Authors: M. Chen   C. Li   W. Jiang   Y. Liu
         Huawei   Huawei   China Mobile   China Mobile
         X. Chen
         Huawei

Segment Routing for Unaffiliated BFD Echo Function

Abstract

This document describes how to leverage Segment Routing (SR) to
ensure that the Unaffiliated BFD (U-BFD) Echo packets must reach the
remote system before being looped back to the local system. This
enables that U-BFD works not only for one hop scenario but for
multiple hops scenario as well.

In addition, this document also defines a way to explicitly specify
the loop back path of the U-BFD Echo packets. This is useful in the
case where the forward and reverse path of the Echo packets are
required to follow the same path.

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] [RFC8174] when, and only when, they appear in all
capitals, as shown here.

Status of This Memo

Table of Contents

1. Introduction

   BFD Echo function was originally defined in [RFC5880] and [RFC5881],
   where the remote system is required to loop the BFD Echo packets
   back to the local system. To support BFD Echo Function, some
   negotiations between the local system and remote system are needed,
   and both the local and remote system need to maintain the BFD
   session state.

   Unaffiliated BFD Echo Function (U-BFD) is defined in [I-D.ietf-bfd-
   unaffiliated-echo]. Where the destination IP address of the BFD Echo
   packets is set to one of the IP addresses of the local system.
   Therefore, the Echo packets can be automatically looped back
   (through normal IP forwarding) by the remote system to the local
   system. With U-BFD, the remote system does not need to support any
   BFD related functions and maintain any session states. This further
   simplifies the BFD Echo Function process at the remote system hence
   greatly increases scalability.

But, the U-BFD works when there is only one hop between the local
system and remote system. Otherwise, the Echo packets will be
prematurely looped back by an intermediate node, and the Echo
packets will not reach the remote system. This may result in false
negative issue. Take the following figure (Figure 1) as an example,
if the U-BFD is expected to monitor the path between node A and node
C, node A (as the local system) sets the destination IP to itself
and sends the Echo packets to node B. Since node B has the route to
node A, the Echo packets will be directly forwarded back to node A.
If there is a failure on the path between node B and node C,
obviously, the U-BFD session cannot detect it.

```
    +-+           +-+           +-+
    |A|--------|B|---------|C|
    +-+           +-+           +-+
```
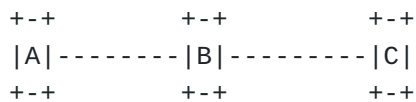
    Figure 1, Multi-hop Scenario

In addition, in some scenarios, for example, mobile backhaul
network, where the forward and reverse direction of a path are
required to along the same path. When apply BFD in mobile backhaul
network, it also expects that the BFD control packets in both
directions follow the same path, otherwise, it may result in false
positive issue. Take the following figure (Figure 2) as an example,
there are two paths (A-B-C, A-D-C) between node A and node C.
Assuming that it expects to monitor the path A-B-C by using BFD,
where node A is the local system and node C is the remote system. If
node C chooses path C-D-A to send the Echo packets, when a failure
occurs on path C-D-A, node A (the local system) may not receive the
BFD packets and consider that path A-B-C is failed. But actually
path A-B-C is working.
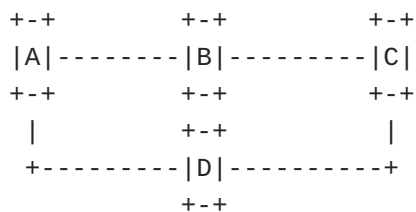
```
    +-+           +-+           +-+
    |A|--------|B|---------|C|
    +-+           +-+           +-+
     |            +-+            |
    +---------|D|----------+
                  +-+
```
 Figure 2, Multi-hop, Multi-path Scenario

To solve the above issues, this document describes how to leverage
the Segment Routing (SR) [RFC8402] to ensure that the U-BFD Echo
packets must reach the remote system before being looped back. This
enables that U-BFD Echo Function works not only for one hop scenario
but for multiple hops scenario. In addition, by using SR policy [I-
D.ietf-spring-segment-routing-policy], the loop back path of the
Echo packets can be specified as required. This is useful in the
case where the forward and loop back path of the Echo packets are
required to follow the same path.

2. Tunnel U-BFD Packets to Remote System

When using U-BFD to monitor a path, the U-BFD echo packets are
constructed as specified in [I-D.ietf-bfd-unaffiliated-echo], then
the U-BFD echo packets are encapsulated in an SR tunnel and sent to
the remote system. It needs to make sure that the SR tunnel and the
path being monitored follow the same path and share the same fate,
otherwise the UP/Down state can not reflect the health of the path
being monitored. This can be satisfied if the path itself is an SR
path, where the U-BFD echo packets, just as other data traffic, are
transmitted over the SR tunnel to the remote system. When the packet
arrives the remote system, the SR encapsulation (e.g, MPLS Label
Stack, or IPv6 Header with/without SRH) is removed, the inner IP
header is exposed. Since the destination IP address of the inner
header is one of the routable IP addresses of the local system, the
echo packet will be forwarded back to the local system via IP
routing.

If the path is an pure IP path, SR over IP [RFC8663] or SRv6 Best
Effort (BE) can be used to tunnel the U-BFD echo packets to the
remote system. Once the packet reaches the remote system, the remote
system decapsulates the echo packet and forwards it back to the
local system based on the inner header.

3. Specify Reverse Path of U-BFD Echo Packets

In some cases, for example, mobile backhaul network, bidirectional
paths are often used. And in general, the forward and reverse
direction of the bidirectional path are required to follow the same
path hence to share the same fate. Therefore, when apply U-BFD to
monitor such a bidirectional path, the forward and reserve path the
U-BFD echo need to follow the same path as well.

In the case of SR, [I-D.ietf-spring-segment-routing-policy] is
normally used to implement bidirectional path. To ensure that the U-
BFD Echo packets can reach the remote system before looping back to
the local system, the SR policy MUST have a candidate path that is
associated with a Segment-List, and the Segment-List MUST include a
SID that identifies the remote system. That means the U-BFD Echo
packets will be tunneled by the SR policy to the remote system, and
then looped back.

To specify the loopback path, a series of SIDs or a Binding SID
(BSID) that is associated with the loopback path MUST be added to
the SID stack of the U-BFD Echo packets. This can be done through
the following ways:

Given the topology in Figure 2, the SR policy can be modeled as
below:

1. Not explicitly specify the reserve path:

       SR policy POL1 <headend = A, color = 1, endpoint = C>
         Candidate-path CP1 <protocol-origin = 20, originator =
                             100:1.1.1.1, discriminator = 1>
           Preference 200
           Weight W1, SID-List <B,C>

   The U-BFD Echo packets are transmitted to the remote system
   according the SR policy. When the Echo packets reach the remote
   system, they will be decapsulated and then forwarded back to
   the local system according to inner IP header.

2. Specify the reverse path by carrying a Reverse Binding Segment
   Identifier (R-BSID):

       SR policy POL2 <headend = A, color = 1, endpoint = C>
         Candidate-path CP1 <protocol-origin = 20, originator =
                             100:1.1.1.1, discriminator = 2>
           Preference 200
           Weight W1, SID-List <B, C>, L-BSID, R-BSID

   Two new attributes, which are referred to as Local BSID (L-
   BSID) and Remote BSID (R-BSID), are introduced to a candidate
   path. Here, the L-BSID or R-BSID is a BSID that correlates to a
   specific SID List rather a candidate path. The L-BSID is a
   local BSID on the headend, in the above example, it identifies
   the SID-List <B, C>. The R-BSID identifies another
   corresponding SID list <B, A> that is deployed on the endpoint
   (Node C) and has the same path and share the same fate with
   SID-list <B, C>. SID List <B, C> and SID List <B, A> form a
   bidirectional SR path. With this, a SID stack <B, C, R-BSID>
   will be attached to the U-BFD echo packets, the SID B and C
   will take the echo packets to the remote system, the R-BSID
   will bring the echo packets back to the local system.

3. Specify the reverse path by explicitly carrying the SID list of
   the reverse path:

       SR policy POL2 <headend = A, color = 1, endpoint = A>
         Candidate-path CP1 <protocol-origin = 20, originator =
                             100:1.1.1.1, discriminator = 3>
           Preference 200
           Weight W1, SID-List <B, C>, Reverse SID-List <B, A>

A new attribute, Reverse SID-List, is introduced to a candidate
path, it corresponds to a SID-List of the candidate path. In
the above example, the SID-List <B, C> and Reverse SID-List <B,
A> form a bidirectional path. With this, a SID stack <B, C, B,
A> will be attached to the U-BFD echo packets, th SID B and C
will take the echo packets to the remote system, and the SID B
and A will bring the back to the local system.

4. Binding SID for Segment-List

With the current SR policy, a BSID corresponds to a candidate path
that may have multiple SID Lists. In the case of bidirectional SR
path, the forward or reverse path corresponds to a specific SID
List. In order to identify a SID List with a SID, the
straightforward idea is to define a BSID to for SID list. A
bidirectional SR path correlates to two SID Lists: the forward SID
List and the reverse SID List. Therefore, two BSIDs (L-BSID and R-
BSID) need to be defined for a bidirectional SR path to identify the
corresponding SID list.

An information model of a SR policy with the L-BSID and R-BSID is as
follows:

```
SR policy POL1 <headend = H1, color = 1, endpoint = E1>
  Candidate-path CP1 <protocol-origin = 20, originator =
                     100:1.1.1.1, discriminator = 1>
    Preference 100
    Weight W1, SID-List1 <SID11...SID1i>, L-BSID-1, R-BSID-1

  Candidate-path CP2 <protocol-origin = 20, originator =
                     100:1.1.1.1, discriminator = 2>
    Preference 200
    Weight W2, SID-List2 <SID21...SID2j>, L-BSID-2, R-BSID-2
```

The POL1 has two candidate paths, CP1 and CP2. Each has a SID-List,
a Local BSID (L-BSID) and a Reverse BSID (R-BSID). The L-BSID
corresponds to the SID List (e.g., L-BSID-1 corresponds to SID-List
<SID11...SID1i>). The R-BSID corresponds a SID List of a candidate
path of a policy that is deployed on the endpoint (E1), as
following. Assume that the SID-List1 of POL1 and the SID-List1 of
POL2 form a bidirectional SR path. For policy POL1, the R-BSID-1 is
the same as the L-BSID-1' of policy POL2; and the R-BSID-1' of POL2
is the same as the L-BSID-1 of policy POL1.

```
SR policy POL2 <headend = E1, color = 1, endpoint = H1>
  Candidate-path CP1 <protocol-origin = 20, originator =
                      200:1.1.1.1, discriminator = 1>
    Preference 100
    Weight W1, SID-List1 <SID11...SID1i>, L-BSID-1', R-BSID-1'

  Candidate-path CP2 <protocol-origin = 20, originator =
                      200:1.1.1.1, discriminator = 2>
    Preference 200
    Weight W2, SID-List2 <SID21...SID2j>, L-BSID-2', R-BSID-2'
```

Therefore, to deploy such a policy, it needs to know the R-BSID of
the corresponding reverse SID List. It assumes that a controller
will learn the L-BSID of each SID list and is responsible for the
configuration of SR policy on both the headend and endpoint of the
SR policies.

The corresponding BGP or PECP extensions in support of SR policy
with BSID for SID List will be defined in future version.

5. IANA Considerations

This document makes no request of IANA.

6. Security Considerations

This document does not introduce additional security requirements
and mechanisms other than the ones described in [I-D.ietf-bfd-
unaffiliated-echo] and [I-D.ietf-spring-segment-routing-policy].

7. Acknowledgements

Many thanks for the Changwang Lin for his valuable comments.

8. Contributors

The following people have substantially contributed to this
document:

Pingwei Fan
Huawei

EMail: fanpingwei@huawei.com

Sheng Fang
Huawei

EMail: fangsheng@huawei.com

9. References

9.1. Normative References

[I-D.ietf-bfd-unaffiliated-echo] Cheng, W., Wang, R., Min, X.,
           Rahman, R., and R. C. Boddireddy, "Unaffiliated BFD
           Echo", Work in Progress, Internet-Draft, draft-ietf-bfd-
           unaffiliated-echo-05, 1 August 2022, <https://
           www.ietf.org/archive/id/draft-ietf-bfd-unaffiliated-
           echo-05.txt>.

[I-D.ietf-spring-segment-routing-policy]
           Filsfils, C., Talaulikar, K., Voyer, D., Bogdanov, A.,
           and P. Mattes, "Segment Routing Policy Architecture",
           Work in Progress, Internet-Draft, draft-ietf-spring-
           segment-routing-policy-22, 22 March 2022, <https://
           www.ietf.org/archive/id/draft-ietf-spring-segment-
           routing-policy-22.txt>.

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

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

9.2. Informative References

[RFC5880] Katz, D. and D. Ward, "Bidirectional Forwarding Detection
           (BFD)", RFC 5880, DOI 10.17487/RFC5880, June 2010,
           <https://www.rfc-editor.org/info/rfc5880>.

[RFC5881] Katz, D. and D. Ward, "Bidirectional Forwarding Detection
           (BFD) for IPv4 and IPv6 (Single Hop)", RFC 5881, DOI
           10.17487/RFC5881, June 2010, <https://www.rfc-editor.org/
           info/rfc5881>.

[RFC8402] Filsfils, C., Ed., Previdi, S., Ed., Ginsberg, L.,
           Decraene, B., Litkowski, S., and R. Shakir, "Segment
           Routing Architecture", RFC 8402, DOI 10.17487/RFC8402,
           July 2018, <https://www.rfc-editor.org/info/rfc8402>.

[RFC8663] Xu, X., Bryant, S., Farrel, A., Hassan, S., Henderickx,
           W., and Z. Li, "MPLS Segment Routing over IP", RFC 8663,
           DOI 10.17487/RFC8663, December 2019, <https://www.rfc-
           editor.org/info/rfc8663>.

Authors' Addresses

Mach(Guoyi) Chen
Huawei

Email: mach.chen@huawei.com

Cheng Li
Huawei

Email: c.l@huawei.com

Wenying Jiang
China Mobile

Email: jiangwenying@chinamobile.com

Yisong Liu
China Mobile

Email: liuyisong@chinamobile.com

Xinjun Chen
Huawei

Email: ifocus.chen@huawei.com