Internet Engineering Task Force (IETF)
Request for Comments: 9764
Category: Standards Track
ISSN: 2070-1721

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

April 2025

Bidirectional Forwarding Detection (BFD) Encapsulated in Large Packets

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 not only that the path between two systems is reachable, but also that it is capable of carrying a payload of a particular size. This document specifies how to implement such a mechanism using BFD in Asynchronous mode.

YANG modules for managing this mechanism are also defined in this document. These YANG modules augment the existing BFD YANG modules defined in RFC 9314. The YANG modules in this document conform to the Network Management Datastore Architecture (NMDA) (RFC 8342).

Status of This Memo

This is an Internet Standards Track document.

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

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at https://www.rfc-editor.org/info/rfc9764.

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 criterion. 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 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 (e.g., [BFD-ECHO-PATH-MTU]) have been made for testing Path MTU for such directly connected systems. However, in the case of multihop BFD [RFC5883], this guarantee does not hold.

The encapsulation of BFD in multihop sessions is a simple UDP packet. The BFD elements of procedure (Section 6.8.6 of [RFC5880]) cover 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. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

3. 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 (in bytes) is increased to this value. The contents of this additional payload MUST be zero. The contents of this additional payload SHOULD NOT be validated by the receiver. The minimum size of this variable MUST NOT be smaller than 24 or 26 bytes, as permitted by the element of BFD

procedure; 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.

4. Implementation and Deployment Considerations

4.1. 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.

Section 6.8.6 of [RFC5880] discusses the procedures for receiving BFD Control packets. The length of the BFD Control packet is validated to be less than or equal to the payload of the encapsulating protocol. When a receiving 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 receiving implementation would follow normal BFD procedures regarding not having received Control packets.

If large BFD packets is enabled on a session that is already in the Up state and the remote BFD system does not (or cannot) support receiving the padded BFD control packets, the session will go Down.

4.2. Selecting MTU Size To Be Detected

Since the consideration is Path MTU, BFD sessions using this feature only need to use an appropriate value of bfd.PaddedPduSize 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 1512 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. Failure to select the largest size will result in BFD sessions going to the Up state and dependent applications not having their MTU requirements satisfied.

4.3. 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 bidirectional Path MTU, it is necessary to configure the

bfd.PaddedPduSize parameter on each side of the BFD session. For example, if the desire is to verify a 1512-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 1512. In the absence of such consistent configuration, BFD with Large Packets may correctly determine unidirectional connectivity at the tested MTU, but bidirectional 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 BFD session supports such scenarios.

4.4. Detecting MTU Changes

Once BFD sessions using Large Packets has reached the Up state, connectivity at the tested MTU(s) for the session is being validated. If the Path MTU tested by the BFD with Large Packets session falls below the tested MTU, the BFD session will go Down.

In the opposite circumstance (where the Path MTU increases), the BFD session will continue without being impacted. BFD for Large Packets only ensures that the minimally acceptable MTU for the session can be used.

4.5. Equal-Cost Multipath (ECMP) or Other Load-Balancing Considerations

Various mechanisms are utilized to increase throughput between two endpoints at various network layers. Such features include Link Aggregation Groups (LAGs) or ECMP forwarding. Such mechanisms balance traffic across multiple physical links while hiding the details of that balancing 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 interfaces, [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 implementation-specific load balancing, it is not possible to standardize such additional mechanisms for BFD.

Misconfiguration 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.

This section applies to most, if not all, BFD techniques.

4.6. S-BFD

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

5. BFD Encapsulated in Large Packets YANG Module

5.1. Data Model Overview

This YANG module augments the "ietf-bfd" module to add a flag ’padding’ to enable this feature. The feature statement ’padding’ needs to be enabled to indicate that BFD encapsulated in large packets is supported by the implementation.

Further, this YANG module augments the YANG modules for single-hop, multihop, LAG, and MPLS to add the "pdu-size" parameter to those session types to configure large BFD packets.

Finally, similar to the grouping "client-cfg-parms" defined in Section 2.1 of [RFC9314], this YANG module defines a grouping "bfd-large-common" that may be utilized by BFD clients using "client-cfg-params" to uniformly add support for the feature defined in this RFC.

module: ietf-bfd-large

```
  augment /rt:routing/rt:control-plane-protocols
            /rt:control-plane-protocol/bfd:bfd/bfd-ip-sh:ip-sh
            /bfd-ip-sh:sessions/bfd-ip-sh:session:
    +--rw pdu-size?   padded-pdu-size {padding}?
  augment /rt:routing/rt:control-plane-protocols
            /rt:control-plane-protocol/bfd:bfd/bfd-ip-mh:ip-mh
            /bfd-ip-mh:session-groups/bfd-ip-mh:session-group:
    +--rw pdu-size?   padded-pdu-size {padding}?
  augment /rt:routing/rt:control-plane-protocols
            /rt:control-plane-protocol/bfd:bfd/bfd-lag:lag
            /bfd-lag:sessions/bfd-lag:session:
    +--rw pdu-size?   padded-pdu-size {padding}?
  augment /rt:routing/rt:control-plane-protocols
            /rt:control-plane-protocol/bfd:bfd/bfd-mpls:mpls
            /bfd-mpls:session-groups/bfd-mpls:session-group:
    +--rw pdu-size?   padded-pdu-size {padding}?
```

Figure 1

5.2. YANG Module

This YANG module imports "A YANG Data Model for Routing Management (NMDA Version)" [RFC8349] and "YANG Data Model for Bidirectional Forwarding Detection (BFD)" [RFC9314].

<CODE BEGINS> file "ietf-bfd-large@2025-04-04.yang"

```
module ietf-bfd-large {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-bfd-large";
  prefix bfdl;

  import ietf-routing {
    prefix rt;
    reference
      "RFC 8349: A YANG Data Model for Routing Management
       (NMDA version)";
  }

  import ietf-bfd {
    prefix bfd;
    reference
      "RFC 9314: YANG Data Model for Bidirectional
       Forwarding Detection.";
  }
```

```
import ietf-bfd-ip-sh {
  prefix bfd-ip-sh;
  reference
    "RFC 9314: YANG Data Model for Bidirectional
     Forwarding Detection.";
}

import ietf-bfd-ip-mh {
  prefix bfd-ip-mh;
  reference
    "RFC 9314: YANG Data Model for Bidirectional
     Forwarding Detection.";
}

import ietf-bfd-lag {
  prefix bfd-lag;
  reference
    "RFC 9314: YANG Data Model for Bidirectional
     Forwarding Detection.";
}

import ietf-bfd-mpls {
  prefix bfd-mpls;
  reference
    "RFC 9314: YANG Data Model for Bidirectional
     Forwarding Detection.";
}

organization
  "IETF BFD Working Group";

contact
  "WG Web:   <https://datatracker.ietf.org/wg/bfd>
   WG List:  <rtg-bfd@ietf.org>

   Authors: Jeffrey Haas (jhaas@juniper.net)
            Albert Fu (afu14@bloomberg.net).";

description
  "This YANG module augments the base BFD YANG module to add
   attributes related to support for BFD Encapsulated in Large
   Packets.  In particular, it adds a per-session parameter for the
   BFD Padded PDU Size.

   Copyright (c) 2025 IETF Trust and the persons identified as
   authors of the code.  All rights reserved.

   Redistribution and use in source and binary forms, with or
   without modification, is permitted pursuant to, and subject to
   the license terms contained in, the Revised BSD License set
   forth in Section 4.c of the IETF Trust's Legal Provisions
   Relating to IETF Documents
   (https://trustee.ietf.org/license-info).

   This version of this YANG module is part of RFC 9764
   (https://www.rfc-editor.org/info/rfc9764); see the RFC itself
   for full legal notices.

   The key words 'MUST', 'MUST NOT', 'REQUIRED', 'SHALL', 'SHALL
   NOT', 'SHOULD', 'SHOULD NOT', 'RECOMMENDED', 'NOT RECOMMENDED',
   'MAY', and 'OPTIONAL' in this document are to be interpreted as
   described in BCP 14 (RFC 2119) (RFC 8174) when, and only when,
   they appear in all capitals, as shown here.";

revision 2025-04-04 {
  description
```

```
    "Initial Version.";
  reference
    "RFC 9764, Bidirectional Forwarding Detection (BFD)
     Encapsulated in Large Packets.";
}

feature padding {
  description
    "If supported, the feature allows for BFD sessions to be
     configured with padded PDUs in support of BFD Encapsulated in
     Large Packets.";
}

typedef padded-pdu-size {
  type uint16 {
    range "24..65535";
  }
  units "bytes";
  description
    "The size of the padded and encapsulated BFD control packets
     to be transmitted at Layer 3.  The BFD minimum control packet
     size is 24 or 26 octets; see Section 6.8.6 of RFC 5880.

     If the configured padded PDU size is smaller than the minimum
     sized packet of a given BFD session, then the minimum sized
     packet for the session will be used.

     The maximum padded PDU size may be limited by the supported
     interface MTU of the system.";
  reference
    "RFC 9764, Bidirectional Forwarding Detection (BFD)
     Encapsulated in Large Packets.";
}

grouping bfd-large-common {
  description
    "Common configuration and operational state for BFD
     Encapsulated in Large Packets.";
  reference
    "RFC 9764, Bidirectional Forwarding Detection (BFD)
     Encapsulated in Large Packets.";
  leaf pdu-size {
    if-feature "padding";
    type padded-pdu-size;
    description
      "If set, this configures the padded PDU size for the
       Asynchronous mode BFD session. By default, no additional
       padding is added to such packets.";
  }
}

augment "/rt:routing/rt:control-plane-protocols/"
      + "rt:control-plane-protocol/bfd:bfd/bfd-ip-sh:ip-sh/"
      + "bfd-ip-sh:sessions/bfd-ip-sh:session" {
  uses bfd-large-common;
  description
    "Augment the 'bfd' container to add attributes related to BFD
     Encapsulated in Large Packets.";
}

augment "/rt:routing/rt:control-plane-protocols/"
      + "rt:control-plane-protocol/bfd:bfd/bfd-ip-mh:ip-mh/"
      + "bfd-ip-mh:session-groups/bfd-ip-mh:session-group" {
  uses bfd-large-common;
  description
    "Augment the 'bfd' container to add attributes related to BFD
```

```
        Encapsulated in Large Packets.";
    }

    augment "/rt:routing/rt:control-plane-protocols/"
          + "rt:control-plane-protocol/bfd:bfd/bfd-lag:lag/"
          + "bfd-lag:sessions/bfd-lag:session" {
      uses bfd-large-common;
      description
        "Augment the 'bfd' container to add attributes related to BFD
         Encapsulated in Large Packets.";
    }

    augment "/rt:routing/rt:control-plane-protocols/"
          + "rt:control-plane-protocol/bfd:bfd/bfd-mpls:mpls/"
          + "bfd-mpls:session-groups/bfd-mpls:session-group" {
      uses bfd-large-common;
      description
        "Augment the 'bfd' container to add attributes related to BFD
         Encapsulated in Large Packets.";
    }
  }
  <CODE ENDS>
```

Figure 2

6. Security Considerations

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

On-path attackers that can selectively drop BFD packets, including those with large MTUs, can cause BFD sessions to go Down.

The contents of the padding payload are set to zero. This avoids implementation issues where the local uninitialized data may be leaked.

6.1. YANG Security Considerations

This section is modeled after the template described in Section 3.7 of [YANG-GUIDELINES].

The "ietf-bfd-large" YANG module defines a data model that is designed to be accessed via YANG-based management protocols, such as NETCONF [RFC6241] and RESTCONF [RFC8040]. These protocols have to use a secure transport layer (e.g., SSH [RFC4252], TLS [RFC8446], and QUIC [RFC9000]) and have to use mutual authentication.

The Network Configuration Access Control Model (NACM) [RFC8341] provides the means to restrict access for particular NETCONF or RESTCONF users to a preconfigured subset of all available NETCONF or RESTCONF protocol operations and content.

There is one data node defined in this YANG module that is writable/creatable/deletable (i.e., "config true", which is the default). All writable data nodes are likely to be reasonably sensitive or vulnerable in some network environments. Write operations (e.g., edit-config) and delete operations to these data nodes without proper protection or authentication can have a negative effect on network operations. The data node has particular sensitivities/vulnerabilities:

* 'pdu-size' specifies the targeted size of BFD control packets encapsulated according to this proposal. Changing this value for a session in the Up state may cause the session to go down, perhaps intentionally, if the session cannot accommodate such BFD

control packets. Operators should be mindful that multiple BFD clients may rely on the status of a given BFD session when changing this value.

There are no particularly sensitive readable data nodes.

There are no particularly sensitive RPC or action operations.

Modules that use the groupings that are defined in this document should identify the corresponding security considerations. For example, reusing some of these groupings will expose privacy-related information (e.g., 'node-example'). This module defines one such grouping, "bfd-large-common", which contains the "pdu-size" data node whose security considerations are documented above.

7. IANA Considerations

7.1. The "IETF XML" Registry

IANA has registered the following URI in the "ns" subregistry of the "IETF XML Registry" [RFC3688].

URI: urn:ietf:params:xml:ns:yang:ietf-bfd-large
Registrant Contact: The IESG
XML: N/A; the requested URI is an XML namespace.

7.2. The "YANG Module Names" Registry

IANA has registered the following YANG module in the "YANG Module Names" registry [RFC6020].

Name: ietf-bfd-large
Maintained by IANA: N
Namespace: urn:ietf:params:xml:ns:yang:ietf-bfd-large
Prefix: bfdl
Reference: RFC 9764

8. References

8.1. 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>.

[RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.17487/RFC3688, January 2004, <https://www.rfc-editor.org/info/rfc3688>.

[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>.

[RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, DOI 10.17487/RFC6020, October 2010, <https://www.rfc-editor.org/info/rfc6020>.

[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>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC8341] Bierman, A. and M. Bjorklund, "Network Configuration Access Control Model", STD 91, RFC 8341, DOI 10.17487/RFC8341, March 2018, <https://www.rfc-editor.org/info/rfc8341>.

[RFC8349] Lhotka, L., Lindem, A., and Y. Qu, "A YANG Data Model for Routing Management (NMDA Version)", RFC 8349, DOI 10.17487/RFC8349, March 2018, <https://www.rfc-editor.org/info/rfc8349>.

[RFC9314] Jethanandani, M., Ed., Rahman, R., Ed., Zheng, L., Ed., Pallagatti, S., and G. Mirsky, "YANG Data Model for Bidirectional Forwarding Detection (BFD)", RFC 9314, DOI 10.17487/RFC9314, September 2022, <https://www.rfc-editor.org/info/rfc9314>.

8.2. Informative References

[BFD-ECHO-PATH-MTU] Min, X., Ed. and J. Haas, Ed., "Application of the BFD Echo function for Path MTU Verification or Detection", Work in Progress, Internet-Draft, draft-haas-xiao-bfd-echo-path-mtu-01, 11 July 2011, <https://datatracker.ietf.org/doc/html/draft-haas-xiao-bfd-echo-path-mtu-01>.

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

[RFC4252] Ylonen, T. and C. Lonvick, Ed., "The Secure Shell (SSH) Authentication Protocol", RFC 4252, DOI 10.17487/RFC4252, January 2006, <https://www.rfc-editor.org/info/rfc4252>.

[RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <https://www.rfc-editor.org/info/rfc6241>.

[RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017, <https://www.rfc-editor.org/info/rfc8040>.

[RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018,

            <https://www.rfc-editor.org/info/rfc8446>.

   [RFC9000] Iyengar, J., Ed. and M. Thomson, Ed., "QUIC: A UDP-Based
              Multiplexed and Secure Transport", RFC 9000,
              DOI 10.17487/RFC9000, May 2021,
              <https://www.rfc-editor.org/info/rfc9000>.

   [YANG-GUIDELINES]
              Bierman, A., Boucadair, M., Ed., and Q. Wu, "Guidelines
              for Authors and Reviewers of Documents Containing YANG
              Data Models", Work in Progress, Internet-Draft, draft-
              ietf-netmod-rfc8407bis-22, 14 January 2025,
              <https://datatracker.ietf.org/doc/html/draft-ietf-netmod-
              rfc8407bis-22>.

Acknowledgments

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

Authors' Addresses

Jeffrey Haas
Juniper Networks, Inc.
1133 Innovation Way
Sunnyvale, CA 94089
United States of America
Email: jhaas@juniper.net

Albert Fu
Bloomberg L.P.
731 Lexington Avenue
New York, NY 10022
United States of America
Email: afu14@bloomberg.net