INTERNET-DRAFT
Intended Status: Informational
Expires: December 13, 2014

Sam Aldrin
(Huawei)
Manav Bhatia
(Ionos)
Greg Mirsky
(Ericsson)
Nagendra Kumar
(Cisco)
Satoru Matsushima
(Softbank)

June 11, 2014

Seamless Bidirectional Forwarding Detection (BFD) Use Case
draft-ietf-bfd-seamless-use-case-00

Abstract

This document provides various use cases for Bidirectional Forwarding Detection (BFD) such that simplified solution and extensions could be developed for detecting forwarding failures.

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 and lightweight, there are certain use cases, where a much faster setting up of sessions and continuity check of the data forwarding paths is necessary. This document identifies those use cases such that necessary enhancements could be made to BFD protocol to meet those requirements.

There are various ways to detecting faults and BFD protocol was designed to be a lightweight "Hello" protocol to detect data plane failures. With dynamic provisioning of forwarding paths at a large scale, establishing BFD sessions for each of those paths creates complexity, not only from operations point of view, but also the speed at which these sessions could be established or deleted. The existing session establishment mechanism of the BFD protocol need to be enhanced in order to minimize the time for the session to come up and validate the forwarding path.

This document specifically identifies those cases where certain requirements could be derived to be used as reference, so that, protocol enhancements could be developed to address them. Whilst the use cases could be used as reference for certain requirements, it is outside the scope of this document to identify all of the requirements for all possible enhancements. Specific solutions and enhancement 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 SR terminology and protocol constructs. This section identifies only the new terminology introduced.

2. Introduction to Seamless BFD

BFD as defined in standard [RFC5880] requires two network nodes, as part of handshake, exchange discriminators. This will enable the sender and receiver of BFD packets of a session to be identified and check the continuity of the forwarding path. [RFC5881] defines single hop BFD whereas [RFC5883] and [RFC5884] defines multi-hop BFD.

In order to establish BFD sessions between network entities and seamlessly be able to have the session up and running, BFD protocol should be capable of doing that. These sessions have to be established a priori to traffic flow and ensure the forwarding path is available and connectivity is present. With handshake mechanism

within BFD protocol, establishing sessions at a rapid rate and ensuring the validity or existence of working forwarding path, prior to the session being up and running, becomes complex and time consuming. In order to achieve seamless BFD sessions, it requires a mechanism where the ability to specify the discriminators and the ability to respond to the BFD control packets by the network node, should already be negotiated ahead of the session becoming active. Seamless BFD by definition will be able to provide those mechanisms within the BFD protocol in order to meet the requirements and establish BFD sessions seamlessly, with minimal overhead, in order to detect forwarding failures.

As an example of how Seamless BFD (S-BFD) works, a set of network entities are first identified, to which BFD sessions have to be established. Each of those network nodes, will be assigned a special BFD discriminator, to establish a BFD session. These network nodes will also create a BFD session instance that listens for incoming BFD control packets. Mappings between selected network entities and corresponding special BFD discriminators are known to other network nodes belonging in the same network. A network node in such network is then able to send a BFD control packet to a particular target with corresponding special 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 of the use cases where the existing mechanism may not be able to satisfy the requirements. In addition, some of the use cases will also be identifying the need for expedited BFD session establishment with 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 when only one side of the BFD, not both, is interested in verifying continuity of the data plane 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. Validating the reverse direction is not required in this case. Many of these network scenarios are being proposed as part of segment routing [TBD]. Another example is when a unidirectional tunnel uses BFD to validate reachability to the 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. But with unidirectional BFD, the need to provision on the target network entity is not needed. Once the mechanism within the BFD protocol is in place, where the source network entity knows the target network entity's discriminator, it starts the session right away. When the targeted network entity receives the packet, it knows that BFD packet, based on the discriminator and processes it. That do not require to have a bi-directional session establishment, hence the two way handshake to exchange discriminators is not needed as well.

The primary requirement in this use case is to enable session establishment from source network entity to target network entity. This translates to, the target network entity for the BFD session, upon receiving the BFD packet, should start processing for the discriminator received. This will enable the source network entity to establish a unidirectional BFD session without 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 does not require to have sequences for session negotiation and discriminator exchanges in order to establish the BFD session.

When these sequences for handshake are eliminated, the network entities need to know what the discriminator values to be used for the session. The same is the case for S-BFD, i.e., when the three-way handshake mechanism is eliminated during bootstrap of BFD sessions. Due to this faster reachability validation of BFD provisioned paths/LSPs could be achieved. In addition, it is expected that some MPLS technologies will require traffic engineered LSPs to get created dynamically, driven by external applications, e.g. in Software Defined Networks (SDN). It would be desirable to perform BFD validation very quickly to allow applications to utilize dynamically created LSPs in timely manner.

3.3. Centralized Traffic Engineering

Various technologies in the SDN domain have evolved which involves

controller based networks, where the intelligence, traditionally placed in the distributed and dynamic control plane, is separated from the data plane and resides in a logically centralized place. There are various controllers which perform this exact function in establishing forwarding paths for the data flow. Traffic engineering is one important function, where the traffic is engineered depending upon various attributes of the traffic as well as the network state.

When the intelligence of the network resides in the 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 to establish BFD sessions within the network. When traffic engineering 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 is not only the desire to verify the forwarding path but also an arbitrary path in the network. With tunnels being engineered from the centralized entity, when the network state changes, traffic has to be switched without much latency and black holing of the data.

Traditional BFD session establishment and validation of the forwarding path must not become 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 too high. Session negotiation and establishment of BFD sessions to identify valid paths is way to high in terms of time and providing network redundancy becomes a critical issue.

3.4. BFD in Centralized Segment Routing

Centralized controller based Segment Routing network monitoring technique, is described in [I-D.geib-spring-oam-usecase]. 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.

3.5. BFD to Efficiently Operate under Resource Constraints

When BFD sessions are being setup, torn down or parameters (i.e. interval, multiplier, etc) are being modified, BFD protocol requires additional packets outside of 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 without requiring additional BFD packets to be exchanged.

3.6. BFD for Anycast Address

BFD protocol requires the 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.

3.7. BFD Fault Isolation

BFD multi-hop and BFD MPLS traverse multiple network nodes. BFD has been designed to declare failure upon lack of consecutive packet reception, which can be caused by any fault anywhere along the path. Fast failure detection provides great benefits, as it can trigger recovery procedures rapidly. However, operators often have to follow up, manually or automatically, to attempt to identify and localize the fault which caused the BFD sessions to fail. Usage of other tools to isolate the fault may cause the packets to traverse differently throughout the network (i.e. ECMP). In addition, longer it takes from BFD session failure to fault isolation attempt, more likely that fault cannot be isolated, i.e. 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 which went missing.

3.8. Multiple BFD Sessions to Same Target

BFD is capable of providing very fast failure detection, as relevant network nodes continuously transmitting 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. One way to achieve BFD session per ECMP path of LSP is to define 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.

4. Security Considerations

There are no new security considerations introduced by this draft.

5. IANA Considerations

There are no new IANA considerations introduced by this draft

6. References

6.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC4379] Kompella, K. and G. Swallow, "Detecting Multi-Protocol Label Switched (MPLS) Data Plane Failures", RFC 4379, February 2006.

[RFC5880] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD)", RFC5880, June 2010.

[RFC5881] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD)", RFC5881, June 2010.

[RFC5883] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD) for Multihop Paths", RFC5883, June 2010.

[RFC5884] Aggarwal, R., Kompella, K., Nadeau, T., and G. Swallow, "Bidirectional Forwarding Detection (BFD) for MPLS Label Switched Paths (LSPs)", RFC5884, June 2010.

7. Authors' Addresses

Sam Aldrin
Huawei Technologies
2330 Central Expressway
Santa Clara, CA 95051

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

8. 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