Service Function Chaining (sfc)   M. Boucadair, Ed.
Internet-Draft   Orange
Intended status: Informational   October 23, 2016
Expires: April 26, 2017

Service Function Chaining (SFC) Control Plane Components & Requirements

draft-ietf-sfc-control-plane-08

Abstract

This document describes requirements for conveying information between Service Function Chaining (SFC) control elements and SFC data plane functional elements. Also, this document identifies a set of control interfaces to interact with SFC-aware elements to establish, maintain or recover service function chains. This document does not specify protocols nor extensions to existing protocols.

This document exclusively focuses on SFC deployments that are under the responsibility of a single administrative entity. Inter-domain considerations are out of scope.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction 3
1.1. Scope 3
1.2. Terminology 4
1.3. Assumptions 5
2. Generic Considerations 6
2.1. Generic Requirements 6
2.2. SFC Control Plane Bootstrapping 6
2.3. SFC Dynamics 7
2.4. Coherent Setup of an SFC-enabled Domain 8
3. SFC Control Plane: Reference Architecture & Interfaces . . . 8
3.1. Reference Architecture 8
3.2. Centralized vs. Distributed 10
3.3. Interface Reference Points 11
3.3.1. C1: Interface between SFC Control Plane & SFC Classifier 11
3.3.2. C2: Interface between SFC Control Plane & SFF 13
3.3.3. C3: Interface between SFC Control Plane & SFC-aware SFs 14
3.3.4. C4: Interface between SFC Control Plane & SFC Proxy . 16
4. Additional Considerations 16
4.1. Discovery of the SFC Control Element 16
4.2. SF Symmetry 17
4.3. Pre-deploying SFCs 17
4.4. Withdraw a Service Function (SF) 17
4.5. SFC/SFP Operations 18
4.6. Unsolicited (Notification) Messages 18
4.7. Liveness Detection 18
4.8. Monitoring & Counters 19
4.9. Validity Lifetime 19
4.10. Considerations Specific to the Centralized Path Computation Model 20
4.10.1. Service Function Path Adjustment 20
4.10.2. Head End Initiated SFP Establishment 21
4.10.3. (Regional) Restoration of Service Functions 21
4.10.4. Fully Controlled SFF/SF Sequence for a SFP 22
5. Security Considerations 23
5.1. Secure Communications 23
5.2. Pervasive Monitoring 24
5.3. Privacy 24
5.4. Denial-of-Service (DoS) 24

5.5. Illegitimate Discovery of SFs and SFC Control Elements . 24
6. IANA Considerations 24
7. Acknowledgments 24
8. Contributors 25
9. References 27
9.1. Normative References 27
9.2. Informative References 27
Author's Address 29

1. Introduction

The dynamic enforcement of a service-derived forwarding policy for packets entering a network that supports advanced Service Functions (SFs) has become a key challenge for operators. Typically, many advanced Service Functions (e.g., Performance Enhancement Proxies ([RFC3135]), NATs [RFC3022][RFC6333][RFC6146], firewalls [I-D.ietf-opsawg-firewalls], etc.) are solicited for the delivery of value-added services, particularly to meet various service objectives such as IP address sharing, avoiding covert channels, detecting and protecting against ever increasing Denial-of-Service (DoS) attacks, etc.

Because of the proliferation of such advanced service functions together with complex service deployment constraints that demand more agile service delivery procedures, operators need to rationalize their service delivery logics and master their complexity while optimising service activation time cycles. The overall problem space is described in [RFC7498]. A more in-depth discussion on use cases can be found in [I-D.ietf-sfc-use-case-mobility] and [I-D.ietf-sfc-dc-use-cases].

[RFC7665] presents a model addressing the problematic aspects of existing service deployments, including topological dependence and configuration complexity. It also describes an architecture for the specification, creation, and ongoing maintenance of Service Function Chains (SFC) within a network. That is, how to define an ordered set of Service Functions and ordering constraints that must be applied to packets and/or frames and/or flows selected as a result of classification. [I-D.ietf-sfc-nsh] specifies the SFC encapsulation as per [RFC7665].

1.1. Scope

While [RFC7665] focuses on data plane considerations, this document describes requirements for conveying information between SFC control elements and SFC data plane functional elements. Also, this document identifies a set of control interfaces to interact with SFC-aware elements to establish, maintain or recover service function chains.

Both distributed and centralized control plane schemes to install SFC-related state and influence forwarding policies are discussed.

This document does not make any assumption on the deployment use cases. In particular, the document implicitly covers fixed, mobile, data center networks, and any combination thereof.

This document does not make any assumption about which control protocol to use, whether one or multiple control protocols are required, or whether the same or distinct control protocols will be invoked for each of the control interfaces. It is out of scope of this document to specify a profile for an existing protocol, to define protocol extensions, or to select a protocol.

Considerations related to the chaining of Service Functions (SFs) that span domains owned by multiple administrative entities are out of scope.

It is out of scope of this document to discuss SF-specific control and policy enforcement schemes; only SFC considerations are elaborated, regardless of the various connectivity services that may be supported in the SFC-enabled domain. Likewise, only the control of SFC-aware elements is discussed.

Service catalogue (including guidelines for deriving service function chains) is out of scope.

This document does not specify any flow exchange to illustrate the comprehensive SFC operation. Instead, it focuses on the required information to be conveyed via each control interface. Note that sketching a comprehensive flow exchange is also a function of deployment considerations that are out of scope.

1.2. Terminology

The reader should be familiar with the terms defined in [RFC7498] and [RFC7665].

The document makes use of the following terms:

- SFC data plane functional element: Refers to SFC-aware Service Function, Service Function Forwarder (SFF), SFC proxy, or classifier as defined in the SFC data plane architecture [RFC7665].

- SFC Control Element: A logical entity that instructs one or more SFC data plane functional elements on how to process packets within an SFC-enabled domain.

- SFC Classification rule: Refers to a rule maintained by a classifier that reflects the policies for binding an incoming flow/packet to a given SFC and Service Function Path (SFP). Actions are associated with matching criteria. The set of classification entries maintained by a classifier are referred to as in the classification policy table.

- SFP Forwarding Policy Table: this table reflects the SFP-specific traffic forwarding policy enforced by SFF components for every relevant incoming packet that is associated to one of the existing SFCs. The SFP Identifier (SFP-id) is used as a lookup key to determine forwarding action regardless of whether the SFC is fully constrained, partially constrained, or not constrained at all. Additional information such as a flow identifier, Service Index (SI), and/or other characteristics (e.g., the 5-tuple transport coordinates of the original packet) may be used for lookup purposes. The set of information to use for lookup purposes may be instructed by the control plane.

1.3. Assumptions

This document adheres to the assumptions listed in Section 1.2 of [RFC7665].

As a reminder, a Service Function Path (SFP) designates a subset of the collection designated by the SFC. For some SFPs, in some deployments, that will be a set of 1. For other SFPs (in the same or other deployments) it may be a larger set. For some SFPs in some deployments the SFP may designate the same set of choices as the SFC. This document accommodates all those deployments.

This document does not make any assumptions about the co-location of SFC data plane functional elements; this is deployment-specific. This document can accommodate a variety of deployment contexts such as (but not limited to):

- A Service Function Forwarder (SFF) can connect instances of the same or distinct SFs.
- An SF instance can be serviced by one or multiple SFFs.
- One or multiple SFs can be co-located with an SFF.
- A boundary node (that connects one SFC-enabled domain to a node either located in another SFC-enabled domain or in a domain that is SFC-unaware) can act as an egress node and an ingress node for the same flow.
- Distinct ingress and egress nodes may be crossed by a packet when forwarded in an SFC-enabled domain.
- Distinct ingress nodes may be solicited for each traffic direction (e.g., upstream and downstream).

- o The same boundary node may act as an ingress node, an egress node, and also embed a classifier.
- o A classifier can be hosted in a node that embeds one or more SFs.
- o Many network elements within an SFC-enabled domain may behave as egress/ingress nodes.

Furthermore, the following assumptions are made:

- o A Control Element can be co-located with a classifier, SFF or SF.
- o One or multiple Control Elements can be deployed in an SFC-enabled domain.
- o State synchronization between Control Elements is out of scope.

2. Generic Considerations

2.1. Generic Requirements

Some deployments require that forwarding within an SFC-enabled domain must be allowed even if no control protocols are enabled. Static configuration must be allowed.

A permanent association between an SFC data plane element with a Control Element must not be required; specifically, the SFC-enabled domain must keep on processing incoming packets according to the SFC instructions even during temporary unavailability events of control plane components. SFC implementations that do not meet this requirement will suffer from another flavor of the constrained high availability issue, discussed in Section 2.3 of [RFC7498], supposed to be solved by SFC designs.

2.2. SFC Control Plane Bootstrapping

The interface that is used to feed the SFC control plane with service objectives and guidelines is not part of the SFC control plane itself. Therefore, this document assumes the SFC control plane is provided with a set of required information for proper SFC operation with no specific assumption about how this information is collected/ provisioned, nor about the structure of such information. The following information that is recommended to be provided to the SFC control plane prior to bootstrapping includes:

- o Locators for classifiers/SFF/SFs/SFC proxies, etc.
- o SFs serviced by each SFF.
- o A list of service function chains, including how they are structured and unambiguously identified.
- o Status of each SFC: active/pre-deployment phase/etc. An SFC can be defined at the management level and instantiated in an SFC-enabled domain for pre-deployment purposes (e.g., testing).

Actions to activate, modify or withdraw an SFC are triggered by the control plane. Nevertheless, this document does not make any assumption about how an operator instructs the control plane.

- A list of classification guidelines and/or rules to bind flows to SFCs/SFPs.
- Security credentials.
- Context information that needs to be shared on a per SFC basis.

Optionally, load balancing objectives at the SFC level or on a per node (e.g., per-SF/SFF/SFC proxy) basis may also be provided to the SFC control plane. Likewise, the set of metadata that is supported by SFC-aware SFs, SFFs, and SFC proxies may be provided to the SFC control plane.

Also, the SFC control plane may gather the following information from an SFC-enabled domain at bootstrapping (non-exhaustive list). How this information is collected is left unspecified in this document:

- The list of active SFC-aware SFs (including their locators).
- The list of SFFs and the SFs that are attached to.
- The list of enabled SFC proxies, and the list of SFC-unaware SFs attached to.
- The list of active SFCs/SFPs as enabled in an SFC-enabled domain.
- The list of classifiers and their locators, so as to retrieve the classification policy table for each classifier, in particular.
- The SFP Forwarding Policy Tables maintained by SFFs.
- The set of metadata that is supported by SFC-aware SFs, SFFs, and SFC proxies. Additional capabilities (e.g., supported transport encapsulation scheme(s), supported SFC header version(s)) may also be collected.

During the bootstrapping phase, a Control Element may detect a conflict between the running configuration in an SFC data plane element and the information maintained by the control plane. Consequently, the control plane undertakes appropriate actions to fix those conflicts. This is typically achieved by invoking one of the interfaces defined in Section 3.3.

After bootstrapping, the SFC control plane is fed (dynamically or on a per request basis) with a set of information that is required for proper SFC operation. More details about this information are discussed in Section 3 and Section 4.

2.3. SFC Dynamics

By default, SFC data and control plane elements must assume that SFC control information are dynamic by nature. This requirement applies even for policies that are communicated via an upper layer to

communicate service objectives and guidelines to a control element. Additionally, the SFC control plane must not assume that the capabilities of SFC data plane elements are frozen. The SFC control architecture must be designed to accommodate any dynamic of SFs/SFFs attachments, software updates, dynamic network condition events, etc.

The overall SFC orchestration is not discussed in this document because SFC operations are likely to be policy-driven. Nevertheless, the document specifies required interfaces that can be invoked in the context of an SFC orchestration fed with policies that are local to an SFC-enabled domain. No assumption is made about those policies nor their change dynamics. The control interfaces are designed to cover both dynamic control information exchange, but also to issue request solicitations to the appropriate SFC data plane elements.

2.4. Coherent Setup of an SFC-enabled Domain

Various transport encapsulation schemes and/or versions of SFC header implementations may be supported by one or several nodes of an SFC-enabled domain. For the sake of coherent configuration, the SFC control plane is responsible for instructing all the involved SFC data plane functional elements about the behavior to adopt to select the transport encapsulation scheme(s), the version of the SFC header to enable, etc.

3. SFC Control Plane: Reference Architecture & Interfaces

3.1. Reference Architecture

The SFC control plane is responsible for the following:

- o Build and monitor the service-aware topology. For example, this can be achieved by means of dynamic SF discovery techniques. Those means are out of scope of this document.
- o Maintain a repository of service function chains, SFC matching criteria to bind flows to a given service function chain, and mapping between service function chains and SFPs.
- o Guarantee the coherency of the configuration and the operation of an SFC-enabled domain.
- o Dynamically compute a service forwarding path (distributed model, see Section 3.2).
- o Determine a forwarding path in the context of a centralized deployment model (see Section 3.2).
- o Update service function chains or adjust SFPs (e.g., for restoration purposes) based on various inputs (e.g., external policy context, path alteration, SF unavailability, SF withdrawal, service decommissioning, etc.).

o Provision SFP Forwarding Policy Tables of involved SFFs and
   provide classifiers with traffic classification rules.

```
                  +----------------------------------------------+
                  |                                              |
                  |               SFC  Control Plane             |
          +-------|                                              |
          |       |                                              |
          C1      +------^-----------^-------------^-------------+
   +---------------------|C3---------|-------------|-------------+
   |      |            +----+        |             |             |
   |      |            | SF |        |C2           |C2           |
   |      |            +----+        |             |             |
   | +----V--- --+       |           |             |             |
   | |   SFC     |     +----+      +-|--+        +----+          |
   | |Classifier |---->|SFF |----->|SFF |------->|SFF |          |
   | |   Node    |<----|    |<-----|    |<-------|    |          |
   | +-----------+     +----+      +----+        +----+          |
   |                     |           |              |            |
   |                     |C2      -------           |            |
   |                     |       |       |     +-----------+ C4  |
   |                     V     +----+ +----+   | SFC Proxy |-->  |
   |                           | SF | |SF  |   +-----------+     |
   |                           +----+ +----+                     |
   |                             |C3    |C3                      |
   |  SFC Data Plane Components  V      V                        |
   +-------------------------------------------------------------+
```

Figure 1: SFC Control Plane Interfaces

Figure 1 shows the overall SFC control plane architecture, including interface reference points. Particularly, Figure 1 shows the various interfaces that are required for conveying control information between the SFC control plane and underlying SFC data plane elements:

1. Interface between SFC Control Plane & SFC classifier (C1): This interface is used to manage SFC classification rules in classifiers. These rules can be added, modified, or deleted. Additional information is provided in Section 3.3.1. In order to avoid stale classification rules and to allow for local sanity checks, a validity lifetime is associated with each classification rule (Section 4.9).

2. Interface between SFC Control Plane & SFF (C2): This interface is used to communicate with an SFF for various purposes (e.g., communicate required information for SFC forwarding decision-making, collect state information to adjust SFPs, collected connected SFs, etc.). Section 3.3.2 specifies such interface.

3. Interface between SFC Control Plane & SFC-aware SFs (C3): The SFC control plane uses this interface to interact with SFC-aware SFs. This interaction may be direct or via dedicated SF management systems (Section 3.3.3).

4. Interface between SFC Control Plane & SFC proxy (C4): The SFC control plane uses this interface to interact with an SFC proxy to communicate SFC instructions and to retrieve state information required, e.g., for dynamic SFP adjustments (Section 3.3.4).

This document does not elaborate on the internal decomposition of the SFC control plane functional blocks. The components within the SFC control plane and their interactions are out of scope.

Note, the SFC control plane must be able to invoke SFC OAM mechanisms, and to determine the results of OAM operations.

3.2. Centralized vs. Distributed

The SFC control plane can be (logically) centralized, distributed or a combination thereof. Whether one or multiple SFC Control Elements are enabled is deployment-specific. Nevertheless, the following comments can be made:

SFC management (including SFC monitoring and supervision): is likely to be centralized.

SFC mapping rules: i.e., service instructions to bind a flow to a service function chain and SFP are likely to be managed by a central SFC Control Element, but the resulting policies can be shared among several Control Elements. Note, these policies can be complemented with local information (e.g., an IPv4 address/IPv6 prefix assigned to a customer) because such information may not be available to the central entity but known only during network attachment phase.

Path computation: can be either distributed or centralized. Distributed path computation means that the selection of the exact sequence of SFs that a packet needs to invoke (along with instances and/or SFF locator information) is a result of a distributed path selection algorithm executed by involved nodes. For some traffic engineering proposes, the SFP may be constrained by the control plane; as such, some SFPs can be fully specified (i.e., list all the SFF/SFs that need to be solicited) or partially specified (e.g., exclude some nodes, explicitly select which instance of a given SF needs to be invoked, etc.).

SFP resiliency (including restoration) refers to mechanisms to ensure high available service function chains. It includes means to detect node/link/path failures. Both centralized and distributed mechanism to ensure SFP resiliency can be envisaged.

Implementing a (logically) centralized path computation engine requires information to be dynamically communicated to the central SFC Control Element, such as the list of available SF instances, SFF locators, load status, SFP availability, etc.

3.3. Interface Reference Points

The following sub-sections describe the interfaces between the SFC control plane, as well as various SFC data plane elements.

3.3.1. C1: Interface between SFC Control Plane & SFC Classifier

As a reminder, a classifier is a function that is responsible for classifying traffic based on (pre-defined) rules.

This interface is used to install SFC classification rules in classifiers. Once classification rules are populated, classifiers are responsible for binding incoming traffic to service function chains and SFPs according to these classification rules. Note, the SFC control plane must not make any assumption on how the traffic is to be bound to a given service function chain. In other words, classification rules are deployment-specific. 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. Installing classification rules must be immediate. The status of enforcing such rules must be communicated to the control plane as part of the communication procedure. In particular, specific error codes must be returned to the Control Element in case an error is encountered during the enforcement procedure.

The SFC control plane should be responsible for removing invalid (and stale) mappings from the classification tables maintained by the classifiers. Also, local sanity checks mechanisms may be supported locally by the classifiers, but those are out of scope.

The classifier may be notified (regularly or upon eventual change) by the control plane about the available SFs (including the SFFs they are attached to) or be part of the service function discovery procedure.

Classification rules may be updated, deleted or disabled by the control plane. Criteria that would trigger those operations are deployment-specific.

This interface is also used to retrieve the list of classification rules that are maintained by a classifier. This retrieval can be on demand (at the initiative of the Control Element) or on a regular basis (at the initiative of the classifier).

Given that service function chaining solutions may be applied to very large sets of traffic, any control solution should take scaling issues into consideration as part of the design. For example, because a large number (e.g., 1000s) of classification entries may be configured to a classifier, means to reduce classification lookup time such as optimizing the size of the classification table (e.g., by means of aggregation capabilities) should be supported by the SFC control plane (and/or the classifier).

Below are listed some functional objectives that can be achieved thanks to the invocation of this interface:

- o Rationalize the management of classification rules.
- o Maintain a global view of instantiated rules in all classifiers in an SFC-enabled domain.
- o Check the consistency of instantiated classification rules within the same classifier or among multiple classifiers.
- o Assess the impact of removing or modifying a classification rule on packets entering an SFC-enabled domain.
- o Aggregate classification rules for the sake of performance optimization (mainly reduce lookup delays).
- o Adjust classification rules when rules are based on volatile identifiers (e.g., an IPv4 address, IPv6 prefix).
- o Allow to rapidly restore SFC/SFP states during failure events that occurred at a classifier (or a Control Element).

The control plane must instruct the classifier about the initial values of the Service Index (SI).

Also, the control plane must instruct the classifier about the set of metadata to be supplied in the context of a given chain.

SFC encapsulation protocol [I-D.ietf-sfc-nsh] includes metadata type 1 (MD#1) with mandatory context headers that can be used to convey metadata along an SFP. [I-D.ietf-sfc-nsh] allows defining different semantics in the context headers, but the NSH header does not convey that semantics in the context header. [I-D.ietf-sfc-nsh] requires an SFC-aware SF using the data placed in the MD#1 mandatory context headers to use information external to the NSH data plane to

understand the semantics of the context data. Therefore, this interface must provide such context semantics, including any suitable scoping information.

The control plane must instruct the classifier whether it can trust an existing SFC information carried in an incoming packet or whether it must be ignored.

A classifier should send unsolicited messages through this interface to notify the SFC control plane about specific events. Triggers for sending unsolicited messages should be configurable parameter.

When re-classification is allowed in an SFC-enabled domain, this interface can be used to control classifiers co-resident with SFC-aware SFs, SFC proxies, or SFFs to manage re-classification rules.

When an incoming packet matches more than one classification rule, tie-breaking criteria should be followed (e.g., priority). Such tie-breaking criteria should be instructed by the control plane.

The identification of instantiated SFCs/SFPs is local to each administrative domain; it is policy-based and deployment-specific.

3.3.2. C2: Interface between SFC Control Plane & SFF

SFFs make traffic forwarding decisions according to the entries maintained in their SFP Forwarding Policy Table. Such table is populated by the SFC control plane through the C2 interface. In particular, this interface is used to instruct the SFF about the set of information to use for lookup purposes (e.g., SFP-id, 5-tuple transport coordinates). One or many entries may be installed using one single control message. Installing new entries in the SFP Forwarding Policy Table must be immediate. The status of enforcing such entries must be communicated to the control plane as part of the communication procedure. In particular, specific error codes must be returned to the Control Element in case an error is encountered during the enforcement procedure.

This interface is used to instruct an SFF about the SFC-aware SFs that it can service. Such instruct typically occurs at the bootstrapping of the SFF, in the event of a new SF is added to the SFC-enabled domain, etc.

This interface is also used by the SFF to report the connectivity to their attached (including embedded) SFs. Local means may be enabled between the SFC-aware SFs and SFFs to allow for the dynamic attachment of SFs to an SFF and/or discovery of SFs by an SFF but those means are unspecified in this document.

The C2 interface is also used for collecting states of attributes (e.g., availability, workload, latency), for example, to dynamically adjust Service Function Paths. Such state can be collected using an explicit request from a Control Element or by unsolicited notification of the SFF on a regular basis or when an event occurs. A configuration parameter should be supported by the SFF to instruct the exact behavior to follow.

The C2 interface may be used to configure groups of functionally equivalent SFs. In particular, this group may be used for load-balancing purposes.

An SFF must be instructed to strip the SFC information for the chains it terminates. Forwarding policies for handling packets bound to chains that are terminated by an SFF may be communicated via this interface. By default, an SFF relies on legacy processing for forwarding these packets.

3.3.3. C3: Interface between SFC Control Plane & SFC-aware SFs

SFs may need to output some processing results of packets to the SFC control plane. This information can be used by the SFC control plane to update the SFC classification rules and the SFP Forwarding Policy Table entries.

This interface is used to collect such kind of feedback information from SFs. For example, the following information can be exchanged between an SF and the SFC control plane:

- SF execution status: Some SFs may need to send information to the control plane to fine tune SFPs. For example, a threat-detecting SF can periodically send the threat characteristics via this interface, such as high probability of threat with packet of a given size. The control plane can then add an appropriate matching criteria to SFF to steer traffic to a scrubbing center.

- SF load update: When SFs are under stress that yielded the crossing of some performance thresholds, the SFC control plane needs to be notified to adjust SFPs accordingly (especially when the centralized path computation mode is enabled). It is out of scope of this document to specify the exact methods to monitor the performance threshold or stress level of SFs, nevertheless the SFC control plane can invoke those methods for its operations.

- SF bypass: An SF may use this interface to notify the Control Plane about its desire to be bypassed. The exact details about SF bypass logic are out of scope of this document.

The SFC control needs the above status information for various tasks it undertakes, but this information may be acquired directly from SFs or indirectly from other management and control systems in the operational environment.

This interface is used by an SFC-aware SF to report the set of context information (a.k.a., metadata) that it supports and any change of its capabilities, for example, as a result of a software update. Such change notifications should be dynamic, by default. A configuration parameter may be supported to disable such behavior.

This interface is also used to instruct an SFC-aware SF about any metadata it needs to attach to packets for a given SFC. This instruction may occur any time during the validity lifetime of an SFC/SFP.

Also, this interface informs the SFC-aware SF about the semantics of a context information, which would otherwise have opaque meaning. Several attributes may be associated with a context information such as (but not limited to) the "scope" (e.g., per-packet, per-flow, or per host), whether it is "mandatory" or "optional" to process flows bound to a given chain, etc. Note that a context may be mandatory for "chain 1", but optional for "chain 2". In particular, this interface must provide NSH MD#1 mandatory context semantics, including any suitable scoping information.

The control plane may indicate, for a given service function chain, an order for consuming a set of contexts supplied in a packet. This order may be indicated any time during the validity lifetime of an SFC/SFP.

An SFC-aware SF can also be instructed about the behavior it should adopt after consuming a context information that was supplied in the SFC header. For example, the context can be maintained, updated, or stripped.

Multiple SFs may be located within the same physical node, but no SFF is enabled in that same node, means to unambiguously forward the traffic from the SFF to the appropriate SF must be supported. Concretely, each SF must have a unique locator for unambiguous forwarding. This locator may be configured using this interface.

The controller may use the C3 interface to specify how the reverse path of flows, that are processed for a given direction, is selected by the SF. This feature is useful, for example, for packets generated by an SFC-aware SF to ensure these packets are forwarded to the corresponding source node with the same set of SFs, involved in the forward path, are invoked in the reverse order when forwarding

back these packets. Special care should be considered to avoid that instructions provided to distinct SFs lead to loops. Additional considerations are discussed in Section 4.2.

3.3.4. C4: Interface between SFC Control Plane & SFC Proxy

This interface is used by an SFC proxy to report the set of context information (a.k.a., metadata) that it supports and any change of its capabilities that may result, for example, in a software update. Such change notifications should be dynamic, by default. A configuration parameter may be supported to disable such behavior.

The SFC proxy can be instructed about authorized SFC-unaware SFs it can service. This instruction may occur during the bootstrapping of the SFC proxy or anytime during the SFP proxy operation time.

An SFC proxy may be instructed about the behavior it should adopt to process the context information that was supplied in the SFC header on behalf of an SFC-unaware SF, e.g., the context can be maintained or stripped.

The SFC proxy is also instructed about the semantics of a context information (including MD#1), which would otherwise have opaque meaning. Several attributes may be associated with a context information such as (but not limited to) the "scope" (e.g., per-packet, per-flow or per host), whether it is "mandatory" or "optional" to process flows bound to a given chain, etc.

The SFC proxy may also be instructed to add some new context information into the SFC header on behalf of an SFC-unaware SF.

The C4 interface is also used for collecting attribute states (e.g., availability, workload, latency), for example, to dynamically adjust Service Function Paths.

This interface may also be used to instruct the SFC proxy about the state and information to maintain for proper handling of packets received back from an SFC-unaware SF.

4. Additional Considerations

4.1. Discovery of the SFC Control Element

SFC data plane functional elements need to be provisioned with the locators of the Control Elements. This can be achieved using a variety if mechanisms such as static configuration or the activation of a service discovery mechanism. The exact specification of how this provisioning is achieved is out of scope.

4.2. SF Symmetry

Some SFs require both directions of a flow to traverse. Some service function chains require full symmetry. If an SF (e.g., stateful firewall or NAT) needs both direction of a flow, it is the SF instantiation that needs both direction of a flow to traverse, not the abstract SF (which can have many instantiations spread across the network).

Typically:

- C1 interface is used to instruct the classifier how both directions of a flow should be processed when crossing an SFC-enabled domain.

- C2 interface may be used to ensure that the same SF instance is involved in both directions of a flow (including, to ensure full chain symmetry).

4.3. Pre-deploying SFCs

Enabling service function chains should preserve some deployment practices adopted by Operators. Particularly, installing a service function chain (and its associated SFPs) should allow for pre-deployment testing and validation purposes (that is a restricted and controlled usage of such service function chain (and associated SFPs)).

4.4. Withdraw a Service Function (SF)

During the lifetime of an SFC, a given SF can be decommissioned. To accommodate such context and any other case where an SF is to be withdrawn, the control plane should instruct the SFC data plane functional element about the behavior to adopt. For example:

1. a first approach would be to update the service function chains and/or associated SFPs where that SF is present by removing any reference to that SF. The update concerns service function chains if the decommissioned SF is not provided by any active node. SFPs are impacted when alternate SF instances can provide the same service of the decommissioned SF instance.

2. a second approach would be to delete/deactivate any service function chain (and its associated SFPs) that involves that SF but install new service function chains.

4.5. SFC/SFP Operations

Various actions can be executed on a service function chain (and associated SFPs) that is structured by the SFC control plane. Indeed, a service function chain (and associated SFPs) can be enabled, disabled, its structure modified by adding a new SF hop or remove an SF from the sequence of SFs to be invoked, its classification rules modified, etc.

A modification of a service function chain can trigger control messages with the appropriate SFC-aware nodes accordingly.

The approach to be followed to migrate traffic to a new SFP from an old SFP is deployment-specific. For example, in order to avoid service disruption, a make-before-break mechanism can be followed where a new SFP is allocated to replace an existing SFP. Once the new SFP is set up, tested and the traffic is migrated to it, the old SFP can be removed. Other strategies may be followed within an SFC-enabled domain.

4.6. Unsolicited (Notification) Messages

SFC data plane functional elements must be instructed to send unsolicited notifications when loops are detected, a problem in the structure of a service function chain is encountered, a long unavailable forwarding path time is observed, etc.

Specific criteria to send unsolicited notifications to a Control Element should be fine tuned by the control plane using the interface defined in Section 3.3.

4.7. Liveness Detection

The control plane must allow to detect the liveliness of SFC data plane elements of an SFC-enabled domain. Note that a data element may responsive from a connectivity standpoint, but the service it is supposed to provide may not be available.

In particular, the control plane must allow to dynamically detect that an SF instance is out of service and notify the relevant Control Element accordingly. The liveness information may be acquired directly from SFs or indirectly from other management and control systems in the operational environment.

Liveness status records for all SF instances, and service function chains (including the SFPs bound to a given chain) are maintained by the SFC Control.

The classifier may be notified by the control plane or be part of the liveness detection procedure.

The ability of an SFC Control Element to check the liveness of each SF present in service function chain has several advantages, including:

- o Enhanced status reporting by the control plane (i.e., an operational status for any given service chain derived from liveness state of its SFs).
- o Ability to support various resiliency policies (i.e., bypass a node embedding an SF, use alternate node, use alternate chain, drop traffic, etc.) .
- o Ability to support load balancing capabilities to solicit multiple SF instances that provide equivalent functions.

Local failure detect and repair mechanisms may be enabled by SFC-aware nodes. Control Elements may be fed directly or indirectly with inputs from these mechanisms.

Because a node embedding an SF can be responsive from a reachability standpoint (e.g., IP level) while the function it provides may be broken (e.g., a NAT module may be down), additional means to assess whether an SF is up and running are required. These means may be service-specific.

4.8. Monitoring & Counters

SFC-specific counters and statistics must be provided using the interfaces defined in Section 3.3. These data include (but not limited to):

- o Number of flows ever and currently assigned to a given service function chain and a given SFP.
- o Number of flows, packets, bytes dropped due to policy.
- o Number of packets and bytes in/out per service function chain and SFP.
- o Number of flows, packets, bytes dropped due to unknown service function chain (this is valid in particular for an SF node).

Even if setting the data collection cycle is deployment-specific, it is recommended to support dynamic means for better SFC automation.

4.9. Validity Lifetime

SFC instructions communicated via the various interfaces introduced in Section 3.3 may be associated with validity lifetimes, in which case classification and SFP Forwarding Policy Table entries will be

automatically removed upon the expiry of the validity lifetime without requiring an explicit action from a Control Element.

Lifetimes are used in particular by an SFC data plane element to clear invalid control entries that would be maintained in the system if, for some reason, no appropriate action was undertaken by the control plane to clear such entries.

Both short and long lifetimes may be assigned.

4.10. Considerations Specific to the Centralized Path Computation Model

This section focuses on issues that are specific to the centralized deployment model (Section 3.2).

4.10.1. Service Function Path Adjustment

An SFP is determined by composing SF instances and overlay links among SFFs. Thus, the status of an SFP depends on the states or attributes (e.g., availability, topological location, latency, workload, etc.) of its components. For example, failure of a single SF instance results in failure of the whole SFP. Since these states or attributes of SFP components may vary in time, their changes should monitored and SFPs should be dynamically adjusted.

Examples of use cases for SFP adjustment are listed below:

SFP fail-over: re-construct an SFP with replacing the failed SF instance with another instance of the same SF or withdraw the failed SF from being invoked. Note that withdrawing an SF may be envisaged if the resulting connectivity service is not broken (that is, packets bound to the updated SFP can be successfully delivered to their ultimate destinations). Rerouting the traffic to another SF instance or withdrawing the failed SF is deployment-specific.

SFP with better latency experience: re-construct an SFP with a low path stretch considering the changes in topological locations of SF instances and the latency induced by the (overlay) connectivity among SFFs.

Traffic engineered SFP: re-construct SFPs to localize the traffic in the network considering various TE goals such as bypass a node, bypass a link, etc. These techniques may be used for planned maintenance operations on an SFC-enabled domain.

SF/SFP Load-balancing: re-construct SFPs to distribute the workload among various SF instances. Particularly, load distribution

policies can be taken into account by the Control Element to re-compute an SFP or be provisioned as attributes to SFPs that will be installed using the control interfaces (C2 interface, typically).

For more details about the use cases, refer to [I-D.lee-nfvrg-resource-management-service-chain].

The procedures for SFP adjustment may be handled by the SFC control plane as follows:

- o Collect and monitor states and attributes of SF instances and overlay links via the C2 interface (Section 3.3.2) and the C3 interface (Section 3.3.3).

- o Evaluate SF instances and overlay links based on the monitoring results.

- o Select SF instances to re-determine an SFP according to the evaluation results.

- o Replace target SF instances (e.g., in a failure or overladed) with newly selected ones.

- o Enforce the updated SFP for upcoming SFC traversal to SFFs via the C1 interface (Section 3.3.1) or the C2 interface (Section 3.3.2).

4.10.2. Head End Initiated SFP Establishment

In some scenarios where an SFC Control Element is not connected to all SFFs in an SFC-enabled domain, the SFC control plane can send the explicit SFF/SF sequence or SF sequence to the SFC head-end, e.g., the classifier via the C1 interface (Section 3.3.1). SFC head-end can use a signaling protocol to establish the SFF/SF sequence based on the SF sequence. Additional information (e.g., SF/SFF load) may be communicated to the SFC head-end to adjust an SFP.

4.10.3. (Regional) Restoration of Service Functions

There are situations that it might not be feasible for the classifier to be notified of the changes of SFF-sequence or SFF/SF Sequence for a given SFP because of the time taken for the notification and the limited capability of the classifiers.

If an SF has a large number of instantiations, it scales better if the classifier doesn't need to be notified with status of visible instantiations of SFs on an SFP.

It might not be always feasible for the classifier to be aware of the exact SF instances selected for a given SFP due to too many instances for each SF, notifications not being promptly sent to the classifier, or other reasons. This is about multiple instances of the same SF attached to one SFF node; those instances can be handled by the SFF via local load balancing schemes.

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 SFFs or network nodes. Typically, the regional ingress node should be:

o on the data path of the flow of the given SFC;
o in front of the relevant SFFs or network nodes that are impacted by the change of the SFP;
o capable of encoding the detailed SFP to the Service Chain Header of data packets of the identified flow; and
o capable of removing the detailed SFP encoding in data packets after all the impacted SFFs and network nodes completed the policy installation.

4.10.4. Fully Controlled SFF/SF Sequence for a SFP

This section discusses some information that can be exchanged over C2 interface (Section 3.3.2) when the SFC Control Element explicitly passes the steering policies to all SFFs for the SFF/SF sequence of a given SFC. In this model, each SFF doesn't need to signal other SFFs for the SFP.

The SFF nodes are not required to be directly adjacent to each other. As such, they can be interconnected using an overlay technique, such as Generic Routing Encapsulation (GRE), Virtual eXtensible Local Area Network (VXLAN), etc. SFs are attached to an SFF node or SFC proxy node via Ethernet link or other link types. As a local decision, there may be multiple different steering policies that work in conjunction with the SFC encapsulation [I-D.ietf-sfc-nsh] for one flow within one SFF.

For example, the semantics of traffic steering rules can be a match condition and an action, similar to, e.g., the route described in Section 2.3 of [I-D.ietf-i2rs-rib-info-model]. The match conditions and action for distinct ports can be different.

The matching criteria for SFF can be more sophisticated. For example, it could be the SFP-id carried within the SFC encapsulation with any fields in the data packets, such as (non-exhaustive list):

- o Destination MAC address
- o Source MAC address
- o VLAN-ID,
- o Destination IP address
- o Source IP address
- o Source port number
- o Destination port number
- o Differentiated Services Code Point (DSCP)
- o Packet size, etc., or any combination thereof.

An SFF node may not support some of the matching criteria listed above. It is important that SFC control plane can retrieve the supported matching criteria by SFF nodes. The actions for traffic steering could be to steer traffic to the attached SF instances via a specific port.

The actions to SFC proxy may include a method to map the SFP identifier carried in the packet header to a locally significant link identifier, e.g., VLAN-ID, and a method to construct and encapsulate the SFC header back to the packets when they come back from the attached SFs.

This approach does not require using an end-to-end signaling protocol among classifier nodes and SFF nodes. However, there may be problems encountered if SFF nodes are not updated in the proper order or not at the same time. For example, if the SFF "A" and SFF "C" get flow steering policies at slightly different times, some packets might not be directed to some service functions on a chain.

5. Security Considerations

5.1. Secure Communications

The SFC Control Elements and the participating SFC data plane elements must mutually authenticate. SFC data plane elements must ignore instructions received from unauthenticated SFC Control Elements. The credentials details used during authentication can be used by the SFC control plane to decide whether specific authorization may be granted to a Service Function with regards to some specific operations (e.g., authorize a given SF to access specific context information).

In case multiple SFC data plane elements are embedded in the same node, the authentication mechanism may be executed as a whole; not for each instance.

An SFC data plane element must be able to send authenticated unsolicited notifications to an SFC Control Element.

The communication between a Control Element and SFC data plane elements must provide integrity and replay protection.

A Service Function must by default discard any action from an SFC Control Element that requires specific right privileges (e.g., access to a legal intercept log, mirror the traffic, etc.).

5.2. Pervasive Monitoring

The authentication mechanism should be immune to pervasive monitoring [RFC7258]. An attacker can intercept traffic by installing classification rules that would lead to redirect all or part of the traffic to an illegitimate network node. Means to protect against attacks that would lead to install, remove, or modify classification rules must be supported.

5.3. Privacy

The SFC control plane must be able to instruct SFC data plane elements about the information to be leaked outside an SFC-enabled domain. Particularly, the SFC control plane must support means to preserve privacy [RFC6973]. Context headers may indeed reveal privacy information (e.g., IMSI, user name, user profile, location, etc.). Those headers must not be exposed outside the operator's domain.

5.4. Denial-of-Service (DoS)

In order to protect against denial of service that would be caused by a misbehaving trusted SFC Control Element, SFC data plane elements should rate limit the messages received from an SFC Control Element.

5.5. Illegitimate Discovery of SFs and SFC Control Elements

Means to defend against soliciting illegitimate SFs/SFFs that do not belong to the SFC-enabled domain must be enabled. Such means must be defined in service function discovery and SFC Control Element discovery specification documents.

6. IANA Considerations

This document does not require any IANA actions.

7. Acknowledgments

This document is the result of merging with [I-D.lee-sfc-dynamic-instantiation].

Hongyu Li, Qin Wu, and Yong(Oliver) Huang edited an early version of the individual submission of this document.

Many thanks to Shibi Huang, Lac Chidung, Taeho Kang, Sumandra Majee, Dave Dolson, Paul Bottorff, Reinaldo Penno, Jim Guichard, Shunsuke Homma, Ken Gray, Henry Fourie, and Dirk von Hugo for the feedback and discussion on the mailing list.

The text about the semantic of a context information is provided by Dave Dolson and Lucy Yong.

Many thanks to Paul Quinn and Uri Elzur for the detailed review.

Thanks to Catherine Meadows for the SecDir review, and to Stephen Farrell and Tero Kivinen for scheduling an early SecDir review.

Special thanks to Alia Atlas for the careful AD review.

8. Contributors

The following individuals have contributed significantly to this document:

Hongyu Li
Huawei
Huawei Industrial Base,Bantian,Longgang
Shenzhen
China

EMail: hongyu.li@huawei.com

Qin Wu
Huawei
101 Software Avenue, Yuhua District
Nanjing, Jiangsu 210012
China

EMail: bill.wu@huawei.com

Yong(Oliver) Huang
Huawei
Huawei Industrial Base,Bantian,Longgang
Shenzhen
China

EMail: oliver.huang@huawei.com

Christian Jacquenet
Orange
Rennes 35000
France

EMail: christian.jacquenet@orange.com

Walter Haeffner
Vodafone D2 GmbH
Ferdinand-Braun-Platz 1
Duesseldorf 40549
DE

EMail: walter.haeffner@vodafone.com

Seungik Lee
ETRI
218 Gajeong-ro Yuseung-Gu
Daejeon 305-700
Korea

Phone: +82 42 860 1483
EMail: seungiklee@etri.re.kr

Ron Parker
Affirmed Networks
Acton
MA 01720
USA

EMail: ron_parker@affirmednetworks.com

Linda Dunbar
Huawei Technologies
USA

EMail: ldunbar@huawei.com

Andrew Malis
Huawei Technologies
USA

EMail: agmalis@gmail.com

Joel M. Halpern
Ericsson

EMail: joel.halpern@ericsson.com

Tirumaleswar Reddy
Cisco Systems, Inc.
Cessna Business Park, Varthur Hobli
Sarjapur Marathalli Outer Ring Road
Bangalore, Karnataka 560103
India

EMail: tireddy@cisco.com

Prashanth Patil
Cisco Systems, Inc.
Bangalore
India

EMail: praspati@cisco.com

9. References

9.1. Normative References

[RFC7665] Halpern, J., Ed. and C. Pignataro, Ed., "Service Function Chaining (SFC) Architecture", RFC 7665, DOI 10.17487/RFC7665, October 2015, <http://www.rfc-editor.org/info/rfc7665>.

9.2. Informative References

[I-D.ietf-i2rs-rib-info-model]
Bahadur, N., Kini, S., and J. Medved, "Routing Information Base Info Model", draft-ietf-i2rs-rib-info-model-09 (work in progress), July 2016.

[I-D.ietf-opsawg-firewalls]
Baker, F. and P. Hoffman, "On Firewalls in Internet Security", draft-ietf-opsawg-firewalls-01 (work in progress), October 2012.

[I-D.ietf-sfc-dc-use-cases]
Surendra, S., Tufail, M., Majee, S., Captari, C., and S. Homma, "Service Function Chaining Use Cases In Data Centers", draft-ietf-sfc-dc-use-cases-05 (work in progress), August 2016.

[I-D.ietf-sfc-nsh]
Quinn, P. and U. Elzur, "Network Service Header", draft-ietf-sfc-nsh-10 (work in progress), September 2016.

[I-D.ietf-sfc-use-case-mobility]
Haeffner, W., Napper, J., Stiemerling, M., Lopez, D., and J. Uttaro, "Service Function Chaining Use Cases in Mobile Networks", draft-ietf-sfc-use-case-mobility-07 (work in progress), October 2016.

[I-D.lee-nfvrg-resource-management-service-chain]
Lee, S., Pack, S., Shin, M., and E. Paik, "Resource Management in Service Chaining", draft-lee-nfvrg-resource-management-service-chain-01 (work in progress), March 2015.

[I-D.lee-sfc-dynamic-instantiation]
Lee, S., Pack, S., Shin, M., and E. Paik, "SFC dynamic instantiation", draft-lee-sfc-dynamic-instantiation-01 (work in progress), October 2014.

[RFC3022] Srisuresh, P. and K. Egevang, "Traditional IP Network Address Translator (Traditional NAT)", RFC 3022, DOI 10.17487/RFC3022, January 2001, <http://www.rfc-editor.org/info/rfc3022>.

[RFC3135] Border, J., Kojo, M., Griner, J., Montenegro, G., and Z. Shelby, "Performance Enhancing Proxies Intended to Mitigate Link-Related Degradations", RFC 3135, DOI 10.17487/RFC3135, June 2001, <http://www.rfc-editor.org/info/rfc3135>.

[RFC6146] Bagnulo, M., Matthews, P., and I. van Beijnum, "Stateful NAT64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers", RFC 6146, DOI 10.17487/RFC6146, April 2011, <http://www.rfc-editor.org/info/rfc6146>.

[RFC6333] Durand, A., Droms, R., Woodyatt, J., and Y. Lee, "Dual-Stack Lite Broadband Deployments Following IPv4 Exhaustion", RFC 6333, DOI 10.17487/RFC6333, August 2011, <http://www.rfc-editor.org/info/rfc6333>.

[RFC6973] Cooper, A., Tschofenig, H., Aboba, B., Peterson, J., Morris, J., Hansen, M., and R. Smith, "Privacy Considerations for Internet Protocols", RFC 6973, DOI 10.17487/RFC6973, July 2013, <http://www.rfc-editor.org/info/rfc6973>.

[RFC7258] Farrell, S. and H. Tschofenig, "Pervasive Monitoring Is an Attack", BCP 188, RFC 7258, DOI 10.17487/RFC7258, May 2014, <http://www.rfc-editor.org/info/rfc7258>.

[RFC7498] Quinn, P., Ed. and T. Nadeau, Ed., "Problem Statement for Service Function Chaining", RFC 7498, DOI 10.17487/RFC7498, April 2015, <http://www.rfc-editor.org/info/rfc7498>.

Author's Address

Mohamed Boucadair (editor)
Orange
Rennes
35000
France

EMail: mohamed.boucadair@orange.com