SFC
Internet-Draft
Intended status: Informational
Expires: August 17, 2015

M. Boucadair
C. Jacquenet
France Telecom
Y. Jiang
Huawei Technologies Co., Ltd.
R. Parker
Affirmed Networks
K. Naito
NTT

February 13, 2015

Requirements for Service Function Chaining (SFC)
draft-boucadair-sfc-requirements-06

Abstract

This document identifies the requirements for the Service Function Chaining (SFC).

Requirements Language

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].

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."

This Internet-Draft will expire on August 17, 2015.

Copyright Notice

Table of Contents

1. Introduction

This document identifies the requirements for the Service Function Chaining (SFC).

The overall problem space is described in [I-D.ietf-sfc-problem-statement].

2. Terminology

The reader should be familiar with the terms defined in [I-D.ietf-sfc-problem-statement].

The document makes use of the following terms:

- SFC-enabled domain: denotes a network (or a region thereof) that implements SFC.

- Service Function Loop: If a Service Function Chain is structured to not invoke Service Functions multiple times, a loop is the error that occurs when the same Service Function is invoked several times when handling data bound to that Service Function Chain. In other words, a loop denotes an error that occurs when a packet handled by a Service Function, forwarded onwards, and arrives once again at that Service Function while this is not allowed by the Service Function Chain it is bound to.

- Service Function Spiral: denotes a Service Function Chain in which data is handled by a Service Function, forwarded onwards, and arrives once again at that Service Function.

  * Note that some Service Functions support built-in functions to accommodate spirals; these service-specific functions may require that the data received in a spiral should differ in a way that will result in a different processing decision than the original data. This document does not make such assumption.

  * A Service Function Chain may involve one or more Service Function Spirals.

  * Unlike Service Function loop, spirals are not considered as errors.

3. Detailed Requirements List

The following set of functional requirements should be considered for the design of the Service Function Chaining solution.

3.1. Instantiating and Invoking Service Functions

SF_REQ#1: The solution MUST NOT make any assumption on whether Service Functions (SF) are deployed directly on physical hardware, as one or more Virtual Machines, or any combination thereof.

SF_REQ#2: The solution MUST NOT make any assumption on whether Service Functions each reside on a separate addressable Network Element, or as a horizontal scaling of Service Functions, or are co-resident in a single addressable Network Element, or any combination thereof.

Note: Communications between Service Functions having the same locator are considered implementation-specific. These considerations are therefore out of scope of the SFC specification effort.

SF_REQ#3: The solution MUST NOT require any IANA registry for Service Functions.

SF_REQ#4: The solution MUST allow multiple instances of a given Service Function (i.e., instances of a Service Function can be embedded in or attached to multiple Network Elements).

A. This is used for load-balancing, load-sharing, to minimize the impact of failures (e.g., by means of a hot or cold standby protection design), to accommodate planned maintenance operations, etc.

B. How these multiple devices are involved in the service delivery is deployment-specific.

SF_REQ#5: The solution MUST separate SF-specific policy provisioning-related aspects from the actual handling of packets (including forwarding decisions).

3.2. Chaining Service Functions

SFC_REQ#1: The solution MUST NOT assume any predefined order of Service Functions. In particular, the solution MUST NOT require any IANA registry to store typical Service Function Chains.

SFC_REQ#2: The identification of instantiated Service Function Chains is local to each administrative domain; it is policy-based and deployment-specific.

SFC_REQ#3: The solution MUST allow for multiple Service Chains to be simultaneously enforced within an administrative domain.

SFC_REQ#4: The solution MUST allow the same Service Function to belong to multiple Service Function Chains.

SFC_REQ#5: The solution MUST support the ability to deploy multiple SFC-enabled domains within the same administrative domain.

SFC_REQ#6: The solution MUST be able to associate the same or distinct Service Function Chains for each direction

(inbound/outbound) of the traffic pertaining to a specific service. In particular, unidirectional Service Function Chains, bi-directional Service Function Chains, or any combination thereof MUST be supported.

Note, the solution must allow to involve distinct SFC Boundary Nodes for upstream and downstream. Multiple SFC Boundary Nodes may be deployed within an administrative domain.

SFC_REQ#7: The solution MUST be able to dynamically enforce Service Function Chains. In particular, the solution MUST allow the update or the withdrawal of existing Service Function Chains, the definition of a new Service Function Chain, the addition of new Service Functions without having any impact on other existing Service Functions or other Service Function Chains.

SFC_REQ#8: The solution MUST provide means to control the SF-inferred information to be leaked outside an SFC-enabled domain. In particular, an administrative entity MUST be able to prevent the exposure of the Service Function Chaining logic and its related policies outside the administrative domain.

SFC_REQ#9: The solution MUST prevent infinite Service Function Loops.

A. Service Functions MAY be invoked multiple times in the same Service Function Chain (denoted as SF Spiral), but the solution MUST prevent infinite forwarding loops.

3.3. MTU Requirements

Packet fragmentation can be very expensive in SFC environment where fragmented packets have to be reassembled before sending to each SF on the chain. It is also worth noting that IPv6 traffic can only be fragmented by the end systems.

MTU_REQ#1: The solution SHOULD minimize fragmentation; in particular, a minimal set of SFC-specific information should be conveyed in the data packet.

MTU_REQ#2: Traffic forwarding on a SFC basis MUST be undertaken without relying on dedicated resources to treat fragments. In particular, Out of order fragments MUST be

forwarded on a per-SFC basis without relying on any state.

MTU_REQ#3: Some SFs (e.g., NAT) may require dedicated resources (e.g., resources to store fragmented packets) or they may adopt a specific behavior (e.g, limit the time interval to accept fragments). The solution MUST NOT interfere with such practices.

3.4. Independence from the Underlying Transport Infrastructure Requirements

UN_REQ#1: The solution MUST NOT make any assumption on how RIBs (Routing Information Bases) and FIBs (Forwarding Information Bases) are populated. Particularly, the solution does not make any assumption on protocols and mechanisms used to build these tables.

UN_REQ#2: The solution MUST be transport independent.

A. The Service Function Chaining should operate regardless of the network transport used by the administrative entity. In particular, the solution can be used whatever the switching technologies deployed in the underlying transport infrastructure.

B. Techniques such as MPLS are neither required nor excluded.

UN_REQ#3: The solution MUST allow for chaining logics where involved Service Functions are not within the same layer 3 subnet.

UN_REQ#4: The solution MUST NOT exclude Service Functions to be within the same IP subnet (because this is deployment-specific).

3.5. Traffic Classification Requirements

TC_REQ#1: The solution MUST NOT make any assumption on how the traffic is to be bound to a given chaining policy. In other words, classification rules are deployment-specific and policy-based. For instance, classification can rely on a subset of the information carried in a received packet such as 5-tuple classification, be subscriber-aware, be driven by traffic engineering considerations, or any combination thereof.

Because a large number (e.g., 1000s) of classification policy entries may be configured, means .Means to reduce classification look-up time such as optimizing the size of the classification table (e.g., aggregation) should be supported by the Classifier.

TC_REQ#2: The solution MUST NOT require every Service Function to be co-located with a SFC Classifier; this is a deployment-specific decision.

TC_REQ#3: The solution MAY allow traffic re-classification at the level of Service Functions (i.e., a Service Function can also be co-located with a Classifier). The configuration of classification rules in such context are the responsibility of the administrative entity that operates the SFC-enabled domain.

TC_REQ#4: The solution MUST allow Service Function Nodes to be configured (or pushed) with the detailed policies on which local Service Functions to invoke for packets associated with some Service Function Chains. The solution MUST allow those steering policies to be updated based on demand.

3.6. Data Plane Requirements

DP_REQ#1: The solution MUST be able to forward traffic between two Service Functions (involved in the same Service Function Chain) without relying upon the destination address field of the a data packet.

DP_REQ#2: The solution MUST allow for the association of a context with the data packets. In particular:

A. The solution MUST support the ability to invoke differentiated sets of policies for a Service Function (such sets of policies are called Profiles). A profile denotes a set of policies configured to a local Service Function (e.g., content-filter-child, content-filter-adult).

a. Few profiles should be assumed per Service Function to accommodate the need for scalable solutions.

b. A finer granularity of profiles may be configured directly to each Service Function; there is indeed

no need to overload the design of Service Function Chains with policies of low-level granularity.

DP_REQ#3: Service Functions may be reachable using IPv4 and/or IPv6. The administrative domain entity MUST be able to define and enforce policies with regards to the address family to be used when invoking a Service Function.

A. A Service Function Chain may be composed of IPv4 addresses, IPv6 addresses, or a mix of both IPv4 and IPv6 addresses.

B. Multiple Service Functions can be reachable using the same IP address. Each of these Service Functions is unambiguously identified with a Service Function Identifier.

DP_REQ#4:

3.7. OAM Requirements

OAM_REQ#1: The solution MUST allow for Operations, Administration, and Maintenance (OAM) features [RFC6291]. In particular, the solution MUST:

A. Support means to verify the completion of the forwarding actions until the SFC Border Node is reached (see Section 3.4.1 of [RFC5706]).

B. Support means to ensure coherent classification rules are installed in and enforced by all the Classifiers of the SFC domain.

C. Support means to correlate classification policies with observed forwarding actions.

D. Support in-band liveliness and functionality checking mechanisms for the instantiated Service Function Chains and the Service Functions that belong to these chains.

OAM_REQ#2: The solution MUST support means to detect the liveliness of Service Functions of an SFC-enabled domain. In particular, the solution MUST support means to (dynamically) detect that a Service Function instance is out of service and notify the relevant elements accordingly (PDP and Classifiers, for one).

OAM_REQ#3: Detailed diagnosis requirements are listed below:

A. The solution MUST allow to assess the status of the serviceability of a Service Function (i.e., the Service Function provides the service(s) it is configured for).

B. The solution MUST NOT rely only on IP reachability to assess whether a Service Function is up and running.

C. The solution MUST allow to diagnose the availability of a Service Function Chain (including the availability of a particular Service Function Path bound to a given Service Function Chain).

D. The solution MUST allow to retrieve the set of Service Function Chains that are enabled within a domain.

E. The solution MUST allow to retrieve the set of s Service Function Chains in which a given Service Function is involved.

F. The solution MUST allow to assess whether an SFC-enabled domain is appropriately configured (including the configured chains are matching what should be configured in that domain).

G. The solution MUST allow to assess the output of the classification rule applied on a packet presented to a Classifier of an SFC-enabled domain.

H. The solution MUST support the correlation between a Service Function Chain and the actual forwarding path followed by a packet matching that SFC.

I. The solution MUST allow to diagnose the availability of a segment of a Service Function Chain, i.e., a subset of Service Functions that belong to the said chain.

J. The solution MUST support means to notify the PDPs whenever some events occur (for example, a malfunctioning Service Function instance).

K. The solution MUST allow for local diagnostic procedures specific to each Service Function (i.e., SF built-in diagnostic procedures).

L. The solution MUST allow for customized service diagnostic.

OAM_REQ#4: Liveness status records for all Service Functions (including Service Function instances), Service Function Nodes, Service Function Chains (including the Service Function Paths bound to a given chain) MUST be maintained.

OAM_REQ#5: SFC-specific counters and statistics MUST be provided. These data include (but not limited to):

* Number of flows ever and currently assigned to a given Service Function Chain and a given Service Function Path.

* Number of flows, packets, bytes dropped due to policy.

* Number of packets and bytes in/out per Service Function Chain and per Service Function Path.

* Number of flows, packets, bytes dropped due to unknown Service Function Chain or Service Function Path (this is valid in particular for a Service Function Node).

3.8. Recovery and Load Balancing Requirements

LB_REQ#1: The solution MUST allow for load-balancing among multiple instances of the same Service Function.

A. Load-balancing may be provided by legacy technologies or protocols (e.g., make use of load-balancers)

B. Load-balancing may be part of the Service Function itself.

C. Load-balancer may be considered as a Service Function element.

D. Because of the possible complications, load balancing SHOULD NOT be driven by the SFC Classifier.

LB_REQ#2: The solution MUST separate SF-specific policy provisioning-related aspects from the actual handling of packets (including forwarding decisions).

LB_REQ#3: The solution SHOULD support protection of the failed or over-utilized Service Function instances. The protection

mechanism can rely on local decisions among the nodes that are connected to both active/standby Service Function instances.

3.9. Compatibility with Legacy Service Functions Requirements

LEG_REQ#1: The solution MUST allow for gradual deployment in legacy infrastructures, and therefore coexist with legacy technologies that cannot support SFC-specific capabilities, such as Service Function Chain interpretation and processing. The solution MUST be able to work in a domain that may be partly composed of opaque elements, i.e., elements that do not support SFC-specific capabilities.

3.10. QoS Requirements

QoS_REQ#1: The solution MUST be able to provide different SLAs (Service Level Agreements, [RFC7297]). In particular,

A. The solution MUST allow for different levels of service to be provided for different traffic streams (e.g., configure Classes of Service (CoSes)).

B. The solution MUST be able to work properly within a Diffserv domain [RFC2475].

C. The solution SHOULD support the two modes defined in [RFC2983].

QoS_REQ#2: ECN re-marking, when required, MUST be performed according to [RFC6040].

3.11. Security Requirements

SEC_REQ#1: The solution MUST provide means to prevent any information leaking that would be used as a hint to guess internal engineering practices (e.g., network topology, service infrastructure topology, hints on the enabled mechanisms to protect internal service infrastructures, etc.).

The solution MUST support means to protect the SFC domain as a whole against attacks that would lead to the discovery of Service Functions enabled in a SFC domain.
In particular, topology hiding means MUST be supported to avoid the exposure of the SFC-enabled domain

topology (including the set of the service function chains supported within the domain and the corresponding Service Functions that belong to these chains).

SEC_REQ#2: The solution MUST support means to protect the SFC-enabled domain against any kind of denial-of-service and theft of service (e.g., illegitimate access to the service) attack.

For example, a user should not be granted access to connectivity services he/she didn't subscribe to (including direct access to some SFs), at the risk of providing illegitimate access to network resources.

SEC_REQ#3: The solution MUST NOT interfere with IPsec [RFC4301] (in particular IPsec integrity checks).

4. IANA Considerations

This document does not require any action from IANA.

5. Security Considerations

Some security-related requirements to be taken into account when designing the Service Function Chaining solution are listed in Section 3.11. These requirements do not cover the provisioning interface used to enforce policies into the Classifier, Service Functions, and Service Function Nodes.

6. Contributors

The following individuals contributed text to the document:

Hongyu Li
Huawei Technologies Co., Ltd.
Bantian, Longgang district
Shenzhen 518129,
China

EMail: hongyu.lihongyu@huawei.com

Jim Guichard
Cisco Systems, Inc.
USA

EMail: jguichar@cisco.com

Paul Quinn
Cisco Systems, Inc.
USA

Email: paulq@cisco.com

Linda Dunbar
Huawei Technologies
5430 Legacy Drive, Suite #175
Plano TX
USA

EMail: linda.dunbar@huawei.com

7. Acknowledgements

Many thanks to K. Gray, N. Takaya, H. Kitada, H. Kojima, D. Dolson, B. Wright, and J. Halpern for their comments.

8. References

8.1. Normative References

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

8.2. Informative References

[I-D.ietf-sfc-problem-statement]
Quinn, P. and T. Nadeau, "Service Function Chaining Problem Statement", draft-ietf-sfc-problem-statement-11 (work in progress), February 2015.

[RFC2475] Blake, S., Black, D., Carlson, M., Davies, E., Wang, Z., and W. Weiss, "An Architecture for Differentiated Services", RFC 2475, December 1998.

[RFC2983] Black, D., "Differentiated Services and Tunnels", RFC 2983, October 2000.

[RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, December 2005.

[RFC5706] Harrington, D., "Guidelines for Considering Operations and Management of New Protocols and Protocol Extensions", RFC 5706, November 2009.

[RFC6040] Briscoe, B., "Tunnelling of Explicit Congestion Notification", RFC 6040, November 2010.

[RFC6291] Andersson, L., van Helvoort, H., Bonica, R., Romascanu, D., and S. Mansfield, "Guidelines for the Use of the "OAM" Acronym in the IETF", BCP 161, RFC 6291, June 2011.

[RFC7297] Boucadair, M., Jacquenet, C., and N. Wang, "IP Connectivity Provisioning Profile (CPP)", July 2014.

Authors' Addresses

Mohamed Boucadair
France Telecom
Rennes 35000
France

EMail: mohamed.boucadair@orange.com

Christian Jacquenet
France Telecom
Rennes 35000
France

EMail: christian.jacquenet@orange.com

Yuanlong Jiang
Huawei Technologies Co., Ltd.
Bantian, Longgang district
Shenzhen 518129,
China

EMail: jiangyuanlong@huawei.com

Ron Parker
Affirmed Networks
Acton, MA
USA

EMail: Ron_Parker@affirmednetworks.com

Kengo Naito
NTT
Midori-Cho 3-9-11
Musashino-shi, Tokyo 180-8585
Japan

EMail: naito.kengo@lab.ntt.co.jp