

Network working group
Internet Draft
Intended status: Standard Track
Expires: October 2014

L. Dunbar
A. Malis
Huawei

April 29, 2014

Framework for Service Function Instances Restoration
[draft-dunbar-sfc-fun-instances-restoration-00.txt](#)

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

This Internet-Draft is submitted in full conformance with the provisions of [BCP 78](#) and [BCP 79](#). This document may not be modified, and derivative works of it may not be created, except to publish it as an RFC and to translate it into languages other than English.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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."

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/ietf/lid-abstracts.txt>

The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>

This Internet-Draft will expire on October 30, 2014.

Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](http://trustee.ietf.org/license-info) 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.

Abstract

This draft describes the framework of protection and restoration of Service Chain Instance Path when some instances on the path fail or need to be replaced.

Table of Contents

1.	Introduction.....	2
2.	Conventions used in this document.....	3
3.	Background.....	4
3.1.	Multiple Instances of one Service Function.....	4
3.2.	Multiple ways for expressing Service Chain Instance Path..	4
3.3.	Virtualized Service Function Instances impact to Service Chain.....	6
4.	Local Repair of Service Function Instances.....	7
5.	Global Repair of Service function instances.....	8
6.	Regional Repair of Service function instances.....	10
7.	Conclusion and Recommendation.....	10
8.	Manageability Considerations.....	10
9.	Security Considerations.....	10
10.	IANA Considerations.....	10
11.	References.....	11
11.1.	Normative References.....	11
11.2.	Informative References.....	11
12.	Acknowledgments.....	11

[1.](#) Introduction

This draft describes the framework for protection and restoration of a Service Chain Instance Path when some instances on the path fail or need to be replaced.

Protection and restoration become more crucial in virtualized environments (e.g. ETSI NFV), where there is higher chance of

Service function instances failing, being decommissioned or over-utilized.

[2.](#) Conventions used in this document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC-2119](#) [[RFC2119](#)].

In this document, these words will appear with that interpretation only when in ALL CAPS. Lower case uses of these words are not to be interpreted as carrying [RFC-2119](#) significance.

[3.](#) Definition of Terms

NFV: Network Function Virtualization [[NFV-Terminology](#)].

SF: Service Function [[SFC-Problem](#)].

SFF: Service Function Forwarder.

SFIC: Service Function Instance Component. One service function (e.g. NAT44) could have two different service function instantiations, one that applies policy-set-A (NAT44-A) and other that applies policy-set-B (NAT44-B). There could be multiple "entities" of NAT44-B (e.g. one "entity" only has 10G capability), and many "entities" of NAT44-B. Each entity has its own unique address. The "entity" in this context is called "Service Function Instance Component" (SFIC).

Service Chain: The sequence of service functions, e.g. Chain#1 {s1, s4, s6}, Chain#2{s4, s7} at functional level. Also see the definition of "Service Function Chain" in [[SFC-Problem](#)].

Service Chain Instance Path: The actual Service Function Instance Components selected for a service chain.

SFF: Service Function Forwarding Node.

VNF: Virtualized Network Function [[NFV-Terminology](#)].

[4.](#) Background

[4.1.](#) Multiple Instances of one Service Function

One service function (say, NAT44) could have two different service function instantiations, one that applies to policy-set-A (NAT44-A) and other that applies to policy-set-B (NAT44-B). There could be multiple "entities" of NAT44-A (e.g. one "entity" only has 10G capability), and many "entities" of NAT44-B. Each entity has its own unique address (or Locator in [[SFC-Reduction](#)]). The "Entity" in this context is called "Service Function Instance Component" (SFIC).

Identical SFICs could be attached to different Service Function Forwarder (SFF) nodes. It is also possible to have multiple identical SFICs attached to one Service Function Forwarder (SFF) node, especially in a Network Function Virtualization (NFV) environment where each SFIC is a virtual service function instance with limited capacity.

At the functional level, the order of service functions, e.g. Chain#1 {s1, s4, s6}, Chain#2{s4, s7}, is important, but very often which SFIC of the Service Function "s1" is selected for the Chain #1 is not. It is also possible that multiple SFICs of one service function can be reached by different network nodes. The actual SFIC selected for a service chain is called "Service Chain Instance Path".

[4.2.](#) Multiple ways for expressing Service Chain Instance Path

How SFICs are selected for a given Service Chain to form the actual Service Chain Instance Path is outside the scope of this draft. It is assumed that there is an entity (e.g. service chain orchestration system) that is responsible for selecting the SFICs for a Service

Chain.

This document focuses on how Service Function Forwarder nodes or network nodes are informed of the selected SFICs for a particular Service Chain, especially when there are changes of SFICs on the Service Chain. To make description easier, the following Service Chain architecture reference is used:

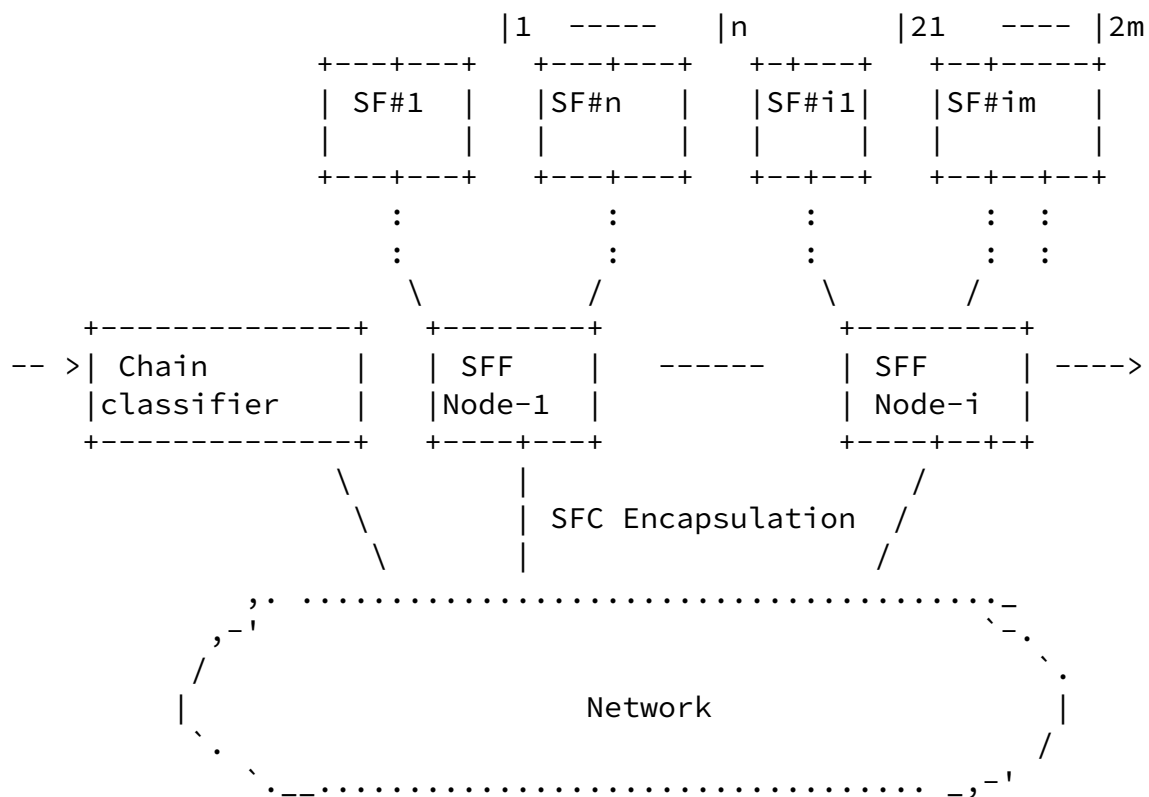


Figure 1

Framework of Serv

Some head end Service Chain Classifier can be configured with (or has the ability to specify) the exact Service Chain Instance Path

for a given service chain. Under this scenario, the exact Service Chain Instance Path can be expressed by:

- Being encoded in every data packet;
- Being signaled in-band via the data path from the head end Service Chain Classifier node to all the relevant nodes to install the appropriate flow steering policies (similar to MPLS traffic engineering signaling);
- Being sent as out-of-band control messages to all the relevant nodes to install the appropriate flow steering policies (similar to GMPLS signaling); or
- Being provisioned into each node by a centralized network controller (similar to SDN) or by a network management system.

The benefit of encoding the exact path in every data packet is less contention when there the Service Chain Instance Path changes. However, there are major drawbacks, such as

- extra packet header fields are needed to carry the exact instance path, that can increase the likelihood of packet fragmentation due to MTU size, and
- extra encapsulation processing load at the head end Service Chain classifier node.

Packet fragmentation and reassembly is very processor and memory intensive. Good practice is to avoid packet fragmentation and reassembly as much as possible. Carry an exact instance path in every packet might be possible if service function instances can be represented by compact labels, similar to the MPLS label stack.

When the in-band or out-of-band signaling methods are used, i.e. sending flow steering policies to relevant SFF nodes or network nodes, the packets associated with a specific flow can be classified with a simple identifier (or Service Chain ID). Packet size is smaller and processing at the SC Classifier can be simpler as well.

The out-of-band method doesn't even require the head end Service Chain Classifier to be configured with, nor has the capability to specify, the exact Service Chain Instance Path. The out-of-band

steering policies can be sent from an external entity, such as a centralized network controller or service chain orchestration system. Under this scenario, it doesn't require the head end Chain Classifier node to be aware of any change to the instances on the chain.

At times it might not be feasible for the head end Service Chain Classifier to be aware of the exact instances selected for a given Service Chain because they are managed by different administrative entities.

If each Service Function has a large number of SFICs, it scales better if the Service Chain classifier only identifies the service chain at the functional level, and there is another entity managing the detailed service instance path.

[4.3.](#) Virtualized Service Function Instances impact to Service Chain

When Service Chain Instance Path consists of virtualized service function instances, e.g. in an ETSI NFV environment, the likelihood

Dunbar, et al.

Expires October 29, 2014

[Page 6]

Internet-Draft

SF Instances Restoration Framework

April 2014

or frequent changes to the Service Chain Instance Path might be higher due to:

- Higher failure rate of virtualized service function instances because most of them will not have build-in protection mechanism
- When some instances are over-utilized, it is relatively easy to replace them by other instances or instantiate more instances to take over the work load.

[5.](#) Local Restoration of Service Function Instances

When one SF Forwarder (SFF) node has multiple Service Function Instance Components (SFICs) of the same service function attached, the SFF can make a local decision on which instance is selected for a specific service chain.

E.g. In the diagram below, The SF Forwarder (SFF) "A" has two instances of Service Function #7(SF7-1 & SF7-2), and 3 instances of Service Function #2 (SF2-2, SF2-4, SF2-5).

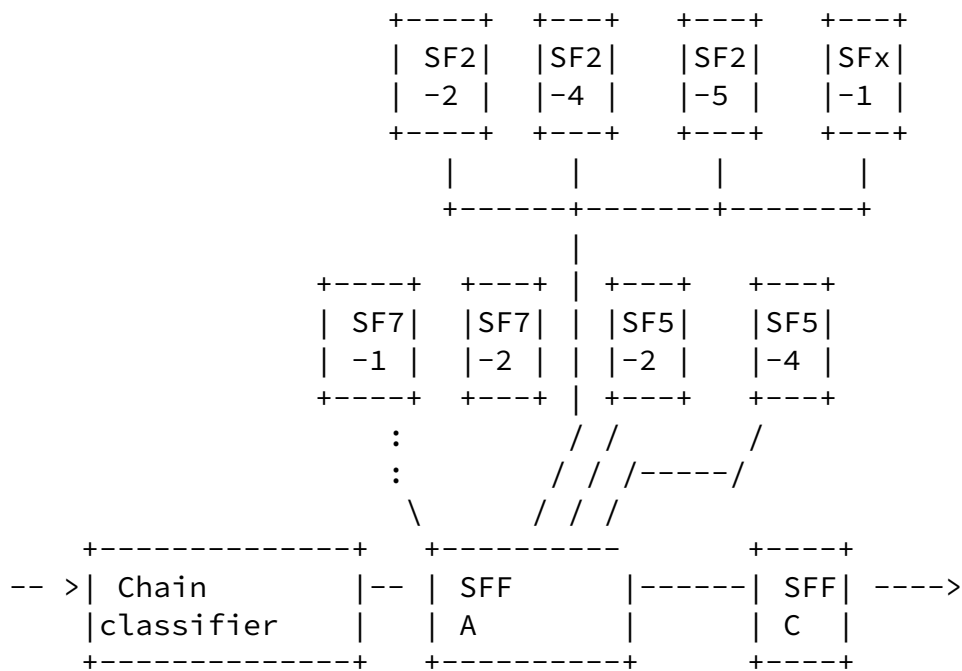


Figure 2 Local Restoration among multiple service

For a service chain that consists of "Service Function #7" followed by "Service Function #2", which is represented by SF7->SF2, the steering policy to SFF "A" could be:

{SF7-1, SF7-3} -> {SF2-2, SF2-4, SF2-5}.

The multiple components within the {} represents the equal function instances that SFF "A" can select locally.

When one service function instance fails, the SFF "A" can locally choose another instance without informing the SC Classifier node, or other SFF or network nodes.

The local protection and restoration is relatively simple and clean. ECMP can be used to balance all the available service function instances attached locally.

6. Global Restoration of Service function instances

Sometimes changing the Service Chain Instance Path involves using service function instances at different SF Forwarding (SSF) nodes.

For example, for a Chain #7 -> #2 -> #3 -> #5 in the figure above:

- Original instance path: #7 & #2 at SFF "A"; #3 & #5 at SFF "C".
- New instance path: #7 at SFF "A" and #2& #3 & #5 at SFF "C".

This section examines possible ways to achieve the restoration when the change of instance path involves multiple nodes.

[6.1](#). Encoding the Exact Instance Path in Data Packets

If the detailed Service Chain Instance Path is encoded in data packets, the SC Classifier can be notified of the change and encode the new instance path in the data packets of the flow. This method won't cause any contention issue among all the involved nodes.

As mentioned in the previous section, encoding exact instance path in every packet can cause packets fragmentation, which is very processing intensive. Therefore, it's not optimal to require every data packet to carry an exact instance path, especially when the Service Chain instance path doesn't change very frequently, as in minutes or hours.

[6.2](#). In-Band Signaling of an Instance Path change

A similar method to MPLS RSVP-TE [[RSVP-TE](#)] signaling can be considered for the head end node to signal a required service instance path, and then let the data packets traverse the established path.

The drawback of this approach is that the head end node might receive packets belonging to the service chain before the instance path has been established. It is very similar to the issues encountered by MPLS Fast Reroute [[FRR](#)]. MPLS FRR requires that packets be dropped if a restoration path is being dynamically signaled because there was not a pre-established backup path..

[6.3](#). Out-Of-Band Signaling of an Instance Path change

If the out-of-band method is used, i.e. sending the updated flow steering policies to indicate the changes of the instance path, there could be issues of synchronization and race conditions. For example, if the SFF "A" and SFF "C" get flow steering policies at slightly different times, some packets of the flow might miss some service functions on the chain.

[6.4.](#) Provisioning an Instance Path change

In SDN or SDN-like environments, changes to the Instance Path can be provisioned or programmed into network nodes via a central controller or Network Management System (NMS). This simplifies the nodes, since they are not required to use a signaling protocol, but there may be problems introduced (such as loops or dropped packets) if network nodes are not updated in the proper order or very soon to each other; the nodes should be updated in a similar time scale to the use of a signaling protocol. In addition, the network may have a single point of failure if the controller or NMS is not itself redundant.

[6.5.](#) Hybrid Method

For global restoration of service function instances, it is worthwhile to explore a hybrid mode, i.e. when there are changes involving using service instances at different SFF nodes, the SC Classifier node is informed to encode the detailed instance path to data packets until all the involved SFF nodes complete the installation of the new steering policy for the flow.

[7.](#) Regional Restoration of Service Function Instances

It might not be always be feasible for the head end Service Chain Classifier to be aware of the exact instances selected for a given Service Chain due to being managed by multiple administrative entities. Then Regional restoration should be considered.

Regional restoration can take the similar approach as the Global restoration: choosing a regional ingress node that can take over the responsibility of installing the new steering policies to the

involved SFF nodes or network nodes.

The Regional ingress node should be:

- on the data path of the flow of the given service chain;
- in front of the relevant the SFF nodes or network nodes that are impacted by the change of the Service Chain Instance Path;
- capable of encoding the detailed Service Chain Instance Path to the data packets of the identified flow; and
- capable of removing the detailed Service Chain Instance Path encoding in data packets after all the impacted SFF nodes and network nodes completed the policy installation.

[8.](#) Conclusion and Recommendation

TBD

[9.](#) Manageability Considerations

TBD

[10.](#) Security Considerations

TBD

[11.](#) IANA Considerations

This document requires no IANA actions. RFC Editor: Please remove this section before publication.

[12.](#) References

[12.1.](#) Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate

Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.

[12.2](#). Informative References

- [SFC-Problem] P. Quinn, et al, "Service Function Chaining Problem statement", [draft-ietf-sfc-problem-statement-02](#), work in progress, April 2014
- [NFV-Terminology] ETSI NFV ISG, "Network Functions Virtualisation (NFV); Terminology for Main Concepts in NFV", ETSI GS NFV 003 V1.1.1, Oct. 2013,
http://www.etsi.org/deliver/etsi_gs/NFV/001_099/003/01.01.01_60/gs_NFV003v010101p.pdf
- [SFC-Reduction] R. Parker, "Service Function Chaining: Chain to Path Reduction", [draft-parker-sfc-chain-to-path-00](#), work in progress, Nov. 2013
- [RSVP-TE] D. Awduche, Berger, L., Gan, D., Li, T., Srinivasan, V., and G. Swallow, "RSVP-TE: Extensions to RSVP for LSP Tunnels", [RFC 3209](#), December 2001.
- [FRR] P. Pan, Swallow, G., and Atlas, A., "Fast Reroute Extensions to RSVP-TE for LSP Tunnels", [RFC 4090](#), May 2005

[13](#). Acknowledgments

Many thanks to Ron Bonica for the discussion in formulating the content for the draft.

This document was prepared using 2-Word-v2.0.template.dot.

Authors' Addresses

Linda Dunbar
Huawei Technologies
5340 Legacy Drive, Suite 175
Plano, TX 75024, USA
Phone: (469) 277 5840
Email: ldunbar@huawei.com

USA
Email: rbonica@juniper.net

Andrew G. Malis
Huawei Technologies
USA
Email: agmalis@gmail.com