Secure BFD Sequence Numbers

draft-ietf-bfd-secure-sequence-numbers-08

Abstract

This document describes a security enhancement for the sequence number used in BFD control packets. This document updates RFC 5880.

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 https://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 September 9, 2021.

Copyright Notice

Copyright (c) 2021 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents (https://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect
1. Introduction

BFD [RFC5880] section 6.7 describes the use of monotonically incrementing 32-bit sequence numbers for use in authentication of BFD packets. While this method protects against simple replay attacks, the monotonically incrementing sequence numbers are predictable and vulnerable to more complex attack vectors. This document proposes the use of non-monotonically-incrementing sequence numbers in the BFD authentication section to enhance the security of BFD sessions. Specifically, the document presents a method to generate pseudo-random sequence numbers on the frame by algorithmically hashing monotonically increasing sequence numbers. Since the monotonically increasing sequence number does not appear on the wire, it is difficult for a third party to launch a replay attack.

2. Requirements Language

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 [RFC2119].

3. Theory of operation

Instead of inserting a monotonically, sometimes occasionally, increasing sequence number in BFD control packets, the ciphertext result from a symmetric key algorithm operation (Symmetric-key algorithms require both the sender and the recipient of a message to have the same shared secret key) is inserted. The result is computed, using a shared key, on the sequence number. That
ciphertext result is then inserted into the sequence number field of the packet. In case of BFD Authentication [I-D.ietf-bfd-optimizing-authentication], the sequence number used in computing an authenticated packet would be this new computed ciphertext. Even though the BFD Authentication [I-D.ietf-bfd-optimizing-authentication] sequence number is independent of this enhancement, it would benefit by using the computed ciphertext.

As currently defined in BFD [RFC5880], a BFD packet with authentication will undergo the following steps, where:

\[O\]: original RFC 5880 packet with monotonically increasing sequence number

\[S\]: pseudo random sequence number

\[A\]: Authentication

\[
\begin{array}{cccc}
\text{Sender} & \text{Receiver} \\
\hline
\[O\] & \[S\] & \[A\] & \[A\] \quad \[S\] \quad \[O\]
\end{array}
\]

This document proposes that for enhanced security in sequence number encoding, the sender would identify a symmetric key algorithm that would create a 32 bit ciphertext. The symmetric key is provisioned securely on the sender and receiver of the BFD session. The mechanism of provisioning such a key is outside the scope of this document. This key SHOULD be different from the symmetric key used to authenticate the packet. Instead of sending the sequence number, the sender encrypts the sequence number using it as input to the symmetric algorithm to produce the ciphertext, which is then inserted in place of the sequence number.

Upon receiving the BFD Control packet, the receiver decrypts the ciphertext using the same provisioned shared key to produce the received sequence number. It compares the received sequence number against the expected sequence number. The mechanism used for comparing is an implementation detail (implementations may pre-calculate the expected sequence number, or decrypt the received sequence number before comparing against expected value). To tolerate dropped frames, the receiver must compare the received sequence number against the current expected sequence number. BFD [RFC5880] mentions that received sequence number should be between \(\text{bd.RcvAuthSeq}(+1)\) to \(\text{bd.RcvAuthSeq}+(3\times\text{Detect Mult})\) inclusive. Note: The first sequence number can be obtained using the same principles stated in BFD [RFC5880] i.e. (using \(\text{bd.AuthSeqKnown}\) and \(\text{bd.RcvAuthSeg}\))
K: symmetric key
S: sequence number
S': encrypted sequence number OR ciphertext result
O: original RFC 5880 packet with monotonically increasing sequence number

\[ f(S, K) = S', \text{ where } f \text{ is a symmetric encryption algorithm} \]

\[ f(S', K) = S, \text{ where } f \text{ is a symmetric decryption algorithm} \]

The above diagram describes how the sender encrypts and receiver decrypts the sequence number. The sender starts by taking the monotonically increasing (but non linear) sequence number and encrypting it using a symmetric encryption algorithm. The resulting ciphertext replaces the sequence number. As per BFD [RFC5880], it then calculates the hash for the entire packet and appends the hash value to the end of the packet, before transmitting it.

The receiver hashes the entire packet as part of receiver authentication. On successful authentication, it decrypts the ciphertext with the same key used to encrypt it, in order to obtain the original sequence number. If it is greater than the previously received monotonically increasing sequence number, then the receiver knows it’s a valid sequence number.

4. Impact of using a hash

Under this proposal, every packet’s sequence number is encoded in ciphertext. Therefore, there is some impact on the system and its performance while doing encryption/decryption. As security measures go, this enhancement greatly increases the security of the packet with or without authentication of the entire packet.

5. IANA Considerations

This document makes no request of IANA.

Note to RFC Editor: this section may be removed on publication as an RFC.
6. Security Considerations

In a symmetric key algorithm, the key is shared between the two systems. Distribution of this key to all the systems at the same time can be quite a cumbersome task. BFD sessions running a fast rate will require these keys to be refreshed often, which poses a further challenge. Therefore, it is difficult to change the keys during the operation of a BFD session without affecting the stability of the BFD session. Therefore, it is recommended to administratively disable the BFD session before changing the keys. If the keys are not changed frequently, an attacker can try to guess the key to launch a replay attack.

This method allows the BFD end-points to detect a malicious packet (the decrypted sequence number will not be in sequence). The behavior of the session, when such a packet is detected, is based on the implementation. A flood of such malicious packets may cause a BFD session to be operationally down.

The symmetric algorithm and key size will determine the difficulty for an attacker to decipher the key from the transmitted BFD frames. The sequential nature of the payload (sequence numbers) simplifies the decoding of the key. It is, therefore, recommended to use longer keys or more secure symmetric algorithms.

7. Acknowledgements

The authors would like to thank Jeff Haas and Reshad Rahman for their reviews of and suggestions for the document.

8. References

8.1. Normative References


8.2. Informative References

[I-D.ietf-bfd-optimizing-authentication]
Jethanandani, M., Mishra, A., Saxena, A., and M. Bhatia,

Authors’ Addresses

Mahesh Jethanandani
Kloud Services
Email: mjethanandani@gmail.com

Sonal Agarwal
Cisco Systems, Inc
170 W. Tasman Drive
San Jose, CA  95070
USA
Email: agarwaso@cisco.com
URI:   www.cisco.com

Ashesh Mishra
O3b Networks
Email: mishra.ashesh@gmail.com

Ankur Saxena
Ciena Corporation
3939 North First Street
San Jose, CA  95134
USA
Email: ankurpsaxena@gmail.com

Alan DeKok
Network RADIUS SARL
100 Centrepointe Drive #200
Ottowa, ON  K2G 6B1
Canada
Email: aland@freeradius.org
BFD Stability
draft-ietf-bfd-stability-10

Abstract
This document describes extensions to the Bidirectional Forwarding Detection (BFD) protocol to measure BFD stability. Specifically, it describes a mechanism for detection of BFD packet loss.

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 https://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 October 14, 2021.

Copyright Notice
Copyright (c) 2021 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents
1. Introduction

The Bidirectional Forwarding Detection (BFD) [RFC5880] protocol operates by transmitting and receiving BFD control packets, generally at high frequency, over the datapath being monitored. In order to prevent significant data loss due to a datapath failure, BFD session detection time as defined in BFD [RFC5880] is set to the smallest feasible value.

This document proposes a mechanism to detect lost packets in a BFD session in addition to the datapath fault detection mechanisms of BFD. Such a mechanism presents significant value to measure the stability of BFD sessions and provides data to the operators for the cause of a BFD failure.

This document does not propose any BFD extension to measure data traffic loss or delay on a link or tunnel and the scope is limited to BFD packets.
2. Terminology

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 RFC 2119 [RFC2119] and RFC 8174 [RFC8174].

The reader is expected to be familiar with the BFD [RFC5880], Optimizing BFD Authentication [I-D.ietf-bfd-optimizing-authentication] and BFD Secure Sequence Numbers [I-D.ietf-bfd-secure-sequence-numbers].

3. Use Cases

Bidirectional Forwarding Detection as defined in BFD [RFC5880] cannot detect any BFD packet loss if the loss does not last for detection time. This document proposes a method to detect a dropped packet on the receiver. For example, if the receiver receives BFD control packet k at time t but receives packet k+3 at time t+10ms, and never receives packet k+1 and/or k+2, then it has experienced a drop.

This proposal enables BFD implementations to generate diagnostic information on the health of each BFD session that could be used to preempt a failure on a datapath that BFD was monitoring by allowing time for a corrective action to be taken.

In a faulty datapath scenario, an operator can use BFD health information to trigger delay and loss measurement OAM protocol, Connectivity Fault Management (CFM) [IEEE802.lag] or Loss Measurement (LM)-Delay Measurement (DM) as defined by A One-way Active Measurement Protocol (OWAMP) [RFC4656] to further isolate the issue.

4. BFD NULL-Authentication Type

The functionality proposed for BFD stability measurement is achieved by appending an authentication section with the NULL Authentication type (as defined in Optimizing BFD Authentication [I-D.ietf-bfd-optimizing-authentication] ) to the BFD control packets that do not have authentication enabled.

5. Theory of Operation

This mechanism allows operators to measure the loss of BFD control packets.

When using MD5 or SHA authentication, BFD uses an authentication section that carries the Sequence Number. However, if non-meticulous authentication is being used, or no authentication is in use, then
5.1. Loss Measurement

Loss measurement counts the number of BFD control packets missed at the receiver during any Detection Time period. The loss is detected by comparing the Sequence Number field in the Auth TLV (NULL or otherwise) in successive BFD control packets. The Sequence Number in each successive control packet generated on a BFD session by the transmitter is incremented by one. This loss count can then be exposed using the YANG module defined in the subsequent section.

The first BFD authentication section with a non-zero sequence number, in a valid BFD control packet, processed by the receiver is used for bootstrapping the logic. When using secure sequence numbers, if the expected values are pre-calculated, the value must be matched to detect lost packets as defined in BFD secure sequence numbers [I-D.ietf-bfd-secure-sequence-numbers].

6. ietf-bfd-stability YANG Module

6.1. Data Model Overview

This YANG module augments the "ietf-bfd" module to add to the per-session set of counters a 'loss-packet-count' for BFD packets that are lost but do not necessarily result in the BFD session going down.
module: ietf-bfd-stability
augment /rt:routing/rt:control-plane-protocols
  /bfd-ip-sh:sessions/bfd-ip-sh:session
  /bfd-ip-sh:session-statistics:
    +++ro lost-packet-count? yang:counter32
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
  /bfd-ip-mh:sessions/bfd-ip-mh:session-statistics:
    +++ro lost-packet-count? yang:counter32
augment /rt:routing/rt:control-plane-protocols
  /rt:control-plane-protocol/bfd:bfd/bfd-lag:lag
  /bfd-lag:sessions/bfd-lag:session/bfd-lag:member-links
  /bfd-lag:micro-bfd-ipv4/bfd-lag:session-statistics:
    +++ro lost-packet-count? yang:counter32
augment /rt:routing/rt:control-plane-protocols
  /rt:control-plane-protocol/bfd:bfd/bfd-lag:lag
  /bfd-lag:sessions/bfd-lag:session/bfd-lag:member-links
  /bfd-lag:micro-bfd-ipv6/bfd-lag:session-statistics:
    +++ro lost-packet-count? yang:counter32
augment /rt:routing/rt:control-plane-protocols
  /rt:control-plane-protocol/bfd:bfd/bfd-mpls:mpls
  /bfd-mpls:session-groups/bfd-mpls:session-group
  /bfd-mpls:sessions/bfd-mpls:session-statistics:
    +++ro lost-packet-count? yang:counter32

6.2. YANG Module

This YANG module imports Common YANG Types [RFC6991], A YANG Data Model for Routing [RFC8349], and YANG Data Model for Bidirectional Forwarding Detection (BFD) [I-D.ietf-bfd-yang].

<CODE BEGINS> file "ietf-bfd-stability@2021-04-11.yang"
module ietf-bfd-stability {
  yang-version 1.1;
  prefix "bfds";

  import ietf-yang-types {
    prefix "yang";
    reference
      "RFC 6991: Common YANG Data Types";
  }

  import ietf-routing {
    prefix "rt";
    reference

"RFC 8349: A YANG Data Model for Routing Management
(NM3A version)";

import ietf-bfd {
  prefix bfd;
  reference
    "I-D.ietf-bfd-yang: YANG Data Model for Bidirectional
       Forwarding Detection.";
}

import ietf-bfd-ip-sh {
  prefix bfd-ip-sh;
  reference
    "I-D.ietf-bfd-yang: YANG Data Model for Bidirectional
       Forwarding Detection.";
}

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

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

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

organization
    "IETF BFD Working Group";

contact
  "WG Web:  <http://tools.ietf.org/wg/bfd>
WG List:  <bfd@ietf.org>

Authors: Mahesh Jethanandani (mjethanandani@gmail.com)
          Ashesh Mishra (mishra.ashesh@gmail.com)
This YANG module augments the base BFD YANG model to add attributes related to BFD Stability. In particular it adds a per session count for BFD packets that are lost.

Copyright (c) 2021 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 Simplified 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 XXXX (https://www.rfc-editor.org/info/rfcXXXX); see the RFC itself for full legal notices.


revision "2021-04-11" {
    description
        "Initial Version.";
    reference
        "RFC XXXX, BFD Stability.";
}

augment "/rt:routing/rt:control-plane-protocols/" +
    "rt:control-plane-protocol/bfd:bfd-ip-sh:ip-sh/" +
    "bfd-ip-sh:sessions/bfd-ip-sh:session/" +
    "bfd-ip-sh:session-statistics"
leaf lost-packet-count {
    type yang:counter32;
    description
        "Number of BFD packets that were lost without bringing the session down.";
}
description

"Augment the 'bfd' container to add attributes related to BFD stability.";
}

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/" +
  "bfd-ip-mh:sessions/bfd-ip-mh:session-statistics" {
  leaf lost-packet-count {
    type yang:counter32;
    description
    "Number of BFD packets that were lost without bringing the
     session down.";
  }
  description
  "Augment the 'bfd' container to add attributes related to BFD
   stability.";
}

augment "/rt:routing/rt:control-plane-protocols/" +
  "rt:control-plane-protocol/bfd:bfd-lag:lag/" +
  "bfd-lag:sessions/bfd-lag:session/bfd-lag:member-links/" +
  "bfd-lag:micro-bfd-ipv4/bfd-lag:session-statistics" {
  leaf lost-packet-count {
    type yang:counter32;
    description
    "Number of BFD packets that were lost without bringing the
     session down.";
  }
  description
  "Augment the 'bfd' container to add attributes related to BFD
   stability.";
}

augment "/rt:routing/rt:control-plane-protocols/" +
  "rt:control-plane-protocol/bfd:bfd-lag:lag/" +
  "bfd-lag:sessions/bfd-lag:session/bfd-lag:member-links/" +
  "bfd-lag:micro-bfd-ipv6/bfd-lag:session-statistics" {
  leaf lost-packet-count {
    type yang:counter32;
    description
    "Number of BFD packets that were lost without bringing the
     session down.";
  }
  description
  "Augment the 'bfd' container to add attributes related to BFD
   stability.";
augment "/rt:routing/rt:control-plane-protocols/" +
    "rt:control-plane-protocol/bfd:bfd/bfd-mpls:mpls/" +
    "bfd-mpls:session-groups/bfd-mpls:session-group/" +
    "bfd-mpls:sessions/bfd-mpls:session-statistics" {
  leaf lost-packet-count {
    type yang:counter32;
    description
      "Number of BFD packets that were lost without bringing the
       session down."
  }
  description
    "Augment the 'bfd' container to add attributes related to BFD
     stability."
}

7. IANA Considerations

7.1. The "IETF XML" Registry

This document registers one URIs in the "ns" subregistry of the "IETF
XML" registry [RFC3688]. Following the format in [RFC3688], the
following registration is requested:

Registrant Contact: The IESG
XML: N/A, the requested URI is an XML namespace.

7.2. The "YANG Module Names" Registry

This document registers one YANG module in the "YANG Module Names"
registry YANG [RFC6020]. Following the format in YANG [RFC6020], the
following registrations are requested:

name:         ietf-bfd-stability
prefix:       bfds
reference:    RFC XXXX

8. Security Consideration

The YANG module specified in this document defines a schema for data
that is designed to be accessed via network management protocols such
as NETCONF [RFC6241] or RESTCONF [RFC8040]. The lowest NETCONF layer
is the secure transport layer, and the mandatory-to-implement secure
transport is Secure Shell (SSH) [RFC6242]. The lowest RESTCONF layer is HTTPS, and the mandatory-to-implement secure transport is TLS [RFC8446]. The NETCONF 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.

The YANG module does not define any writeable/creatable/deletable data nodes.

The only readable data nodes in YANG module may be considered sensitive or vulnerable in some network environments. It is thus important to control read access (e.g., via get, get-config, or notification) to these data nodes. The model does not define any readable subtrees and data nodes.

The YANG module does not define any RPC operations.

9. Contributors

Manav Bhatia

10. Acknowledgements

Authors would like to thank Nobo Akiya, Jeffery Haas, Dileep Singh, Basil Saji, Sagar Soni, Albert Fu and Mallik Mudigonda who also contributed to this document.

11. References

11.1. Normative References

[I-D.ietf-bfd-optimizing-authentication]

[I-D.ietf-bfd-secure-sequence-numbers]
[I-D.ietf-bfd-yang]


Access Control Model", STD 91, RFC 8341,
DOI 10.17487/RFC8341, March 2018,

Routing Management (NMDA Version)", RFC 8349,
DOI 10.17487/RFC8349, March 2018,

Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018,

11.2. Informative References

[IEEE802.1ag] Institute of Electrical and Electronics Engineers, Inc.,
"802.1ag - Connectivity Fault Management", September 2007,

[RFC4656] Shalunov, S., Teitelbaum, B., Karp, A., Boote, J., and M.
Zekauskas, "A One-way Active Measurement Protocol
(OWAMP)", RFC 4656, DOI 10.17487/RFC4656, September 2006,

Authors’ Addresses

Ashesh Mishra
SES

Email: mishra.ashesh@gmail.com

Mahesh Jethanandani
Kloud Services
CA
USA

Email: mjethanandani@gmail.com
Ankur Saxena
Ciena Corporation
3939 North 1st Street
San Jose, CA  95134
USA
Email: ankurpsaxena@gmail.com
URI:   www.ciena.com

Santosh Pallagatti
VMware
Bangalore, Karnataka  560103
India
Email: santosh.pallagatti@gmail.com

Mach Chen
Huawei
Email: mach.chen@huawei.com

Peng Fan
China Mobile
32 Xuanwumen West Street
Beijing, Beijing
China
Email: fanp08@gmail.com
Unaffiliated BFD Echo Function
draft-ietf-bfd-unaffiliated-echo-02

Abstract

Bidirectional Forwarding Detection (BFD) is a fault detection protocol that can quickly determine a communication failure between two forwarding engines. This document proposes a use of the BFD Echo function where the local system supports BFD but the neighboring system does not support BFD.

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 https://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 December 24, 2021.

Copyright Notice

Copyright (c) 2021 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents (https://trustee.ietf.org/license-info) in effect on the date of
1. Introduction

To minimize the impact of device/link faults on services and improve network availability, a network device must be able to quickly detect faults in communication with adjacent devices. Measures can then be taken to promptly rectify the faults to ensure service continuity.

BFD [RFC5880] is a low-overhead, short-duration method to detect faults on the communication path between adjacent forwarding engines. The faults can be on interfaces, data link(s), and even the forwarding engines. It is a single, unified mechanism to monitor any media and protocol layers in real time.

BFD defines an Asynchronous mode to satisfy various deployment scenarios. It also supports an Echo function to reduce the device requirement for BFD. When the Echo function is activated, the local system sends BFD Echo packets and the remote system loops back the received Echo packets through the forwarding path. If several consecutive BFD Echo packets are not received by the local system, then the BFD session is declared to be Down.

When using BFD Echo function, there are two typical scenarios as below:
o Full BFD protocol capability with affiliated Echo function: This scenario requires both the local device and the neighboring device to support the full BFD protocol.

o BFD Echo-Only function without full BFD protocol capability: This scenario requires only the local device to support sending and demultiplexing BFD Control packets.

The latter scenario is referred to as Unaffiliated BFD Echo function in this document.

Section 6.2.2 of [BBF-TR-146] describes one use case of the Unaffiliated BFD Echo function, and at least one more use case is known to be deployed.

This document describes the use of the Unaffiliated BFD Echo function over IPv4 and IPv6 for single IP hop.

1.1. Conventions Used in This Document

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.

2. Updates to RFC 5880

The Unaffiliated BFD Echo function described in this document reuses the BFD Echo function as described in [RFC5880] and [RFC5881], but does not require BFD Asynchronous mode. When using the Unaffiliated BFD Echo function, only the local system has the BFD protocol enabled; the remote system just loops back the received BFD Echo packets as regular data packets.

This document updates [RFC5880] with respect to its descriptions on the BFD Echo function as follows.

o The 4th paragraph of Section 3.2 of [RFC5880] is updated as below:

OLD TEXT
An adjunct to both modes is the Echo function.

NEW TEXT
An adjunct or complement to both modes is the Echo function.
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).

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). The Echo function may also be used independently, with neither Asynchronous nor Demand mode.

- The 3rd and 9th paragraphs of Section 6.1 of [RFC5880] are updated as below:

- The 2nd paragraph of Section 6.4 of [RFC5880] is updated as below:
When a system is using the Echo function, it is advantageous to choose a sedate reception rate for Control packets, since liveness detection is being handled by the Echo packets.

NEW TEXT

When a system is using the Echo function with Asynchronous mode, it is advantageous to choose a sedate reception rate for Control packets, since liveness detection is being handled by the Echo packets.

- The 2nd paragraph of Section 6.8 of [RFC5880] is updated as below:

OLD TEXT

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

NEW TEXT

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

- The 7th paragraph of Section 6.8.3 of [RFC5880] is updated as below:

OLD TEXT

When the Echo function is active, a system SHOULD set bfd.RequiredMinRxInterval to a value of not less than one second (1,000,000 microseconds).

NEW TEXT

When the Echo function is active with Asynchronous mode, a system SHOULD set bfd.RequiredMinRxInterval to a value of not less than one second (1,000,000 microseconds).

- The 1st and 2nd paragraphs of Section 6.8.9 of [RFC5880] are updated as below:

OLD TEXT
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.

NEW TEXT

When a system is using the Echo function with either Asynchronous or Demand mode, BFD Echo packets MUST NOT be transmitted when bfd.SessionState is not Up, and 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.

OLD TEXT

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...

NEW TEXT

When a system is using the Echo function with either Asynchronous or Demand mode, BFD Echo packets MAY be transmitted when bfd.SessionState is Up, and 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...

3. Unaffiliated BFD Echo Procedures

As shown in Figure 1, device A supports BFD, whereas device B does not support BFD. Device A would send BFD Echo packets, and after receiving the BFD Echo packets sent from device A, the one-hop-away BFD peer device B immediately loops them back by normal IP forwarding, this allows device A to rapidly detect a connectivity loss to device B. Note that device B would not intercept any received BFD Echo packet or parse any BFD protocol field within the BFD Echo packet.

To rapidly detect any IP forwarding faults between device A and device B, a BFD Echo session MUST be created at device A, and the BFD Echo session is RECOMMENDED to follow the BFD state machine defined in Section 6.2 of [RFC5880], except that the received state is not sent but echoed from the remote system, and AdminDown state is ruled out because AdminDown effectively means removal of BFD Echo session. In this case, although BFD Echo packets are transmitted with destination UDP port 3785 as defined in [RFC5881], the BFD Echo
packets sent by device A are BFD Control packets too, the looped BFD Echo packets back from device B would drive BFD state change at device A, substituting the BFD Control packets sent from the BFD peer. Also note that when device A receives looped BFD Control packets, the validation procedures of [RFC5880] are used.

Once a BFD Echo session is created at device A, it starts sending BFD Echo packets, which SHOULD include a BFD Echo session demultiplexing field, such as BFD "Your Discriminator" defined in [RFC5880] (BFD "My Discriminator" can be set to 0 to avoid confusion), except that device A can use IP source address or UDP source port to demultiplex BFD Echo session, or there is only one BFD Echo session running at device A. Device A would send BFD Echo packets with IP destination address destined for itself, such as the IP address of interface 1 of device A. All BFD Echo packets for the session MUST be sent with a Time to Live (TTL) or Hop Limit value of 255.

"Desired Min TX Interval" and "Required Min RX Interval" defined in [RFC5880] may be populated with one second within the BFD Echo packet, which however has no real application and would be ignored by the receiver.

Considering the BFD peer wouldn’t advertise "Required Min Echo RX Interval" as defined in [RFC5880], the transmission interval for sending BFD Echo packets MUST be provisioned at device A, how to make sure the BFD peer is willing and able to loop back BFD Echo packets sent with the provisioned transmission interval is outside the scope of this document. Similar to what’s specified in [RFC5880], the BFD Echo session begins with the periodic, slow transmission of BFD Echo packets, the slow transmission rate SHOULD be no less than one second per packet, until the session is Up, after that the provisioned transmission interval is applied, and reverting back to the slow rate once the session goes Down. Considering the BFD peer wouldn’t advertise "Detect Mult" as defined in [RFC5880], the "Detect Mult" for calculating the Detection Time MUST be provisioned at device A, the Detection Time in device A is equal to the provisioned "Detect Mult" multiplied by the provisioned transmission interval.

Figure 1: Unaffiliated BFD Echo diagram
4. Unaffiliated BFD Echo Applicability

Some devices that would benefit from the use of BFD may be unable to support the full BFD protocol. Examples of such devices include servers running virtual machines, or Internet of Things (IoT) devices. The Unaffiliated BFD Echo function can be used when two devices are connected and only one of them supports the BFD protocol, and the other is capable of looping BFD Echo packets.

5. Security Considerations

All Security Considerations from [RFC5880] and [RFC5881] apply.

Note that the Unaffiliated BFD Echo function prevents the use of Unicast Reverse Path Forwarding (URPF) [RFC3704] [RFC8704] in strict mode.

As specified in Section 5 of [RFC5880], since BFD Echo packets may be spoofed, some form of authentication SHOULD be included. Considering the BFD Echo packets in this document are also BFD Control packets, the "Authentication Section" as defined in [RFC5880] for BFD Control packet is RECOMMENDED to be included within the BFD Echo packet.

In order to mitigate the potential reflector attack by the remote attackers, or infinite loop of the BFD Echo packets, it’s RECOMMENDED to put two requirements on the device looping BFD Echo packets, the first one is that a packet SHOULD NOT be looped unless it has a TTL or Hop Limit value of 255, and the second one is that a packet being looped MUST NOT reset the TTL or Hop Limit value to 255, and MUST use a TTL or Hop Limit value of 254.

6. IANA Considerations

This document has no IANA action requested.

7. Acknowledgements

The authors would like to acknowledge Ketan Talaulikar, Greg Mirsky and Santosh Pallagatti for their careful review and very helpful comments.

The authors would like to acknowledge Jeff Haas for his insightful review and very helpful comments.
8. Contributors

Liu Aihua
ZTE
Email: liu.aihua@zte.com.cn

Qian Xin
ZTE
Email: qian.xin2@zte.com.cn

Zhao Yanhua
ZTE
Email: zhao.yanhua3@zte.com.cn

9. References

9.1. Normative References


9.2. Informative References


Authors’ Addresses

Weiqiang Cheng
China Mobile
Beijing
China
Email: chengweiqiang@chinamobile.com

Ruixue Wang
China Mobile
Beijing
China
Email: wangruixue@chinamobile.com

Xiao Min
ZTE Corp.
Nanjing
China
Email: xiao.min2@zte.com.cn

Reshad Rahman
Individual
Kanata
Canada
Email: reshad@yahoo.com

Raj Chetan Boddireddy
Juniper Networks
Email: rchetan@juniper.net
Unsolicited BFD for Sessionless Applications
draft-ietf-bfd-unsolicited-07

Abstract

For operational simplification of "sessionless" applications using BFD, in this document we present procedures for "unsolicited BFD" that allow a BFD session to be initiated by only one side, and be established without explicit per-session configuration or registration by the other side (subject to certain per-interface or per-router policies).

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.

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 https://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 27 April 2022.
1.  Introduction

The current implementation and deployment practice for BFD ([RFC5880] and [RFC5881]) usually requires BFD sessions be explicitly configured or registered on both sides. This requirement is not an issue when an application like BGP [RFC4271] has the concept of a "session" that involves both sides for its establishment. However, this requirement can be operationally challenging when the prerequisite "session" does not naturally exist between two endpoints in an application. Simultaneous configuration and coordination may be required on both sides for BFD to take effect. For example:
* When BFD is used to keep track of the "liveness" of the nexthop of static routes. Although only one side may need the BFD functionality, currently both sides need to be involved in specific configuration and coordination and in some cases static routes are created unnecessarily just for BFD.

* When BFD is used to keep track of the "liveness" of the third-party nexthop of BGP routes received from the Route Server [RFC7947] at an Internet Exchange Point (IXP). As the third-party nexthop is different from the peering address of the Route Server, for BFD to work, currently two routers peering with the Route Server need to have routes and nexthops from each other (although indirectly via the Router Server), and the nexthop of each router must be present at the same time. These issues are also discussed in [I-D.ietf-idr-rs-bfd].

Clearly it is beneficial and desirable to reduce or eliminate unnecessary configurations and coordination in these "sessionless" applications using BFD.

In this document we present procedures for "unsolicited BFD" that allow a BFD session to be initiated by only one side, and be established without explicit per-session configuration or registration by the other side (subject to certain per-interface or per-router policies).

With "unsolicited BFD" there is potential risk for excessive resource usage by BFD from "unexpected" remote systems. To mitigate such risks, several mechanisms are recommended in the Security Considerations section.

Compared to the "Seamless BFD" [RFC7880], this proposal involves only minor procedural enhancements to the widely deployed BFD itself. Thus we believe that this proposal is inherently simpler in the protocol itself and deployment. As an example, it does not require the exchange of BFD discriminators over an out-of-band channel before the BFD session bring-up.

When BGP Add-Path [RFC7911] is deployed at an IXP using the Route Server, multiple BGP paths (when exist) can be made available to the clients of the Router Server as described in [RFC7947]. The "unsolicited BFD" can be used in BGP route selection by these clients to eliminate paths with "inaccessible nexthops".

2. Procedures for Unsolicited BFD

With "unsolicited BFD", one side takes the "Active role" and the other side takes only the "Passive role" as described in [RFC5880].
On the passive side, the "unsolicited BFD" SHOULD be explicitly configured on an interface or globally (apply to all interfaces). The BFD parameters can be either per-interface or per-router based. It MAY also choose to use the parameters that the active side uses in its BFD Control packets. The "My Discriminator", however, MUST be chosen to allow multiple unsolicited BFD sessions.

The active side starts sending the BFD Control packets as specified in [RFC5880]. The passive side does not send BFD Control packets.

When the passive side receives a BFD Control packet from the active side with 0 as "Your Discriminator" and does not find an existing BFD session, the passive side MAY create a matching BFD session toward the active side, if permitted by local configuration.

It would then start sending the BFD Control packets and perform necessary procedure for bringing up, maintaining and tearing down the BFD session. If the BFD session fails to get established within certain specified time, or if an established BFD session goes down, the passive side would stop sending BFD Control packets and MAY delete the BFD session created until the BFD Control packets is initiated by the active side again.

When an Unsolicited BFD session goes down, an implementation MAY retain the session state for a period of time, which may be configurable. Retaining this state can be useful for operational purposes.

The "Passive role" may change to the "Active role" when a local client registers for the same BFD session, and from the "Active role" to the "Passive role" when there is no longer any locally registered client for the BFD session.

3. State Variables

This document defines a new state variable called Unsolicited Role.

 bfd.UnsolicitedRole

The operational mode of BFD interface when configured for unsolicited behaviour. Options can be either PASSIVE, ACTIVE or NULL (NULL - not initialized) for unsolicited BFD sessions. Default (not configured for unsolicited behaviour) MUST be set to NULL if present on the interface.
4. YANG Data Model

This section extends the YANG data model for BFD [I-D.ietf-bfd-yang] to cover the unsolicited BFD.

4.1. Unsolicited BFD Hierarchy

module: ietf-bfd-unsolicited
  augment /rt:routing/rt:control-plane-protocols
    /rt:control-plane-protocol/bfd:bfd/bfd-ip-sh:ip-sh:
      +++-rw unsolicited {bfd-unsol:unsolicited-params-global}? 
        +++-rw enable? boolean
        +++-rw local-multiplier? multiplier
        +++-rw (interval-config-type)?
          +++-:(tx-rx-intervals)
            | +++-rw desired-min-tx-interval? uint32
            | +++-rw required-min-rx-interval? uint32
            |+++:(single-interval) {single-minimum-interval}?
                |+++-rw min-interval? uint32
  augment /rt:routing/rt:control-plane-protocols
    /bfd-ip-sh:interfaces:
      +++-rw unsolicited {bfd-unsol:unsolicited-params-per-interface}? 
        +++-rw enable? boolean
        +++-rw local-multiplier? multiplier
        +++-rw (interval-config-type)?
          +++-:(tx-rx-intervals)
            | +++-rw desired-min-tx-interval? uint32
            | +++-rw required-min-rx-interval? uint32
            |+++:(single-interval) {single-minimum-interval}?
                |+++-rw min-interval? uint32
  augment /rt:routing/rt:control-plane-protocols
    /bfd-ip-sh:sessions/bfd-ip-sh:session:
      +++-ro unsolicited
        +++-ro role? bfd-unsol:unsolicited-role

4.2. Unsolicited BFD Module

<CODE BEGINS> file "ietf-bfd-unsolicited@2021-10-24.yang"
module ietf-bfd-unsolicited {
  yang-version 1.1;

prefix "bfd-unsol";

// RFC Ed.: replace occurrences of YYYY with actual RFC numbers
// and remove this note

import ietf-bfd-types {
    prefix "bfd-types";
    reference "RFC 9127: YANG Data Model for BFD";
}

import ietf-bfd {
    prefix "bfd";
    reference "RFC 9127: YANG Data Model for BFD";
}

import ietf-bfd-ip-sh {
    prefix "bfd-ip-sh";
    reference "RFC 9127: YANG Data Model for BFD";
}

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

organization "IETF BFD Working Group";

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

    Editors: Enke Chen (enchen@paloaltonetworks.com),
             Naiming Shen (naiming@zededa.com),
             Robert Raszuk (robert@raszuk.net),
             Reshad Rahman (reshad@yahoo.com)"

description
    "This module contains the YANG definition for BFD unsolicited
    as per RFC YYYY.

    Copyright (c) 2021 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 Simplified BSD License"
/* Feature definitions */

feature unsolicited-params-global {
  description "This feature indicates that the server supports global parameters for unsolicited sessions.";
}

feature unsolicited-params-per-interface {
  description "This feature indicates that the server supports per-interface parameters for unsolicited sessions.";
}

/* Type Definitions */

typedef unsolicited-role {
  type enumeration {
    enum unsolicited-active {
      description "Active role";
    }
    enum unsolicited-passive {
      description "Passive role";
    }
  }
  description "Unsolicited role";
}

/* Augments */

augment "/rt:routing/rt:control-plane-protocols/"
+ "rt:control-plane-protocol/bfd:bfd/bfd-ip-sh:ip-sh" {  
description  
"Augmentation for BFD unsolicited parameters";  
container unsolicited {  
  if-feature bfd-unsol:unsolicited-params-global;  
  description  
  "BFD unsolicited top level container";  
  leaf enable {  
    type boolean;  
    default false;  
    description  
    "Enable BFD unsolicited globally for IP single-hop.";  
  }  
  uses bfd-types:base-cfg-parms;  
}  
}  
}  

augment "/rt:routing/rt:control-plane-protocols/"  
+ "rt:control-plane-protocol/bfd:bfd/bfd-ip-sh:ip-sh/"  
+ "bfd-ip-sh:interfaces" {  
description  
"Augmentation for BFD unsolicited on IP single-hop interface";  
container unsolicited {  
  if-feature bfd-unsol:unsolicited-params-per-interface;  
  description  
  "BFD IP single-hop interface unsolicited top level container";  
  leaf enable {  
    type boolean;  
    default false;  
    description "Enable BFD unsolicited on this interface.";  
  }  
  uses bfd-types:base-cfg-parms;  
}  
}  
}  

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" {  
description  
"Augmentation for BFD unsolicited on IP single-hop session";  
container unsolicited {  
  config false;  
  description  
  "BFD IP single-hop session unsolicited top level container";  
  leaf role {  
    type bfd-unsol:unsolicited-role;  
    description "Role.";  
  }  
}  
}
5. IANA Considerations

This document registers the following namespace URI in the "IETF XML Registry" [RFC3688]:


Registrant Contact: The IESG.

XML: N/A; the requested URI is an XML namespace.

This document registers the following YANG module in the "YANG Module Names" registry [RFC6020]:

Name: ietf-bfd-unsolicited


Prefix: ietf-bfd-unsolicited

Reference: RFC YYYY

6. Acknowledgments

Authors would like to thank Acee Lindem, Greg Mirsky, Jeffrey Haas and Raj Chetan for their review and valuable input.

7. Security Considerations

7.1. BFD Protocol Security Considerations

The same security considerations and protection measures as those described in [RFC5880] and [RFC5881] normatively apply to this document. With "unsolicited BFD" there is potential risk for excessive resource usage by BFD from "unexpected" remote systems. To mitigate such risks, the following measures are mandatory:
* Limit the feature to specific interfaces, and to a single-hop BFD with "TTL=255" [RFC5082]. For numbered interfaces source address of an incoming BFD packet should belong to the subnet of the interface from which the BFD packet is received. For unnumbered interfaces the above check should be aligned with routing protocol addresses running on such pair of interfaces.
* Apply "access control" to allow BFD packets only from certain subnets or hosts.
* Deploy the feature only in certain "trustworthy" environment, e.g., at an IXP, or between a provider and its customers.
* Adjust BFD parameters as needed for the particular deployment and scale.
* Use BFD authentication.

7.2. YANG Module Security Considerations

The YANG module specified in this document defines a schema for data that is designed to be accessed via network management protocols such as NETCONF [RFC6241] or RESTCONF [RFC8040]. The lowest NETCONF layer is the secure transport layer, and the mandatory-to-implement secure transport is Secure Shell (SSH) [RFC6242]. The lowest RESTCONF layer is HTTPS, and the mandatory-to-implement secure transport is TLS [RFC5246].

The NETCONF access control model [RFC6536] 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 are a number of data nodes defined in this YANG module that are writable/creatable/deletable (i.e., config true, which is the default). These data nodes may be considered sensitive or vulnerable in some network environments. Write operations (e.g., edit-config) to these data nodes without proper protection can have a negative effect on network operations. These are the subtrees and data nodes and their sensitivity/vulnerability:

/routing/control-plane-protocols/control-plane-protocol/bfd/ip-sh/unsolicited:
* data node "enable" enables creation of unsolicited BFD IP single-hop sessions globally, i.e. on all interfaces. See Section 7.1.
* data nodes local-multiplier, desired-min-tx-interval, required-min-rx-interval and min-interval all impact the parameters of the unsolicited BFD IP single-hop sessions.

/routing/control-plane-protocols/control-plane-protocol/bfd/ip-sh/interfaces/interface/unsolicited:
* data node "enable" enables creation of unsolicited BFD IP single-hop sessions on a specific interface. See Section 7.1.
* data nodes local-multiplier, desired-min-tx-interval, required-min-rx-interval and min-interval all impact the parameters of the unsolicited BFD IP single-hop sessions on the interface.

Some of the readable data nodes in this YANG module may be considered sensitive or vulnerable in some network environments. It is thus important to control read access (e.g., via get, get-config, or notification) to these data nodes. These are the subtrees and data nodes and their sensitivity/vulnerability:

/routing/control-plane-protocols/control-plane-protocol/bfd/ip-sh /sessions/session/unsolicited: access to this information discloses the role of the local system in the creation of the unsolicited BFD session.

8. References

8.1. Normative References

[I-D.ietf-bfd-yang]


8.2. Informative References

[I-D.ietf-idr-rs-bfd]


Authors’ Addresses

Enke Chen
Palo Alto Networks
Email: enchen@paloaltonetworks.com

Naiming Shen
Zededa
Email: naiming@zededa.com

Robert Raszuk
NTT Network Innovations
940 Stewart Dr
Sunnyvale, CA 94085
United States of America
Email: robert@raszuk.net

Reshad Rahman
Email: reshad@yahoo.com