares its fate with the control plane on either system (either the local system shares fate with the control plane, or the remote system is transmitting BFD packets with the C bit set to zero), it is not useful during graceful restart, as the BFD session is likely to fail regardless of the state of the forwarding plane. The action to take in this case depends on the capabilities of the IGP.

7.4.1. OSPF Graceful Restart With Control Plane Fate Sharing

OSPF has a "planned" restart mechanism, in which the restarting system notifies its neighbors that it is about to perform a restart. In this situation, if a BFD session fails while the neighbor is performing a graceful restart, the graceful restart SHOULD be allowed to complete and the topology change should not be made visible to the network as outlined in section 7.3.

For unplanned restarts (in which the neighbor has not notified the local system of its intention to restart), the OSPF Graceful Restart specification allows a Graceful Restart to take place if the system restarts prior to the expiration of the OSPF neighbor relationship. In this case, the BFD Detection Time is likely to expire prior to the restart, and the neighbor relationship SHOULD be torn down. In the unlikely event that the system restarts quickly enough, and the system chooses to attempt a Graceful Restart, the graceful restart SHOULD be allowed to complete and the topology change should not be made visible to the network as outlined in section 7.3.

7.4.2. ISIS Graceful Restart With Control Plane Fate Sharing

ISIS Graceful Restart does not signal a "planned" restart; its mechanism does not begin until after the system has restarted. If the BFD session expires prior to the restart of the system, there is no way for the neighbors to know that a Graceful Restart will take place.

If a planned restart is about to place, the restarting system MAY change the BFD timing parameters on a temporary basis in such a way as to make the Detection Time greater than or equal to the ISIS adjacency timeout. This will provide the restarting system the same opportunity to enter Graceful Restart as it would have without BFD. In this case, the restarted system SHOULD avoid sending any BFD Control packets until there is a high likelihood that its neighbors know it is performing a Graceful Restart, since the neighbors will tear down their BFD sessions when those sessions restart.

In any case, if a BFD session fails while the neighbor is known to be performing a Graceful Restart, the Graceful Restart SHOULD be allowed to complete and the topology change should not be made visible to the network as outlined in section 7.3.

If the BFD session fails, and it is not known whether the neighbor is performing a Graceful Restart, the BFD session failure SHOULD be made visible to the network as outlined in section 7.3.

7.5. OSPF Virtual Links

If it is desired to use BFD for failure detction of OSPF Virtual Links, the mechanism described in [BFD-MULTI] MUST be used, since OSPF Virtual Links may traverse an arbitrary number of hops. BFD Authentication SHOULD be used and is strongly encouraged.

8. BFD for use with Tunnels

A number of mechanisms are available to tunnel IPv4 and IPv6 over arbitrary topologies. If the tunnel mechanism does not decrement the TTL or hop count of the network protocol carried within, the mechanism described in this document may be used to provide liveness detection for the tunnel. The BFD Authentication mechanism SHOULD be used and is strongly encouraged.

Normative References

[BFD] Katz, D., and Ward, D., "Bidirectional Forwarding Detection", draft-ietf-bfd-base-02.txt, March, 2005.

[BFD-MULTI] Katz, D., and Ward, D., "BFD for Multihop Paths", draft-ietf-bfd-multihop-02.txt, March, 2005.

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

[ISIS] Callon, R., "Use of OSI IS-IS for routing in TCP/IP and dual environments", RFC 1195, December 1990.

[ISIS-GRACE] Shand, M., and Ginsberg, L., "Restart signaling for IS-IS", RFC 3847, July 2004.

[KEYWORD] Bradner, S., "Key words for use in RFCs to Indicate

Requirement Levels", RFC 2119, March 1997.

[OSPFv2] Moy, J., "OSPF Version 2", RFC 2328, April 1998.

[OSPFv3] Coltun, R., et al, "OSPF for IPv6", RFC 2740, December 1999.

[OSPF-GRACE] Moy, J., et al, "Graceful OSPF Restart", RFC 3623, November 2003.

Security Considerations

In this application, the use of TTL=255 on transmit and receive is viewed as supplying equivalent security characteristics to other protocols used in the infrastructure, as it is not trivially spoofable. The security implications of this mechanism are further discussed in [GTSM].

The security implications of the use of BFD Authentication are discussed in [BFD].

The use of the TTL=255 check simultaneously with BFD Authentication provides a low overhead mechanism for discarding a class of unauthorized packets and may be useful in implementations in which cryptographic checksum use is susceptable to denial of service attacks. The use or non-use of this mechanism does not impact interoperability.

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 only substantive changes are to allow the TTL=255 check to be optional when authentication is in use, and to specify that the destination address in BFD Control packets does not change even if the source address of the neighbor's packets changes (which is allowed on unnumbered links.)

Explicatory language on the issues of interaction with multiprotocol routing protocols was added.

All other changes are editorial.

Full Copyright Notice

Acknowledgement

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

This document expires in September, 2005.