Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: May 4, 2020

J. Haas
Juniper Networks, Inc.
A. Fu
Bloomberg L.P.
November 1, 2019

BFD Encapsulated in Large Packets
draft-ietf-bfd-large-packets-02

Abstract

The Bidirectional Forwarding Detection (BFD) protocol is commonly used to verify connectivity between two systems. BFD packets are typically very small. It is desirable in some circumstances to know that not only is the path between two systems reachable, but also that it is capable of carrying a payload of a particular size. This document discusses thoughts on how to implement such a mechanism using BFD in Asynchronous mode.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" are to be interpreted as described in [RFC2119] only when they appear in all upper case. They may also appear in lower or mixed case as English words, without normative meaning.

Status of This Memo

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

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

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

This Internet-Draft will expire on May 4, 2020.

Table of Contents

1. Introduction

The Bidirectional Forwarding Detection (BFD) [RFC5880] protocol is commonly used to verify connectivity between two systems. However, some applications may require that the Path MTU [RFC1191] between those two systems meets a certain minimum criteria. When the Path MTU decreases below the minimum threshold, those applications may wish to consider the path unusable.

BFD may be encapsulated in a number of transport protocols. An example of this is single-hop BFD [RFC5881]. In that case, the link MTU configuration is typically enough to guarantee communication between the two systems for that size MTU. BFD Echo mode

(Section 6.4 of [RFC5880]) is sufficient to permit verification of the Path MTU of such directly connected systems. Previous proposals ([I-D.haas-xiao-bfd-echo-path-mtu]) have been made for testing Path MTU for such directly connected systems. However, in the case of multi-hop BFD [RFC5883], this guarantee does not hold.

The encapsulation of BFD in multi-hop sessions is a simple UDP packet. The BFD elements of procedure (Section 6.8.6 of [RFC5880]) covers validating the BFD payload. However, the specification is silent on the length of the encapsulation that is carrying the BFD PDU. While it is most common that the transport protocol payload (i.e. UDP) length is the exact size of the BFD PDU, this is not required by the elements of procedure. This leads to the possibility that the transport protocol length may be larger than the contained BFD PDU.

2. BFD Encapsulated in Large Packets

Support for BFD between two systems is typically configured, even if the actual session may be dynamically created by a client protocol. A new BFD variable is defined in this document:

bfd.PaddedPduSize
: The BFD transport protocol payload size is increased to this value. The contents of this additional payload MUST be zero. The minimum size of this variable MUST NOT be smaller than permitted by the element of BFD procedure; 24 or 26 - see Section 6.8.6 of [RFC5880].

The Don't Fragment bit (Section 2.3 of [RFC0791]) of the IP payload, when using IPv4 encapsulation, MUST be set.

3. Implementation and Deployment Considerations

3.1. Implementations that do not support Large BFD Packets

While this document proposes no change to the BFD protocol, implementations may not permit arbitrarily padded transport PDUs to carry BFD packets. While Section 6 of [RFC5880] warns against excessive pedantry, implementations may not work with this mechanism without additional support.

[RFC5880], section 6.8.6, discusses the procedures for receiving BFD Control packets. When an implementation is incapable of processing Large BFD Packets, it could manifest in one of two possible ways:

- A receiving BFD implementation is incapable of accepting Large BFD Packets. This is identical to the packet being discarded.

- A receiving BFD implementation is capable of accepting Large BFD Packets, but the Control packet is improperly rejected during validation procedures. This is identical to the packet being discarded.

In each of these cases, the BFD state machine would behave as if it were not receiving Control packets and the implementation would follow normal BFD procedures with regards to not having received Control packets.

3.2. Selecting MTU size to be detected

Since the consideration is path MTU, BFD sessions using this feature only need to use a bfd.PaddedPduSize appropriate to exercise the path MTU for the desired application. This may be significantly smaller than the system's link MTU; e.g. desired path MTU is 1500 bytes while the interface MTU that BFD with large packets is running on is 9000 bytes.

In the case multiple BFD clients desire to test the same BFD endpoints using different bfd.PaddedPduSize parameters, implementations should select the largest bfd.PaddedPduSize parameter from the configured sessions. This is similar to how implementations of BFD select the most aggressive timing parameters for multiple sessions to the same endpoint.

3.3. Detecting MTU mismatches

The accepted MTU for an interface is impacted by packet encapsulation considerations at a given layer; e.g. layer 2, layer 3, tunnel, etc. A common misconfiguration of interface parameters is inconsistent MTU. In the presence of inconsistent MTU, it is possible for applications to have unidirectional connectivity.

When it is necessary for an application using BFD with Large Packets to test the bi-directional Path MTU, it is necessary to configure the bfd.PaddedPduSize parameter on both sides of an interface. E.g., if the desire is to verify a 1500 byte MTU in both directions on an Ethernet or point to point link, each side of the BFD session must have bfd.PaddedPduSize set to 1500. In the absence of such consistent configuration, BFD with Large Packets may correctly determine unidirectional connectivity at the tested MTU, but bi-directional MTU may not be properly validated.

It should be noted that some interfaces may intentionally have different MTUs. Setting the bfd.PaddedPduSize appropriately for each side of the interface supports such scenarios.

3.4. Equal Cost Multiple Paths (ECMP) or other Load Balancing Considerations

Various mechanisms are utilized to increase throughput between two endpoints at various network layers. Such features include Link Aggregate Groups (LAGs) or ECMP forwarding. Such mechanisms balance traffic across multiple physical links while hiding the details of that balacing from the higher networking layers. The details of that balancing are highly implementation specific.

In the presence of such load balancing mechanisms, it is possible to have member links that are not properly forwarding traffic. In such circumstances, this will result in dropped traffic when traffic is chosen to be load balanced across those member links.

Such load balancing mechanisms may not permit all link members to be properly tested by BFD. This is because the BFD Control packets may be forwarded only along links that are up. BFD on LAG, [RFC7130], was developed to help cover one such scenario. However, for testing forwarding over multiple hops, there is no such specified general purpose BFD mechanism for exercising all links in an ECMP. This may result in a BFD session being in the Up state while some traffic may be dropped or otherwise negatively impacted along some component links.

Some BFD implementations utilize their internal understanding of the component links and their resultant forwarding to exercise BFD in such a way to better test the ECMP members and to tie the BFD session state to the health of that ECMP. Due to the implementation specific load balancing, it is not possible to standardize such additional mechanisms for BFD.

Mis-configuration of some member MTUs may lead to Load Balancing that may have an inconsistent Path MTU depending on how the traffic is balanced. While the intent of BFD with Large Packets is to verify path MTU, it is subject to the same considerations above.

3.5. S-BFD

This mechanism also can be applied to other forms of BFD, including S-BFD [RFC7880].

4. Security Considerations

This document does not change the underlying security considerations of the BFD protocol or its encapsulations.

5. IANA Considerations

This document introduces no additional considerations to IANA.

6. Acknowledgments

The authors would like to thank Les Ginsberg, Mahesh Jethandani, Robert Raszuk, and Ketan Talaulikar, for their valuable feedback on this proposal.

7. References

7.1. Normative References

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

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC5880] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD)", RFC 5880, DOI 10.17487/RFC5880, June 2010, <https://www.rfc-editor.org/info/rfc5880>.

[RFC5881] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD) for IPv4 and IPv6 (Single Hop)", RFC 5881, DOI 10.17487/RFC5881, June 2010, <https://www.rfc-editor.org/info/rfc5881>.

[RFC5883] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD) for Multihop Paths", RFC 5883, DOI 10.17487/RFC5883, June 2010, <https://www.rfc-editor.org/info/rfc5883>.

[RFC7130] Bhatia, M., Ed., Chen, M., Ed., Boutros, S., Ed., Binderberger, M., Ed., and J. Haas, Ed., "Bidirectional Forwarding Detection (BFD) on Link Aggregation Group (LAG) Interfaces", RFC 7130, DOI 10.17487/RFC7130, February 2014, <https://www.rfc-editor.org/info/rfc7130>.

[RFC7880] Pignataro, C., Ward, D., Akiya, N., Bhatia, M., and S. Pallagatti, "Seamless Bidirectional Forwarding Detection (S-BFD)", RFC 7880, DOI 10.17487/RFC7880, July 2016, <https://www.rfc-editor.org/info/rfc7880>.

7.2. Informative References

[I-D.haas-xiao-bfd-echo-path-mtu]
Haas, J. and M. Xiao, "Application of the BFD Echo function for Path MTU Verification or Detection", draft-haas-xiao-bfd-echo-path-mtu-01 (work in progress), July 2011.

[RFC1191] Mogul, J. and S. Deering, "Path MTU discovery", RFC 1191, DOI 10.17487/RFC1191, November 1990, <https://www.rfc-editor.org/info/rfc1191>.

[RFC3719] Parker, J., Ed., "Recommendations for Interoperable Networks using Intermediate System to Intermediate System (IS-IS)", RFC 3719, DOI 10.17487/RFC3719, February 2004, <https://www.rfc-editor.org/info/rfc3719>.

Appendix A. Related Features

IS-IS [RFC3719] supports a Padding feature for its hellos. This provides the ability to detect inconsistent link MTUs.

Authors' Addresses

Jeffrey Haas
Juniper Networks, Inc.
1133 Innovation Way
Sunnyvale, CA 94089
US

Email: jhaas@juniper.net

Albert Fu
Bloomberg L.P.

Email: afu14@bloomberg.net