Network Working Group S. Aldrin
Internet-Draft Google, Inc
Intended status: Informational C. Pignataro
Expires: October 17, 2016 Cisco
G. Mirsky
Ericsson
N. Kumar
Cisco
April 15, 2016

Seamless Bidirectional Forwarding Detection (S-BFD) Use Cases
draft-ietf-bfd-seamless-use-case-05

Abstract

This document describes various use cases for a Seamless Bidirectional Forwarding Detection (S-BFD), and provides requirements such that protocol mechanisms allow for a simplified detection of forwarding failures.

These use cases support S-BFD, as a simplified mechanism to use Bidirectional Forwarding Detection (BFD) with large portions of negotiation aspects eliminated, accelerating the establishment of a BFD session. S-BFD benefits include quick provisioning as well as improved control and flexibility to network nodes initiating the path monitoring.

Table of Contents

1. Introduction

Bidirectional Forwarding Detection (BFD) is a lightweight protocol, as defined in [RFC5880], used to detect forwarding failures. Various protocols and applications rely on BFD as its clients for failure detection. Even though the protocol is lightweight and simple, there

are certain use cases where faster setting up of sessions and faster continuity check of the data forwarding paths is necessary. This document identifies these use cases and consequent requirements, such that enhancements and extensions result in a Seamless BFD (S-BFD) protocol.

BFD is a simple lightweight "Hello" protocol to detect data plane failures. With dynamic provisioning of forwarding paths on a large scale, establishing BFD sessions for each of those paths not only creates operational complexity, but also causes undesirable delay in establishing or deleting sessions. The existing session establishment mechanism of the BFD protocol has to be enhanced in order to minimize the time for the session to come up to validate the forwarding path.

This document specifically identifies various use cases and corresponding requirements in order to enhance BFD and other supporting protocols. Specifically, one key goal is removing the time delay (i.e., the "seam") between a network node wants to perform a continuity test and the node completes that continuity test. Consequently, "Seamless BFD" (S-BFD) has been chosen as the name for this mechanism.

While the identified requirements could meet various use cases, it is outside the scope of this document to identify all of the possible and necessary requirements. Solutions to the identified uses cases and protocol specific enhancements or proposals are outside the scope of this document as well. Protocol definitions to support these use cases can be found at [I-D.ietf-bfd-seamless-base] and [I-D.ietf-bfd-seamless-ip].

1.1. Terminology

The reader is expected to be familiar with the BFD [RFC5880], IP [RFC0791] [RFC2460], MPLS [RFC3031], and Segment Routing (SR) [I-D.ietf-spring-segment-routing] terminologies and protocol constructs.

1.2. 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 [RFC2119].

2. Introduction to Seamless BFD

BFD, as defined in [RFC5880], requires two network nodes to exchange locally allocated discriminators. These discriminators enable the identification of the sender and the receiver of BFD packets over the particular session. Subsequently, BFD performs proactive continuity monitoring of the forwarding path between the two. Several specifications describe BFD's multiple deployment uses:

[RFC5881] defines BFD over IPv4 and IPv6 for single IP hops

[RFC5883] defines BFD over multihop paths

[RFC5884] defines BFD for MPLS Label Switched Paths (LSPs)

[RFC5885] defines BFD for MPLS Pseudowires (PWs)

Currently, BFD is best suited to verify that two endpoints are mutually reachable or that an existing connection continues to be up and alive. In order for BFD to be able to initially verify that a connection is valid and that it connects the expected set of endpoints, it is necessary to provide each endpoint with the discriminators associated with the connection at each endpoint prior to initiating BFD sessions. The discriminators are used to verify that the connection is up and verifiable. Currently, the exchange of discriminators and the demultiplexing of the initial BFD packets is application dependent.

If this information is already known to the end-points of a potential BFD session, the initial handshake including an exchange of discriminators is unnecessary and it is possible for the endpoints to begin BFD messaging seamlessly. A key objective of the S-BFD use cases described in this document is to avoid needing to exchange the initial packets before the BFD session can be established, with the goal of getting to the established state more quickly; in other words, the initial exchange of discriminator information is an unnecessary extra step that may be avoided for these cases.

In a given scenario, an entity (such as an operator, or a centralized controller) determines a set of network entities to which BFD sessions might need to be established. In traditional BFD, each of those network entities chooses a BFD discriminator for each BFD session that the entity will participate in (see Section 6.3 of [RFC5880]). However, a key goal of a Seamless BFD is to provide operational simplification. In this context, for S-BFD, each of those network entities is assigned one or more BFD discriminators, and allowing those network entities to use one discriminator value for multiple sessions. Therefore, there may be only one or a few

discriminators assigned to a node. These network entities will create an S-BFD listener session instance that listens for incoming BFD control packets. When the mappings between specific network entities and their corresponding BFD discriminators are known to other network nodes belonging to the same administrative domain, then, without having received any BFD packet from a particular target, a network entity in this network is able to send a BFD control packet to the target's assigned discriminator in the Your Discriminator field. The target network node, upon reception of such BFD control packet, will transmit a response BFD control packet back to the sender.

3. Use Cases

As per the BFD protocol [RFC5880], BFD sessions are established using handshake mechanism prior to validating the forwarding path. This section outlines some use cases where the existing mechanism may not be able to satisfy the requirements identified. In addition, some of the use cases also stress the need for expedited BFD session establishment while preserving benefits of forwarding failure detection using existing BFD mechanics. Both these high-level goals result in the S-BFD use cases.

3.1. Unidirectional Forwarding Path Validation

Even though bidirectional verification of forwarding path is useful, there are scenarios where verification is only required in one direction between a pair of nodes. One such case is, when a static route uses BFD to validate reachability to the next-hop IP router. In this case, the static route is established from one network entity to another. The requirement in this case is only to validate the forwarding path for that statically established unidirectional path. Validation of the forwarding path in the direction of the target entity to the originating entity is not required, in this scenario. Many LSPs have the same unidirectional characteristics and unidirectional validation requirements. Such LSPs are common in Segment Routing and LDP based MPLS networks. A final example is when a unidirectional tunnel uses BFD to validate reachability of an egress node.

Additionally, there are operational implications to the unidirectional path validation. If the traditional BFD is to be used, the target network entity has to be provisioned as well as an initiator, even though the reverse path validation with the BFD session is not required. However, in the case of unidirectional BFD, there is no need for provisioning on the target network entity, only the source one.

In this use case, a BFD session could be established in a single direction. When the targeted network entity receives the packet, the Your Discriminator value in the packet instructs the network entity to process it, and send a response based on the source address of the packet. This does not necessitate the requirement for establishment of a bi-directional session, hence the two way handshake to exchange discriminators is not needed. The target node does not need to know the My Discriminator of the source node.

Thus, a requirement for BFD for this use case is to enable session establishment from source network entity to target network entity without the need to have a session (and state) for the reverse direction. Further, another requirement is that the BFD response from target back to sender can take any (in-band or out-of-band) path. The target network entity (for the BFD session), upon receipt of BFD packet, starts processing the BFD packet based on the discriminator received. The source network entity can therefore establish a unidirectional BFD session without the bidirectional handshake of discriminators for session establishment.

3.2. Validation of the Forwarding Path Prior to Switching Traffic

This use case is when BFD is used to verify reachability before sending traffic via a path/LSP. This comes with a cost, which is that traffic is prevented to use the path/LSP until BFD is able to validate the reachability, which could take seconds due to BFD session bring-up sequences [RFC5880], LSP ping bootstrapping [RFC5884], etc. This use case would be better supported by eliminating the need for the initial BFD session negotiation.

All it takes to be able to send BFD packets to a target, and the target properly demultiplexing these, is for the source network entities to know what the discriminator values to be used for the session. The same is the case for S-BFD: the three-way handshake mechanism is eliminated during the bootstrap of BFD sessions. However, this information is required at each entity to verify that BFD messages are being received from the expected end-points, hence the handshake mechanism serves no purpose. Elimination of the unnecessary handshake mechanism allows for faster reachability validation of BFD provisioned paths/LSPs.

In addition, it is expected that some MPLS technologies will require traffic engineered LSPs to be created dynamically, perhaps driven by external applications, as e.g. in Software Defined Networks (SDN). It will be desirable to perform BFD validation as soon as the LSPs are created, so as to use them.

In order to support this use case, an S-BFD session is established without the need for session negotiation and exchange of discriminators.

3.3. Centralized Traffic Engineering

Various technologies in the SDN domain that involve controller-based networks have evolved such that the intelligence, traditionally placed in a distributed and dynamic control plane, is separated from the networking entities themselves; instead, it resides in a (logically) centralized place. There are various controllers that perform the function in establishment of forwarding paths for the data flow. Traffic engineering (TE) is one important function, where the path of the traffic flow is engineered, depending upon various attributes and constraints of the traffic paths as well as the network state.

When the intelligence of the network resides in a centralized entity, the ability to manage and maintain the dynamic network and its multiple data paths and node reachability becomes a challenge. One way to ensure the forwarding paths are valid and working is done by validation using BFD. When traffic engineered tunnels are created, it is operationally critical to ensure that the forwarding paths are working, prior to switching the traffic onto the engineered tunnels. In the absence of distributed control plane protocols, it may be desirable to verify any arbitrary forwarding path in the network. With tunnels being engineered by a centralized entity, when the network state changes, traffic has to be switched with minimum latency and without black-holing of the data.

It is highly desirable in this centralized traffic engineering use case that the traditional BFD session establishment and validation of the forwarding path does not become a bottleneck. If the controller or other centralized entity is able to very rapidly verify the forwarding path of a traffic engineered tunnel, it could steer the traffic onto the traffic engineered tunnel very quickly thus minimizing adverse effect on a service. This is even more useful and necessary when the scale of the network and number of traffic engineered tunnels grows.

The cost associated with the time required for BFD session negotiation and establishment of BFD sessions to identify valid paths is very high when providing network redundancy is a critical issue.

3.4. BFD in Centralized Segment Routing

A monitoring technique of a Segment Routing network based on a centralized controller is described in [I-D.ietf-spring-oam-usecase]. Specific OAM requirements for Segment Routing are captured in [I-D.ietf-spring-sr-oam-requirement]. In validating this use case, one of the requirements is to ensure that the BFD packet's behavior is according to the monitoring specified for the segment, and that the packet is U-turned at the expected node. This criteria ensures the continuity check to the adjacent segment-id.

To support this use case, the operational requirement is for BFD, initiated from a centralized controller, to perform liveness detection for any given segment under its domain.

3.5. Efficient BFD Operation under Resource Constraints

When BFD sessions are being setup, torn down or modified (i.e., when parameters such as interval and multiplier are being modified), BFD requires additional packets other than scheduled packet transmissions to complete the negotiation procedures (i.e., P/F bits). There are scenarios where network resources are constrained: a node may require BFD to monitor very large number of paths, or BFD may need to operate in low powered and traffic sensitive networks; these include microwave, low powered nano-cells, and others. In these scenarios, it is desirable for BFD to slow down, speed up, stop, or resume at-will and with minimal number of additional BFD packets exchanged to modify the session or establish a new one.

The established BFD session parameters and attributes like transmission interval, receiver interval, etc., need to be modifiable without changing the state of the session.

3.6. BFD for Anycast Addresses

The BFD protocol requires two endpoints to host BFD sessions, both sending packets to each other. This BFD model does not fit well with anycast address monitoring, as BFD packets transmitted from a network node to an anycast address will reach only one of potentially many network nodes hosting the anycast address.

This use case verifies that a source node can send a packet to an anycast address, and that the target node to which the packet is delivered can send a response packet to the source node. Traditional BFD cannot fulfill this requirement, since it does not provide for a set of BFD agents to collectively form one endpoint of a BFD session. The concept of a Target Listener in S-BFD solves this requirement.

To support this use case, the BFD sender transmits BFD packets, which are received by any of the nodes hosting the anycast address to which the BFD packets being sent. The anycast target that receives the BFD packet, responds. This use case does not imply the BFD session establishment with every node hosting the anycast address. Consequently, in this any cast use case, target nodes that do not happen to receive any of the BFD packets do not need to maintain any state, and the source node does not need to maintain separate state for each target node.

3.7. BFD Fault Isolation

BFD for multihop paths [RFC5883] and BFD for MPLS LSPs [RFC5884] perform end-to-end validation, traversing multiple network nodes. BFD has been designed to declare failure upon lack of consecutive packet reception, which can be caused by a fault anywhere along these path. Fast failure detection allows for rapid fault detection and consequent rapid path recovery procedures. However, operators often have to follow up, manually or automatically, to attempt to identify and localize the fault that caused BFD sessions to fail (i.e., fault isolation). The usage of other tools to isolate the fault (e.g., traceroute) may cause the packets to traverse a different path through the network, if Equal-Cost Multipath (ECMP) is used. In addition, the longer it takes from BFD session failure to starting fault isolation, the more likely that the fault will not be able to be isolated (e.g., a fault can get corrected or routed around). If BFD had built-in fault isolation capability, fault isolation can get triggered at the earliest sign of fault detection. This embedded fault isolation will be more effective when those BFD fault isolation packets are load balanced in the same way as the BFD packets that were dropped, detecting the fault.

This use case describes S-BFD fault isolation capabilities using status indicating fields.

3.8. Multiple BFD Sessions to the Same Target Node

BFD is capable of providing very fast failure detection, as relevant network nodes continuously transmit BFD packets at the negotiated rate. If BFD packet transmission is interrupted, even for a very short period of time, BFD can declare a failure irrespective of path liveliness. It is possible, on a system where BFD is running, for certain events (intentionally or unintentionally) to cause a short interruption of BFD packet transmissions. With distributed architectures of BFD implementations, this case can be protected. In this case, the use case of an S-BFD node running multiple BFD sessions to a targets, with those sessions hosted on different system

modules (e.g., in different CPU instances). This can reduce BFD false failures, resulting in more stable network.

To support this use case, a mapping between the multiple discriminators on a single system, and the specific entity within the system is required.

3.9. An MPLS BFD Session Per ECMP Path

BFD for MPLS LSPs, defined in [RFC5884], describes procedures to run BFD as LSP in-band continuity check mechanism, through usage of MPLS echo request [RFC4379] to bootstrap the BFD session on the target (i.e., egress) node. Section 4 of [RFC5884] also describes a possibility of running multiple BFD sessions per alternative paths of LSP. [RFC7726] further clarified the procedures, both for ingress and egress nodes, of how to bootstrap, maintain, and remove multiple BFD sessions for the same <MPLS LSP, FEC> tuple. However, this mechanism still requires the use of MPLS LSP Ping for bootstrapping, round-trips for initialization, and keeping state at the receiver.

In the presence of ECMP within an MPLS LSP, it may be desirable to run in-band monitoring that exercises every path of this ECMP. Otherwise there will be scenarios where in-band BFD session remains up through one path but traffic is black-holing over another path. A BFD session per ECMP path of an LSP requires the definition of procedures that update [RFC5884] in terms of how to bootstrap and maintain the correct set of BFD sessions on the egress node. However, for traditional BFD, that requires the constant use of MPLS Echo Request messages to create and delete BFD sessions on the egress node, when ECMP paths and/or corresponding load balance hash keys change. If a BFD session over any paths of the LSP can be instantiated, stopped and resumed without requiring additional procedures of bootstrapping via an MPLS echo request message, it would greatly simplify both implementations and operations, and benefits network devices as less processing are required by them.

To support this requirement, multiple S-BFD sessions need to be established over different ECMP paths from the same source to target node.

4. Detailed Requirements for a Seamless BFD

REQ#1: A target network entity (for the S-BFD session), upon receipt of the S-BFD packet, MUST process the packet based on the discriminator received in the BFD packet. If the S-BFD context is found, the target network entity MUST be able to send a response.

REQ#2: The source network entity MUST be able to establish a unidirectional S-BFD session without the bidirectional handshake of discriminators for session establishment.

REQ#3: The S-BFD session MUST be able to be established without the need for exchange of discriminators in session negotiation.

REQ#4: In a Segment Routed network, S-BFD MUST be able to perform liveness detection initiated from a centralized controller for any given segment under its domain.

REQ#5: The established S-BFD session parameters and attributes, such as transmission interval, reception interval, etc., MUST be modifiable without changing the state of the session.

REQ#6: An S-BFD source network entity MUST be able to send S-BFD control packets to an anycast address which are received by any node hosting that address, and must be able to receive responses from any of these anycast nodes, without establishing a separate BFD session with every node hosing the anycast address.

REQ#7: S-BFD SHOULD support fault isolation capability, which MAY be triggered when a fault is encountered.

REQ#8: S-BFD SHOULD be able to establish multiple sessions between the same pair of source and target nodes. This requirement enables but does not guarantee the ability to monitor diverge paths in ECMP environments. It also provides resiliency in distributed router architectures. The mapping between BFD discriminators and particular entities (e.g., ECMP paths, or Line Cards) is out the scope of the S-BFD specification.

REQ#9: The S-BFD protocol MUST provide mechanisms for loop detection and prevention, protecting against malicious attacks attempting to create packet loops.

REQ#10: S-BFD MUST incorporate robust security protections against impersonators, malicions actors, and various attacks. The simple and accelerated establishment of an S-BFD session should not negatively affect security.

5. Security Considerations

This document details the use cases and identifies various associated requirements. Some of these requirements are security related. The use cases herein described do not expose a system to abuse or to additional security risks. The proposed new protocols, extensions, and enhancements for a Seamless BFD supporting these use cases and realizing these requirements will address the associated security considerations. A Seamless BFD should not have reduced security capabilities as compared to traditional BFD.

6. IANA Considerations

There are no IANA considerations introduced by this document.

7. Acknowledgements

The authors would like to thank Tobias Gondrom and Eric Gray, for their insightful and useful comments. The authors appreciate the thorough review and comments provided by Dale R. Worley.

8. Contributors

The following are key contributors to this document:

Manav Bhatia, Ionos Networks
Satoru Matsushima, Softbank
Glenn Hayden, ATT
Santosh P K
Mach Chen, Huawei
Nobo Akiya, Big Switch Networks

9. References

9.1. Normative References

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

[RFC5880] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD)", RFC 5880, DOI 10.17487/RFC5880, June 2010, <http://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, <http://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, <http://www.rfc-editor.org/info/rfc5883>.

[RFC5884] Aggarwal, R., Kompella, K., Nadeau, T., and G. Swallow, "Bidirectional Forwarding Detection (BFD) for MPLS Label Switched Paths (LSPs)", RFC 5884, DOI 10.17487/RFC5884, June 2010, <http://www.rfc-editor.org/info/rfc5884>.

[RFC5885] Nadeau, T., Ed. and C. Pignataro, Ed., "Bidirectional Forwarding Detection (BFD) for the Pseudowire Virtual Circuit Connectivity Verification (VCCV)", RFC 5885, DOI 10.17487/RFC5885, June 2010, <http://www.rfc-editor.org/info/rfc5885>.

9.2. Informative References

[I-D.ietf-bfd-seamless-base] Akiya, N., Pignataro, C., Ward, D., Bhatia, M., and J. Networks, "Seamless Bidirectional Forwarding Detection (S-BFD)", draft-ietf-bfd-seamless-base-09 (work in progress), April 2016.

[I-D.ietf-bfd-seamless-ip] Akiya, N., Pignataro, C., and D. Ward, "Seamless Bidirectional Forwarding Detection (S-BFD) for IPv4, IPv6 and MPLS", draft-ietf-bfd-seamless-ip-04 (work in progress), April 2016.

[I-D.ietf-spring-oam-usecase] Geib, R., Filsfils, C., Pignataro, C., and N. Kumar, "A scalable and topology aware MPLS data plane monitoring system", draft-ietf-spring-oam-usecase-02 (work in progress), April 2016.

[I-D.ietf-spring-segment-routing] Filsfils, C., Previdi, S., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing Architecture", draft-ietf-spring-segment-routing-07 (work in progress), December 2015.

[I-D.ietf-spring-sr-oam-requirement]
Kumar, N., Pignataro, C., Akiya, N., Geib, R., Mirsky, G., and S. Litkowski, "OAM Requirements for Segment Routing Network", draft-ietf-spring-sr-oam-requirement-01 (work in progress), December 2015.

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

[RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, DOI 10.17487/RFC2460, December 1998, <http://www.rfc-editor.org/info/rfc2460>.

[RFC3031] Rosen, E., Viswanathan, A., and R. Callon, "Multiprotocol Label Switching Architecture", RFC 3031, DOI 10.17487/RFC3031, January 2001, <http://www.rfc-editor.org/info/rfc3031>.

[RFC4379] Kompella, K. and G. Swallow, "Detecting Multi-Protocol Label Switched (MPLS) Data Plane Failures", RFC 4379, DOI 10.17487/RFC4379, February 2006, <http://www.rfc-editor.org/info/rfc4379>.

[RFC7726] Govindan, V., Rajaraman, K., Mirsky, G., Akiya, N., and S. Aldrin, "Clarifying Procedures for Establishing BFD Sessions for MPLS Label Switched Paths (LSPs)", RFC 7726, DOI 10.17487/RFC7726, January 2016, <http://www.rfc-editor.org/info/rfc7726>.

Authors' Addresses

Sam Aldrin
Google, Inc

Email: aldrin.ietf@gmail.com

Carlos Pignataro
Cisco Systems, Inc.

Email: cpignata@cisco.com

Greg Mirsky
Ericsson

Email: gregory.mirsky@ericsson.com

Nagendra Kumar
Cisco Systems, Inc.

Email: naikumar@cisco.com