Network Working Group
Internet Draft
Intended status: Standards Track
Expires: August 5, 2016

X. Yang
L. Zhu
G. Karagiannis
Huawei Technologies
February 5, 2016

SFC Trace Issue Analysis and Solutions
draft-yang-sfc-trace-issue-analysis-01.txt

Abstract

This document analyzes and provides solutions for some unaddressed SFC Traceroute issues.

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 http://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."

Copyright Notice

Copyright (c) 2015 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 (http://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 to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction

[I-D.ietf-sfc-oam-framework] provides a reference framework for SFC OAM and lists several OAM functions that help to monitor the SFC components. [I-D.penno-sfc-trace] describes a solution of SFC traceroute based on NSH header, but only a subset of the requirements provided in [I-D.ietf-sfc-oam-framework] are addressed. The goal of this draft is to provide solutions for the rest of the requirements and as well as analyze other potential issues.

2. Terminology

The reader should be familiar with the terms contained in [I-D.ietf-sfc-architecture], [I-D.ietf-sfc-oam-framework], [I-D.ietf-sfc-nsh] and [I-D.penno-sfc-trace].

3. SFC Trace

In [I-D.ietf-sfc-oam-framework], four requirements on the SFC trace function are provided:

o) Ability to trigger action from every transit device on the tested layer towards an SF or through an SFC, using TTL (Time To Live) or other means.

o) Ability to trigger every transit device to generate response with OAM code(s) on the tested layer towards an SF or through an SFC, using TTL or other means.

o) Ability to discover and traverse ECMP paths within an SFC.

o) Ability to skip un-supported SF's while tracing SF's in an SFC.

The first two requirements are met and solved in [I-D.penno-sfc-trace], but the third and fourth requirements are not yet addressed.

Besides these two requirements, there are several issues that need to be analyzed, such as reporting SFF information, TTL-agnostic solution, etc. These issues are further described in the following sub-sections of this document.

3.1. Skip Unsupported SFs

As stated above, the SFC trace function is preferred to skip un-supported SF while tracing. The current solution depends on the SF to provide this information. This means that if the SF will not support the SFC trace function, then no information will be reported back. The result is similar to an error situation, and may disrupt the optimal control plane operation.

One possible solution is to move all trace related functionalities to the SFF, without making any assumptions on the SF for supporting the trace functionality. If the SF does not support the trace function, then the SFF can provide additional information, such as the IP address of the SF instead.

3.2. ECMP Support

When ECMP is deployed, there can be multiple rendered service paths corresponding to one service path. One trace packet can only traverse one of the rendered service path and trigger reports along that path. Furthermore, trace packets sent at different time may follow different rendered service path, which makes it harder to monitor the overall situation of the service path.
To fulfill the need of "discover and traverse all ECMP paths ", one possible solution for the SFF is to broadcast the trace packet to all possible next hops. To identify the exact rendered service path that the packet traversed, information needs to be recorded in the trace packet. The most straightforward way is to add each SF/SFF 's information, e.g., name, to the packet. However, uncontrolled broadcasting can generate a significant amount of traffic on the data plane, which may impact the normal forwarding of the service traffic. Using TTL-agnostic solutions can help to reduce the number of broadcasting packets. More study is needed on this topic, but it is considered to be out of the scope of this document.

3.3. Reporting SFF Information

Providing information of SFFs can help identifying errors on the service path in situations like locating the place where forwarding errors occurred, detecting loops, etc.

3.4. TTL-agnostic Solution

Because NSH is not containing a TTL field, the SFC trace function does not necessarily need to follow a traditional TTL based trace solution. In other words, the trace can be done by sending one trace packet to trigger every traversed SFF to send reports of SFs and/or SFFs along the traversed path.

3.5. Sending Report Message to OAM Controller

The SFC OAM control plane can be centralized or distributed. In the centralized case, the trace report packet can be forwarded to the control plane directly. In the distributed case, however, the OAM control entity may not be directly connected with the SFF, so a dedicated control path or a reverse path is needed to forward the report packet.

3.6. More Command Parameters

Information like service path ID, starting service index, and report address are needed to perform a trace. As described in the above sub-sections, there are many aspects impacting the behavior of a particular trace process. They all can be captured as trace command parameters. The following list gives several command parameters that are worth to be taken into consideration:

- o) service path identification
- o) starting service index
- o) Service Index Limit (SIL, described in Section 3.7)
- o) report destination IP address and port
- o) report object: sending report of SF, SFF or both
- o) ECMP support
- o) number of queries to send per hop
- o) time to wait for a response/report
- o) number of queries that can be sent out simultaneously
- o) time interval between sending queries

3.7. Basic SFC Trace Header

The trace headers shown in Figure 1 (Trace Request Header) and in Figure 2 (Trace Report Header) are used as a basis for the SF trace operation described in the following sections.

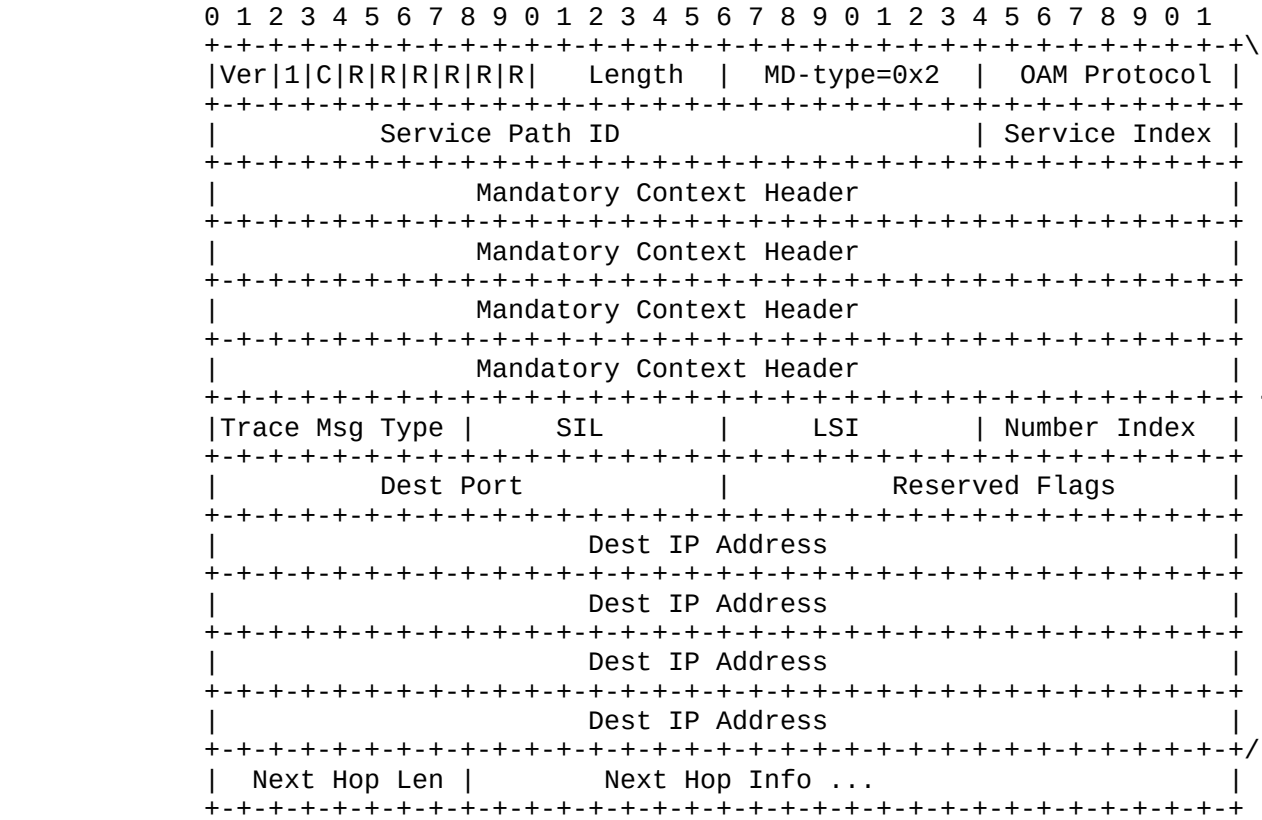

Figure 1: Trace Request Header

Trace Msg Type: 1 for Trace Request and 2 for Trace Report
SIL: Service Index Limit: At least one less than the Starting Index
LSI: Last Service Index, record the service index of the last service function which processed the packet, default valve is the starting SI
Number Index (NI): number of hops the packet has traversed, default value is 0
Reserved Flags: can be used to indicate the function blocks that need to send reports, whether uses ECMP, etc.
Dest Port: The trace report must be sent to this destination Port

Dest IP: the trace report must be sent to this destination IP address, IPv6 format.

Next Hop Len: The length of Next Hop Info in 4-byte words. The field only exists when needed.

Next Hop Info: A string that records the identification of the next hop, e.g., name, IP address, etc. The field only exists when needed.

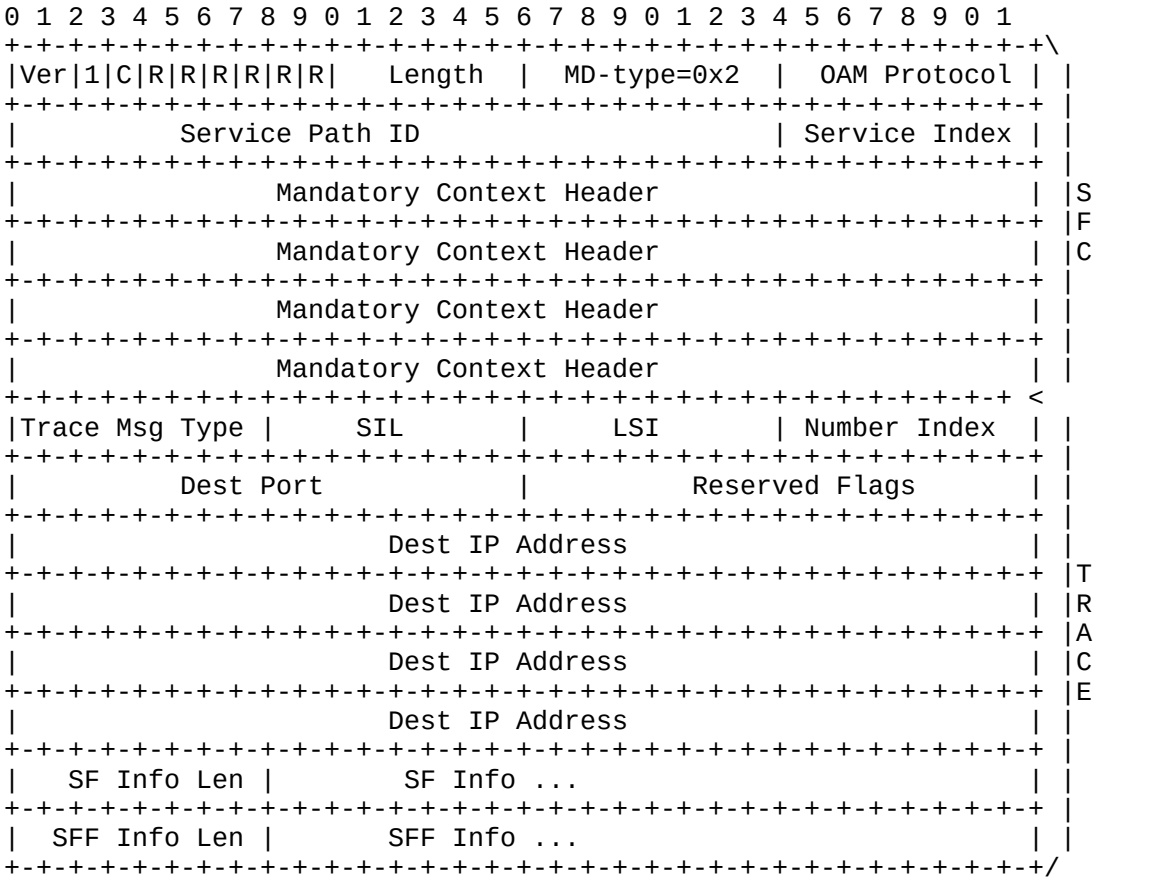

Figure 2: Trace Report Header

SF Info Len: The SF Info length in 4-byte words. This field is omitted when reporting an SFF.

SF Info: A string that represents the identification of an SF. This field is omitted when reporting an SFF.

SFF Info Len: The SFF Info length in 4-byte words. This field is omitted when reporting an SF.

SFF Info: A string that represents the identification of an SFF. This field is omitted when reporting an SFF.

4. Service Function Behavior

As stated in 3.1, in order to skip unsupported SFs the trace functionalities is moved to the SFFs. In this situation, the SF only needs to have the ability to process the NSH and no assumption is made on whether the SF is supporting the trace function.

When an SF receives a trace packet, it performs the following actions:

1. Decrement Service Index in NSH

2. (Only conducted when trace function is supported) If Service Index is equal to the Services Index Limit, replace the Next Hop Info field with its identification information

3. Send packet back to SFF

5. Service Function Forwarder Behavior

The trace functionality is mainly implemented in the SFF. Section 5.1 describes the basic behavior of the SFF. Sections 5.2 and 5.3 describe the changes to the SFF default behavior, assuming either that the SFF information reporting is enabled or by adopting the TTL-agnostic solution.

5.1. Skip Unsupported SFs

When an SFF receives a trace request packet, it performs the following actions:

1. Checking if the trace packet should be dropped

2. If SI is 1 greater than SIL, and if LSI is greater than SI, the SFF will add the Next Hop Info field to the trace header with its next hop information. If SI is 1 greater than SIL, and if LSI is equal to SI, the SFF will overwrite the Next Hop Info field in the header.

   NOTE: This assumes that the SFF cannot identify whether the next hop is an SF or an SFF. If the SFF can identify the type of the next hop, it can then add the Next Hop Info field to the trace header until finding the SI is 1 greater than SIL and the next hop is an SF.

3. If LSI is greater than SI, change the LSI to be equal to SI.

4. Forward the trace packet to the next hop

If at least one of the following conditions is met, the trace packet will be dropped and a trace report packet is generated:

- o) the SI is equal or less than SIL

- o) the SFF cannot find the next hop to forward the packet

- o) the SI is equal to zero

The following steps are applied to generate a trace report packet:

i. Fill in the NSH header with proper values

ii. Copy the information from the trace request header to the trace report header. (The Next Hop Info field's information will be copied to SF Info field.)

iii. Change the Trace Msg Type field to 2 (Trace Report).

5.2. Reporting SFF Information

As described in Section 5.1, a trace request packet will only trigger one report packet which contains the information of the last hop SF. To completely monitor a service path, several trace request packets are needed. When reporting SFF information, similar behavior is needed to avoid redundant reports of SFFs, i.e., a trace request packet will only trigger report packets generated on SFFs between the last hop SF and the second last hop SF.

Compared to the default behavior described in Section 5.1, only the step 2 is changed when an SFF receives a trace request packet:

o) If SI is 1 greater than SIL, and if LSI is greater than SI, the SFF will add information of the next hop to the trace header. If SI is 1 greater than SIL, and if LSI is equal to SI, the SFF will overwrite next hop information in the header, increase NI and trigger an SFF report.

The NI field is used to record the order of the report packets, which helps to sequence the reports in the control plane.

The following steps are taken when an SFF report is triggered:

1. Fill in the NSH header with proper values

2. Copy the information from the trace request header to the trace report header except for the Next Hop Info field (if it exists).

3. Add the SFF Info Len and SFF Info fields to the report header with the SFF's identification information

4. Change the Trace Msg Type field to 2 (Trace Report).

5.3. TTL-agnostic Solution

As described in Section 3.4, when using TTL-agnostic solution, only one trace request packet is needed to conduct a complete trace process.

Compared to the default behavior described in Section 5.1, only the step 2 is changed when an SFF receives a trace request packet:

o) If LSI is greater than SI, the SFF will add information of the next hop to the trace header, increase NI and trigger an SF report

o) If LSI is equal to SI, the SFF will overwrite next hop information in the header, increase NI and trigger an SFF report

In this scenario the Next Hop Len and Next Hop Info fields are always needed in the trace header, except in the situation that the SFF can identify the type of the next hop. In that situation, the two fields are only needed when the next hop is an SF and when its ID needs to be added to the trace header by the SFF.

The report packet generation process is similar to the ones described in Section 5.2 and 5.3.

6. NSH-unware SF

As stated in section 5, the trace functionality described in this draft is mainly implemented by the SFF instead of SF. As a result, the functionality can also be used in the case where the SFs do not support NSH.

In such a case (or more broadly, in a case where the SFs do not support SFC encapsulation), a proxy is needed between the SFF and SF to process the NSH header. The proxy will remove the NSH header before forwarding the packet to the SF and apply the encapsulation to the packet when it’s returned back. It is expected that proxy can conserve the trace request header when removing the NSH header and restore it when applying the encapsulation. Like a SF supporting the trace functionality (as stated in section 3.1), if the proxy supports the trace functionality, it can provides additional information of the SF by modifying the Next Hop Info field. It can apply the modification either when receiving the packet or when applying the encapsulation.

7. IANA Considerations

IANA considerations are needed for the registration of (1) OAM Protocol Type and (2) OAM protocol Message type.

8. Security Considerations

As stated in section 3.5, if the SFC OAM control plane is centralized, the trace report packets can be forwarded to the control plane directly. In this case, sending one trace request packet can cause one or more report packets (based on whether reporting SFF information and/or adopting TTL-agnostic solution) sent to the controller. This may bring potential security issue, like DDoS attack. One possible way to solve this problem is to authenticate the trace request packet. Further study is needed for this aspect.

9. Acknowledgements

To be done.

10. References

10.1. Normative References

10.2. Informative References

[I-D.ietf-sfc-architecture] Halpern, J. and C. Pignataro, "Service Function Chaining (SFC) Architecture", draft-ietf-sfc-architecture-11 (work in progress), July 2015.

[I-D.ietf-sfc-oam-framework] Alfrin, S., Krishnan, R., Akiya, N., Pignataro, C., and A. Ghanwani, "Service Function Chaining Operation, Administration and Maintenance Framework", draft-ietf-sfc-oam-framework-00 (work in progress), August 2015.

[I-D.penno-sfc-trace] Penno, R., Quinn, P., Pignataro, C., and D. Zhoud, "Service Function Chaining Traceroute", draft-penno-sfc-trace-03 (work in progress), September 2015.

[I-D.ietf-sfc-nsh] Quinn, P., and U. Elzur, "Network Service Header", draft-ietf-sfc-nsh-01 (work in progress), July 2015.

Authors' Addresses

Xu Yang
Huawei Technologies
Huawei Building,No. 3, Xinxi Road, Haidian District, Beijing
China
Email: yangxu5@huawei.com

Lei Zhu
Huawei Technologies
Huawei Building,No. 3, Xinxi Road, Haidian District, Beijing
China
Email: Lei.zhu@huawei.com

Georgios Karagiannis
Huawei Technologies
Hansaallee 205,
40549 Dusseldorf,
Germany
Email: Georgios.Karagiannis@huawei.com