Network Working Group
Internet-Draft
Intended status: Informational
Expires: September 22, 2016

S. Aldrin
Google, Inc
M. Bhatia
Ionos Networks
S. Matsushima
Softbank
G. Mirsky
Ericsson
N. Kumar
Cisco

March 21, 2016

Seamless Bidirectional Forwarding Detection (BFD) Use Case
draft-ietf-bfd-seamless-use-case-04

Abstract

This document provides various use cases for Bidirectional Forwarding Detection (BFD) and various requirements such that extensions could be developed to allow for simplified detection of forwarding failures.

Status of This Memo

Copyright Notice

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 for failure detection. Even though the protocol is simple, there are certain use cases, where faster setting up of sessions and continuity check of the data forwarding paths is necessary. This document identifies various use cases and requirements related to those, such that necessary enhancements could be made to 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 creates complexity, not only from an operations point of view, but also in

terms of the speed at which these sessions could be established or deleted. 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. 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.

1.1. Terminology

The reader is expected to be familiar with the BFD, IP, MPLS and Segment Routing (SR) [I-D.ietf-spring-segment-routing] terminology and protocol constructs. This section identifies only the new terminology introduced.

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. The discriminator enables identification of the sender and receiver of BFD packets of the particular session and perform proactive continuity monitoring of the forwarding path between the two. [RFC5881] defines single hop BFD whereas [RFC5883] defines multi-hop BFD, [RFC5884] BFD for MPLS LSPs, and [RFC5885] - BFD for PWs.

Currently, BFD is best suited to verify that two end points are 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 end points, it is necessary to provide the node information associated with the connection at each end point prior to initiating BFD sessions, such that this information can be used to verify that the connection is up and verifiable.

If this information is already known to the end-points of a potential BFD session, the initial handshake including an exchange of this

node-specific information is unnecessary and it is possible for the end points to begin BFD messaging seamlessly. In fact, the initial exchange of discriminator information is an unnecessary extra step that may be avoided for these cases.

In a given scenario, where an entity (such as an operator, or centralized controller) determines a set of network entities to which BFD sessions might need to be established. Each of those network entities is assigned a BFD discriminator, to establish a BFD session. These network entities will create a BFD session instance that listens for incoming BFD control packets. Mappings between selected network entities and corresponding BFD discriminators are known to other network nodes belonging in the same network by some means. A network entity in this network is then able to send a BFD control packet to a particular target with the corresponding BFD discriminator. 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 specifications.

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 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 networks. Another example is when a unidirectional tunnel uses BFD to validate reachability of an egress node.

If the traditional BFD is to be used, the target network entity has to be provisioned as well, even though the reverse path validation with BFD session is not required. However, in the case of

unidirectional BFD, there is no need for provisioning on the target network entity . Once the mechanism within the BFD protocol is in place, session could be established in a single direction. When the targeted network entity receives the packet, it knows that BFD packet, based on the discriminator and processes it. This does not necessitates the requirement for establishment of a bi-directional session, hence the two way handshake to exchange discriminators is not needed.

Thus the 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 in the reverse direction. This requires to ensure that the target network entity (for the BFD session), upon receipt of BFD packet, MUST start processing for the discriminator received in the BFD packet. The source network entity MUST be able to establish a unidirectional BFD session without the bidirectional handshake of discriminators for session establishment.

3.2. Validation of forwarding path prior to traffic switching

BFD provides data delivery confidence when reachability validation is performed prior to traffic utilizing specific paths/LSPs. However this comes with a cost, where, traffic is prevented to use such paths/LSPs 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 could be well supported by eliminating the need for session negotiation and discriminator exchanges in order to establish the BFD session.

All it takes is for the network entities to know what the discriminator values to be used for the session. The same is the case for S-BFD, i.e., the three-way handshake mechanism is eliminated during 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, e.g. in Software Defined Networks (SDN). It will be desirable to perform BFD validation as soon as the LSP?s are created, in order to use them.

In order to support this use case, the BFD session MUST be able to be 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 where intelligence, traditionally placed in a distributed and dynamic control plane, is separated from the networking entities along the data path, instead resides in a logically centralized place. There are various controllers that perform this exact function in establishment of forwarding paths for the data flow. Traffic engineering is one important function, where 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, ability to manage and maintain the dynamic network becomes a challenge. One way to ensure the forwarding paths are valid, and working, is done by validation of the network 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 control plane protocols, it may be desirable to verify, not only the forwarding path but also of any arbitrary 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.

Traditional BFD session establishment and validation of the forwarding path must not become a bottleneck in the case of centralized traffic engineering. If the controller or other centralized entity is able to instantly verify a forwarding path of the TE tunnel , it could steer the traffic onto the traffic engineered tunnel very quickly thus minimizing adverse effect on a service. This is especially useful and needed when the scale of the network and number of TE tunnels is very high.

The cost associated with BFD session negotiation and establishment of BFD sessions to identify valid paths is very high and providing network redundancy becomes 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]. Various OAM requirements for Segment Routing were captured in [I-D.ietf-spring-sr-oam-requirement]. In validating this use case, one of the requirements is to ensure the BFD packet's behavior is according to the requirement and monitoring of the segment, where the packet is U-turned at the expected node. One of the criterion is to ensure the continuity check to the adjacent segment-id.

To support this use case, BFD MUST be able to perform liveness detection initated from centralized controller 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. parameters ? such as interval, multiplier, etc 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, i.e. microwave, low powered nano-cells, etc. In these scenarios, it is desirable for BFD to slow down, speed up, stop or resume at will witho minimal additional BFD packets exchanged to establish a new or modified session.

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

3.6. BFD for Anycast Address

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.

To support this use case, the BFD MUST be able to send packets in order to be received by any of nodes hosting anycast address to which the BFD packets being sent and to respond. This requirement does not require BFD session establishment with every node hosting the anycast address.

3.7. BFD Fault Isolation

BFD multi-hop [RFC5883]and BFD MPLS [RFC5884] traverse 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 the path. Fast failure detection allows for 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. Usage of other tools to isolate the fault may cause the packets to traverse a different path through the network (e.g. if ECMP is used). In addition, the longer it takes from BFD session failure to fault isolation attempt, more

likely that the fault cannot 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 and such packets will get load balanced in very similar way, if not the same, as BFD packets that went missing.

To support this requirement, BFD SHOULD support fault isolation capability using status indicating fields, when encountered.

3.8. Multiple BFD Sessions to Same Target

BFD is capable of providing very fast failure detection, as relevant network nodes continuously transmit BFD packets at negotiated rate. If BFD packet transmission is interrupted, even for a very short period of time, that can result in BFD to declare 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 can be protected, if a node was to run multiple BFD sessions to targets, hosted on different parts of the system (ex: different CPU instances). This can reduce BFD false failures, resulting in more stable network.

3.9. MPLS BFD Session Per ECMP Path

BFD for MPLS, 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 egress node. Section 4 of [RFC5884] also describes a possibility of running multiple BFD sessions per alternative paths of LSP. However, details on how to bootstrap and maintain correct set of BFD sessions on the egress node is absent.

When an LSP has ECMP segment, it may be desirable to run in-band monitoring that exercises every path of ECMP. Otherwise there will be scenarios where in-band BFD session remains up through one path but traffic is black-holing over another path. BFD session per ECMP path of LSP requires definition of procedures that update [RFC5884] in terms of how to bootstrap and maintain correct set of BFD sessions on the egress node. However, that may require 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 MPLS echo request, it would simplify implementations and operations, and benefits network devices as less processing are required by them.

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

4. Detailed Requirements

REQ#1- A target network entity (for the BFD session), upon receipt of BFD packet, MUST start processing for the discriminator received in the BFD packet.

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

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

REQ#4 - BFD MUST be able to perform liveness detection initated from centralized controller for any given segment under its domain.

REQ#5 - The established BFD session parameters and attributes like transmission interval, receiver interval, etc., MUST be modifiable without changing the state of the session.

REQ#6 - The BFD MUST be able to send and receive response to control packets addressed to an anycast address to be received by any of nodes hosting that address. This requirement does not require BFD session establishment with every node hosting the anycast address.

REQ#7 - BFD SHOULD support fault isolation capability and to indicate the same, when fault is encountered.

REQ#8 - BFD MUST be able to establish multiple sessions between the same pair of source and target nodes. This requirement enables but does not guarantee ability to monitor diverge paths in ECMP environment. The mapping between BFD session and particular ECMP path is out the scope of BFD specification.

5. Security Considerations

This document details the use cases and identifies various requirements for the same. As this document do not propose any new protocol or changes to the existing ones, no new security considerations have been identified with this draft.

6. IANA Considerations

There are no IANA considerations introduced by this draft

7. Contributors

Carlos Pignataro

Cisco Systems

Email: cpignata@cisco.com

Glenn Hayden

ATT

Email: gh1691@att.com

Santosh P K

Juniper

Email: santoshpk@juniper.net

Mach Chen

Huawei

Email: mach.chen@huawei.com

Nobo Akiya

Cisco Systems

Email: nobo@cisco.com

8. Acknowledgements

The authors would like to thank Eric Gray for his useful comments.

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>.

[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>.

[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-spring-oam-usecase]
Geib, R., Filsfils, C., Pignataro, C., and N. Kumar, "Use Case for a Scalable and Topology-Aware Segment Routing MPLS Data Plane Monitoring System", draft-ietf-spring-oam-usecase-01 (work in progress), October 2015.

[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.

Authors' Addresses

Sam Aldrin
Google, Inc
1600 Amphitheatre Parkway
Mountain View, CA

Email: aldrin.ietf@gmail.com

Manav Bhatia
Ionos Networks

Email: manav@ionosnetworks.com

Satoru Matsushima
Softbank

Email: satoru.matsushima@g.softbank.co.jp

Greg Mirsky
Ericsson

Email: gregory.mirsky@ericsson.com

Nagendra Kumar
Cisco

Email: naikumar@cisco.com