Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: March 03, 2014

M. Boucadair
C. Jacquenet
France Telecom
R. Parker
Affirmed Networks
D. Lopez
Telefonica I+D
J. Guichard
C. Pignataro
Cisco Systems, Inc.
August 30, 2013

Differentiated Service Function Chaining Framework
draft-boucadair-service-chaining-framework-00

Abstract

IP networks rely more and more on the combination of advanced functions (besides the basic routing and forwarding functions) for the delivery of added value services. This document defines a framework to enforce Service Function Chaining (SFC) with minimum requirements on the underlying network.

The proposed framework allows for Differentiated Forwarding (DiffForward): packets are initially classified and marked at the entry point of an SFC-enabled domain, and are then forwarded on a per SF Map Index basis.

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/.

Copyright Notice

Table of Contents

1. Introduction

1.1. On the Proliferation of Service Functions

IP networks rely more and more on the combination of advanced functions (besides the basic routing and forwarding functions) for the delivery of added value services. Typical examples of such functions include firewall (e.g., [RFC6092]), DPI (Deep Packet Inspection), LI (Lawful Intercept) module, NAT44 [RFC3022], NAT64 [RFC6146], DS-Lite AFTR [RFC6333], NPTv6 [RFC6296], HOST_ID injection, HTTP Header Enrichment function, TCP tweaking and optimization functions, transparent caching, charging function, load-balancer, etc.

Such advanced functions are denoted SF (Service Function) in this document.

The dynamic enforcement of a SF-derived, adequate forwarding policy for packets entering a network that supports such advanced Service Functions has become a key challenge for operators and service providers. SF-inferred differentiated forwarding is ensured by tweaking the set of Service Functions to be invoked. How to bind a flow of packets that share at least one common characteristic to a forwarding plane is policy-based, and subject to the set of SF functions that need to be solicited for the processing of this specific flow.

The overall problem space is described in [I-D.quinn-nsc-problem-statement]. A companion document that lists preliminary requirements is available at [I-D.boucadair-chaining-requirements].

1.2. Scope

This document defines a framework to enforce Service Function Chaining (SFC) with minimum requirements on the underlying network. The proposed solution allows for Differentiated Forwarding (DiffForward): packets are initially classified at the entry point of an SFC-enabled network, and are then forwarded according to the ordered set of SF functions that need to be activated to process these packets in the SFC-enabled domain.

This document does not make any assumption on the deployment context. The proposed framework covers both fixed and mobile networks (e.g., to rationalize the proliferation of advanced features at the Gi Interface [RFC6459]).

Considerations related to the chaining Service Functions that span domains owned by multiple administrative entities is out of scope. Note, a single administrative entity may manage multiple domains.

1.3. Objectives

The main objectives of the proposed framework are listed below:

- Create service-inferred forwarding planes.
- Efficiently master the chained activation of Service functions, regardless of the underlying network topology and routing policies.
- Allow packets to be forwarded to the required Service Functions without changing the underlying network topology or overlay transports necessary for packet delivery to/from Service Functions.
- Allow for differentiated packet forwarding by selecting the set of Service functions to be invoked.
- Allow to easily change the sequentiality of the activation of Service functions to be invoked.
- Allow to easily change the set of Service functions to be invoked.
- Ease management (including withdrawal) of Service functions and minimize any subsequent topology upgrade.
- Automate the overall process of generating and enforcing policies to accommodate a set of network connectivity service objectives.

1.4. Assumptions

The following assumptions are made:

- Not all SFs can be characterized with a standard definition in terms of technical description, detailed specification, configuration, etc.
- There is no global nor standard list of SFs enabled in a given administrative domain. The set of SFs varies as a function of the service to be provided and according to the networking environment.
- There is no global nor standard SF chaining logic. The ordered set of SFs that need to be activated to deliver a given connectivity service is specific to each administrative entity.
- The chaining of SFs and the criteria to invoke some of them are specific to each administrative entity that operates the SF-enabled network (also called administrative domain).
- SF chaining logic and related policies should not be exposed outside a given administrative domain.
- Several SF chaining logics can be simultaneously enforced within an administrative domain to meet various business requirements.
- No assumption is made on how FIBs and RIBs of involved nodes are populated.
- How to bind the traffic to a given SF chaining is policy-based.

1.5. Rationale

Given the assumptions listed in Section 1.4, the rationale of the framework is as follows:

- The framework separates the dynamic provisioning of required SF functions from packet handling operations (e.g., forwarding decisions).
- SFs are handled as black boxes; the technical characterization of each SF is not required.
- No IANA registry is required to store the list of SFs.
- No IANA registry is required to store the SF chaining candidates.
- No specific SF chaining is assumed. The description of SF chains is an information that will be processed by the nodes that participate to the delivery of a network service. The set of listed/chained SF functions is generated by each administrative entity operating the network.
- SF handling is policy-based: SF chains can be updated or deleted, new SFs can be added without any impact on existing SFs, etc. In particular, this design is compliant with the global framework discussed in [I-D.sin-sdnrg-sdn-approach].
- For the sake of efficiency, policy enforcement is automated (but policies can be statically enforced, for example).
- To minimize fragmentation, a minimal set of information needs to be signaled (possibly in data packets).

- Advanced features (e.g., load balancing) are also described and may configured according to policies that can be service-specific. Policy decisions are made by a Policy Decision Point [RFC2753] and the solicited enforcement points are responsible for applying these decisions, whatever the objective to achieve.
- SFs can be embedded in nodes that intervene in the transport service or supported by dedicated nodes (e.g., dedicated servers). The decision to implement one of these two models (or a combination thereof) is deployment-specific and it is orthogonal to the overall procedure.
- Multiple SFC-enabled domains can be deployed within the same administrative domain. Nodes are provisioned with the policy table of the SFC-enabled domain they belong to.
- The overall consistency of the differentiated forwarding policy is ensured by the PDP.
- The PDP can be responsible to enforce other policies than those described in the SFC Policy Tables.

2. Terminology

This document makes use of the following terms:

- DiffForward: refers to the differentiated forwarding procedure as specified in this document.

- SF (Service Function): refers to a function which is enabled in the network operated by an administrative entity. One or many Service Functions can be involved in the delivery of added-value services. A non-exhaustive list of Service Functions include: firewall (e.g., [RFC6092]), DPI (Deep Packet Inspection), LI (Lawful Intercept) module, NAT44 [RFC3022], NAT64 [RFC6146], DS-Lite AFTR [RFC6333], NPTv6 [RFC6296], HOST_ID injection, HTTP Header Enrichment function, TCP optimizer, load-balancer, etc. This document does not make any assumption in the OSI Layer on which the Service Function acts on; the exact definition of each Service Function is deployment-specific.

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

- SF Identifier: is a unique identifier that unambiguously identifies a SF within a SFC-enabled domain. SF Identifiers are assigned, configured and managed by the administrative entity that operates the SFC-enabled domain. SF identifiers can be structured as strings; other formats can be used. SF Identifiers are not required to be globally unique nor be exposed to or used by another SF-enabled domain.

- SF Map: refers to an ordered list of SF identifiers. Each SF Map is identified with a unique identifier called SF Map Index.

- SFC Policy Table: is a table containing a list of SF Maps, SFC classification rules and Locators for all SF Nodes. A SFC Policy Table may contain a default SF Map.

- SF Locator: A SF Node identifier used to reach the said SF node. A locator is typically an IP address or a FQDN.

- Legacy Node (Node for short): refers to any node that is not a SF Node nor a SFC Boundary Node. This node can be located within a SFC-enabled domain or outside a SFC-enabled domain.

3. Functional Elements

The following functional elements are defined in this document:

- SFC Boundary Node (or Boundary Node): denotes a 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.

- SFC Egress Node (or Egress Node): denotes a SFC Boundary Node that handles traffic which leaves the SFC-enabled domain the Egress Node belongs to.

- SFC Ingress Node (or Ingress Node): denotes a SFC Boundary Node that handles traffic which enters the SFC-enabled domain the ingress Node belongs to.

- SF Node: denotes any node within an SFC-enabled domain that embeds one or multiple SFs.

- SFC Classifier (or Classifier): an entity which classifies packets based on the packet header contents according to classification rules defined in a SFC Policy Table. Packets are then marked with the corresponding SF Map Index. SFC Classifier is embedded in a SFC Boundary Node. A SFC Classifier may be identified by a dedicated SF Identifier.

4. SFC Provisioning

4.1. Assign Service Function Identifiers

The administrative entity that operates a SFC-enabled domain maintains a local repository that lists the enabled SFs. This administrative entity assigns a unique SF identifier for each SF type.

SF identifiers can be structured as strings or any other format. The main constraint on the format is that two SFs MUST be assigned with different SF identifiers if they do not provide the exact same function, or do provide the same function but are unable to differentiation packets based on policies provisioned to the SF using an appropriate mechanism. SF identifiers are case-sensitive.

4.2. Service Function Locator

A SF may be embedded in one or several SF Nodes. The SF locator is typically the IP address or the FQDN to reach a given SF Node.

The use of an IP address is RECOMMENDED to avoid any extra complexity related to the support of name resolution capabilities in SF Nodes. Resolution capabilities are supported by the PDP (Policy Decision Point). In the rest of the document, we assume a SF locator is structured as an IP address (IPv4 or IPv6).

A SF Node can be reached by one or more locators, which may therefore be bound to the same SF.

4.3. Service Function Discovery

The local repository that lists the enabled SFs within an SFC-enabled domain may be built as a direct input from the administrative entity, or they may be discovered dynamically through appropriate protocol discovery means.

Whichever method is selected by the administrative entity is a local decision and is therefore outside the scope of this document.

4.4. Building Service Function Maps

Added-value services delivered to the end-user rely on the invocation of several SFs. For each of these services, the administrative entity that operates an SFC-enabled domain builds one or several SF Maps. Each of these maps characterizes the list of SFs to be invoked with their exact invocation order.

Each SF Map is unambiguously identified with a unique identifier called the SF Map Index. The SF Map Index MUST be described as an unsigned integer.

Distinct chains can be applied for inbound and outbound traffic. The directionality of traffic is not included as an attribute of the SF Map, but it may be implicitly described by using two SF Maps installed and maintained in the SFC Policy Table. In such case, incoming packets would be marked with Index_1 for example, while

outgoing packets would be forwarded according to a distinct SF Map identified with Index_2.

An example of SF Map to handle IPv6 traffic destined to an IPv4 remote server is defined as follows:

{15, {IPv6_Firewall, HOST_ID_Inject, NAT64}}.

To handle incoming packets destined to the same IPv6 host, the following SF Map can be defined:

{10, {IPv4_Firewall, NAT64}}.

4.5. Building Service Function Chaining (SFC) Policy Tables

A PDP (Policy Decision Point, [RFC2753]) is the central entity which is responsible for maintaining SFC Policy Tables (Figure 1), and enforcing appropriate policies in SF Nodes and SFC Boundary Nodes (Figure 1). PDP-made decisions can be forwarded to the participating nodes by using a variety of protocols (e.g., NETCONF [RFC6241]).

One or multiple SFC-enabled domains may be under the responsibility of the same PDP. Delimiting the scope of each SFC-enabled domain is under the responsibility of the administrative entity that operates the SF-enabled network.

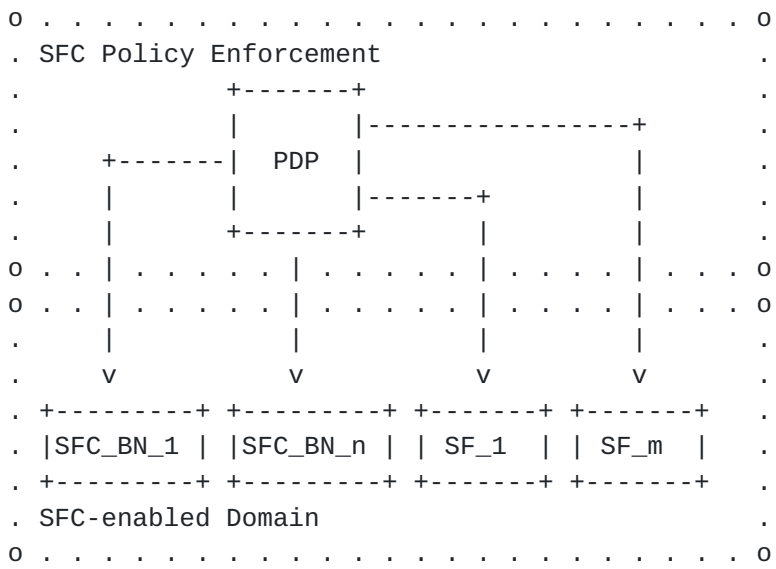


Figure 1: SFC Policy Enforcement Scheme.

The SF Node MUST be provisioned with the following information:

- Local SF Identifier(s): This information is required for an SF to identify itself within an SF Map.
- List of SF Maps: The PDP may configure the full list (default mode) or only as subset of SF Maps in which SF(s) supported by the SF Node is involved (see Section 9.7).
- List of SF Locators: The PDP may configure the full list of locators (default mode) or only the locators of next hop SFs of SF Maps in which SF(s) supported by the local SF node is involved (see Section 9.7).

  [DISCUSSION NOTE: Discuss if we maintain both forms of the SFC Policy table (full and lite) or select only one of them.]

Likewise, the SFC Boundary Node MUST be provisioned with the following information:

- List of SF Maps
- List of SF Locators
- List of SF Map Classification Rules (see Section 5.2).

In addition to the SFC Policy Table, other SF-specific policies can be installed by the PDP (e.g., configure distinct user profiles, activate specific traffic filters, configure traffic conditioners, etc.).

Policies managed by the PDP may be statically instantiated or dynamically triggered by external means (e.g., a AAA server).

In the event of any update (e.g., define a new SF Map, delete an SF Map, add a new SF Locator, update classification policy), the PDP MUST forward the updated policy configuration information in all relevant SF Nodes and SFC Boundary Nodes.

Load-balancing among several SF Nodes supporting the same SF can be driven by the PDP. Indeed, the PDP can generate multiple classification rules and SF Maps to meet load-balancing objectives.

Load balancing may also be achieved locally by an SF Node. If the SF Node, SF Classifier, or SF Boundary Node has a table that provides the SF locator(s) of SF Nodes that provide a particular SF then it is possible to make that local load balancing decision.

The processing of packets by the nodes that belong to a SFC-enabled domain does not necessarily require any interaction with the PDP, depending on the nature of the SF supported by the nodes and the corresponding policies to be enforced. For example, traffic conditioning capabilities [RFC2475] are typical SF functions that may require additional solicitation of the PDP for the SF node to decide what to do with some out-of-profile traffic.

5. Theory Of Operation

The behavior of each node of a SFC-enabled domain is specified in the following sections. We assume that the provisioning operations discussed in Section 4 have been successful (i.e., SF functions have been adequately configured according to the (service-specific) policy to be enforced).

5.1. SFC Boundary Node

SFC Boundary Nodes act both as a SFC Ingress Node and as a SFC Egress Node for the respective directions of the traffic.

Traffic enters a SFC-enabled domain at a SFC Ingress Node (Section 5.3) and exits the domain at a SFC Egress Node (Section 5.4).

5.2. SFC Classifier

The SFC Classifier classifies packets based on (some of) the contents of the packet header. Concretely, it classifies packets based on the possible combination of one or more header fields, such as source address, destination address, DS field, protocol ID, source port and destination port numbers, and any other information.

Each SF Map Classification Rule MUST be bound to one single SF Map (i.e., the classification rule must include only one SF Map Index).

5.3. SFC Ingress Node

When a packet is received through an interface of the SFC Ingress Node that connects to the outside of the SFC domain, the Ingress Node MUST:

- Inspect the received packet and check whether any existing SF Map Index is included in the packet.

    * The SFC Ingress Node SHOULD be configurable with a parameter to indicate whether received SF Map Index is to be preserved or striped. The default behavior is to strip any received SF Map Index.
    * Unless explicitly configured to trust SF Map index, The SFC Ingress Node MUST strip any existing SF Map Index if the packet is received from an SFC-enabled domain that has not explicitly been designated as "trusted".
- Check whether the received packet matches an existing classification rule (see Section 5.2).
- If no rule matches, forward the packet to the next hop according to legacy forwarding behavior (e.g., based upon the IP address conveyed in the DA field of the header).
- If a rule matches, proceed with the following operations:

    * Retrieve the locator of the first SF as indicated in the SF Map entry the rule matches.
    * Check whether the corresponding SF node is an immediate neighbor.

        + If so, update the packet with the SF Map Index of SF Map entry it matches and then forward the packet to the corresponding SF Node.
        + If not, (1) encapsulate the original packet into a new one that will be forwarded to the corresponding SF node, (2) update the encapsulated packet with the SF Map Index of SF Map entry it matches, and (3) forward the packet to the next hop to reach the first SF node.

As a result of this process, the packet will be sent to an SF Node or an Intermediate Node.

5.4. SFC Egress Node

When a packet is received through an interface that connects the SFC Egress Node to its SFC domain, the Egress Node MUST:

- Strip any existing SF Map Index.
- Forward the packet according to legacy forwarding policies.

5.5. SF Node

This section assumes the default behavior is each SF Node does not embed a Classifier as discussed in Section 9.4.

When a packet is received by a SF Node, the SF Node MUST:

- Check whether the packet conveys a SF Map Index.

- o If no SF Map Index is included, forward the packet according to legacy forwarding policies.
- o If the packet conveys a SF Map Index,
  - * Retrieve the corresponding SF Map from the SFC Policy Table. If no entry is found in the table, forward the packet according to legacy forwarding policies.

    [DISCUSSION NOTE: Another design choice is to drop the packet and send a notification to the PDP. The justification for avoiding to drop the packet is that an SF can be part of the forwarding path of an SFC to which it does not belong to.]
  - * If an entry is found in the SFC Policy Table, check whether the local SF Identifier is present in the SF Map:
    - + If not, forward the packet according to legacy forwarding policies.

      [DISCUSSION NOTE: One would argue the packet should be dropped. The justification for avoiding to drop the packet is that an SF can be part of the forwarding path of an SFC to which it does not belong to + the SF node is provisioned with the full SFC Policy Table.]
    - + If so, the packet is decapsulated (if needed) and then presented as an input to the local SF. In case several SFs are co-located in the same node, the packet is processed by all SFs indicated in the SF Map. Once the packet is successfully handled by local SF(s), the packet is forwarded to the next SF Node in the list or to an intermediate node (if the local SFC Node is the last element in the SF Map). If the local SF node is not the last one in the SF Map, it retrieves the next SF Node from the list, retrieve its locator for the SFC Policy Table, and forwards the packet to the next hop. If the local SF Node is the last element in the SF Map, it forwards the packet to the next hop according to legacy forwarding policies.

5.6. Intermediate Nodes

An Intermediate Node is any node that does not support any Service Function and which is located within a SFC-enabled domain.

No modification is required to intermediate nodes to handle incoming packets. In particular, routing and forwarding are achieved using legacy procedures.

6. Fragmentation Considerations

If adding the Service Chaining Header would result in a fragmented packet, the classifier should include a Service Chaining Header in each fragment.

Other fragmentation considerations will be added in a future version of the document.

7. Differentiated Services

When encapsulating an IP packet, the Ingress Node and each SF Node SHOULD use its Diffserv Codepoint (DSCP, [RFC2474]) to derive the DSCP (or MPLS Traffic-Class Field) of the encapsulated packet.

Generic considerations related to Differentiated Services and tunnels are further detailed in [RFC2983].

8. Design Considerations

This section discusses two main protocol issues to be handled in order to deploy DiffForward.

A detailed design analysis is documented in [I-D.boucadair-chaining-design-analysis].

8.1. Transmit A SFC Map Index In A Packet

8.1.1. SFC Map Index

A SF Map Index is an integer that points to a SF Map.

In order to avoid all nodes of a SFC-enabled domain to be SF-aware, this specification recommends to undertake classifiers at boundary nodes while intermediate nodes forward the packets according to the SF Map Index conveyed in the packet (SF Node) or according to typical forwarding policies (any SF-unaware node).

An 8-bit field would be sufficient to accommodate deployment contexts that assume a reasonable set of SF Maps. A 16-bit (or 32-bit) field would be more flexible (e.g., to accommodate the requirement discussed in Section 9.3).

8.1.2. Why Not Loose Or Strict Source Routing (SSR)?

Instead of injecting a Map Index, an alternate solution would be to use the SSRR/LSRR IPv4 option or any similar solution to indicate a loose or strict explicit route. This alternative was not considered because of the likely dramatic impact on the processing and potential fragmentation issues that may jeopardize the overall performance of the DiffForward operation.

Injecting an 8-bit or a 16-bit field would minimize fragmentation issues.

8.1.3. Where To Store SFC Map Indexes In A Packet?

SF Map Indexes can be conveyed in various locations of a packet:

- At L2 level
- Define a new IP option or a new IPv6 extension header
- Use IPv6 Flow Label
- Re-use an existing field (e.g., DS field)
- TCP option
- GRE Key
- Define a new shim
- Etc.

8.2. Steer Paths To Cross Specific SF Nodes

A SFC Ingress Node or a SF Node MUST be able to forward a packet that matches an existing SF Map to the relevant next hop SF Node. The locator of the next SF is retrieved from the SFC Policy Table. In case the next SF Node in the list is not an immediate neighbor, a solution to force the packet to cross that SF Node MUST be supported. This document suggests the use of the IP-in-IP encapsulation scheme. Other tunneling solutions can be considered in the future.

9. Deployment Considerations

9.1. Generic Requirements

The following deployment considerations should be taken into account:

- Avoid inducing severe path stretch compared to the path followed if no SF is involved.
- Minimize path computation delays: due to the enforcement of classification rules in all participating nodes, misconception of Service function chaining, inappropriate choice of nodes elected to embed Service functions, etc., must be avoided.

- o Avoid SF invocation loops: the design of SF chainings should minimize as much as possible SF invocation loops.

9.2. Deployment Models

Below are listed some deployment model examples:

1. A full marking mechanism: Ingress nodes perform the classification and marking functions. Then, involved SF Nodes process received packets according to their marking.

2. SF node mechanism, in which every SF Node embeds also a classifier, and the ingress node only decides the first node to forward to. Packets are forwarded at each node according to local policies. No marking is required when all SFs are co-located with a classifier. This model suffers from some limitations (see Section 9.4).

3. A router-based mechanism: All SF Nodes forward packets once processed to their default router. This default routes is responsible for deciding how the packet should be progressed at each step in the chain. One or multiple routers can be involved in the same Service Function Chain.

4. A combination thereof.

9.3. On Service Function Profiles (a.k.a., Contexts)

Service Functions may often enforce multiple differentiated policy sets. These policy sets may be coarsely-grained or fine-grained. An example of coarsely-grained policy sets would be an entity that performs HTTP content filtering where one policy set may be appropriate for child users whereas another is appropriate for adult users. An example of finely-grained policy sets would be PCEF (3GPP Policy Control Enforcement Function) that has a large number of differentiated QoS and charging profiles that are mapped on a per-subscriber basis.

The Service Function Chaining mechanism directly support coarsely-grained differentiated policy sets and indirectly support finely-grained differentiated policy sets.

From a Service Function Chaining perspective, each coarsely-grained policy set for a Service Function will be considered as a distinct logical instance of that Service Function. Consider the HTTP content filtering example where one physical or virtual entity provides both child and adult content filtering. The single entity is represented as two distinct logical Service Functions, each with their own

Service Function Identifier from a chaining perspective. The two (logical) Service Functions may share the same IP address or may have distinct IP addresses.

Finely-grained policy sets, on the other hand, would unacceptably explode the number of distinct Service Chains that were required with an administrative domain. For this reason, Service Functions that utilize finely-grained policy sets are represented as a single Service Function that has its own internal classification mechanism in order to determine which of its differentiated policy sets to apply. Doing so avoids from increasing the size of the SFC Policy Table.

The threshold, in terms of number of policies, between choosing the coarsely-grained policy or finely-grained policy technique is left to the administrative entity managing a given domain.

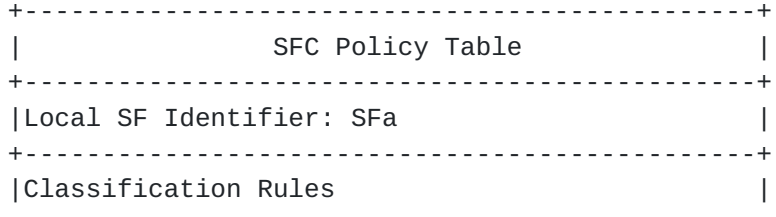
> [DISCUSSION NOTE: This section will be updated to reflect the conclusions of the discussions from the design analysis draft.]

9.4. SF Node is also a Classifier

If SF Nodes are also configured to behave as Classifiers, the SF Map Index is not required to be explicitly signalled in each packet. Concretely, the SFC Policy Table maintained by the SF Node includes classification rules. These classification rules are enforced to determine whether the local SF must be involved. If an incoming packet matches at least one classification rule pointing to an SF Map in which the SF Identifier is listed, the SF Node retrieves the next hop SF from the SF Map indicated in the classification rule.

The packet is then handled by the local SF, and the SF Node subsequently forwards the packet to the next hop SF. If not, the packet is forwarded to the next hop according to a typical IP forwarding policy.

Let us consider the example shown in Figure 2. The local SF Node embeds SFa. Once classification rules and the SF Maps are checked, the SF Node concludes SFa must be invoked only when a packet matches Rules 1 and 3. If a packet matches Rule 1, the next SF is SFC. If a packet matches Rule 3, the next SF is SFh.



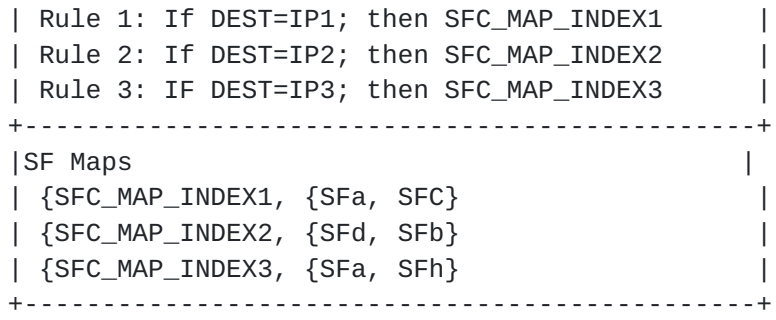


Figure 2: SFC Policy Table Example.

9.5. Direct Adjacency

SF Nodes may be enabled in a SFC-enabled domain so that each of them has a direct adjacency with other SF Nodes. In such configuration, no encapsulation scheme is required to exchange traffic between these nodes.

9.6. Service Function Loops

SF Nodes use the SFC Policy Table to detect whether the local SF was already applied to the received packet (i.e., detect SF Loop). The SF Node MUST invoke the local SF only if the packet is received from a SFC Boundary Node or a SF Node having an identifier listed before the local SF in the SF Map matched by the packet. SF Loop detection SHOULD be a configurable feature.

Figure 3 shows an example of a SFC Policy Table of a SF Node embedding SFa. Assume a packet received from Locb that matches Rule 2. SFa must not be invoked because SFb is listed after SFa (see the SF Map list). That packet will be forwarded without invoking SFa.

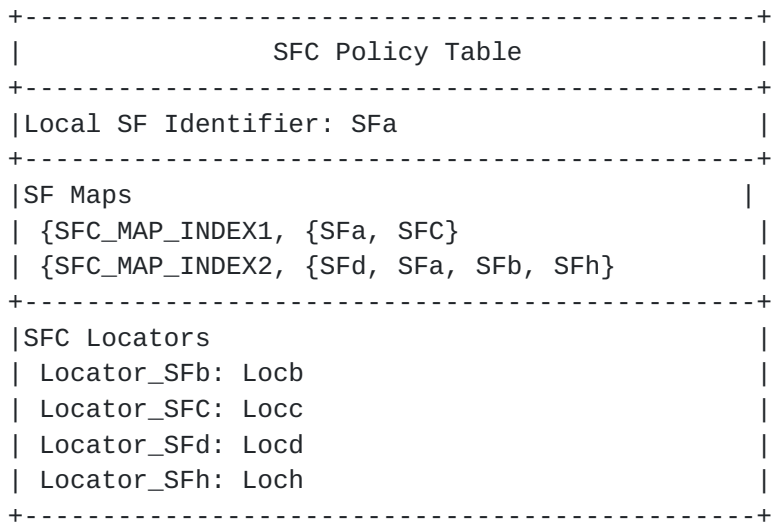


Figure 3: Dealing With SF Loops.

9.7. Lightweight SFC Policy Table

If SF loop detection is not activated in an SFC-enabled domain, the PDP may provision SF nodes with a "lightweight" SFC Policy Table. A lightweight SFC Policy Table is a subset of the full SFC Policy Table that includes:

- Only the SF Maps in which the local SF is involved.
- Only the next hop SF instead of the full SF chain.

An example of a lightweight SFC Policy Table is shown in Figure 4.

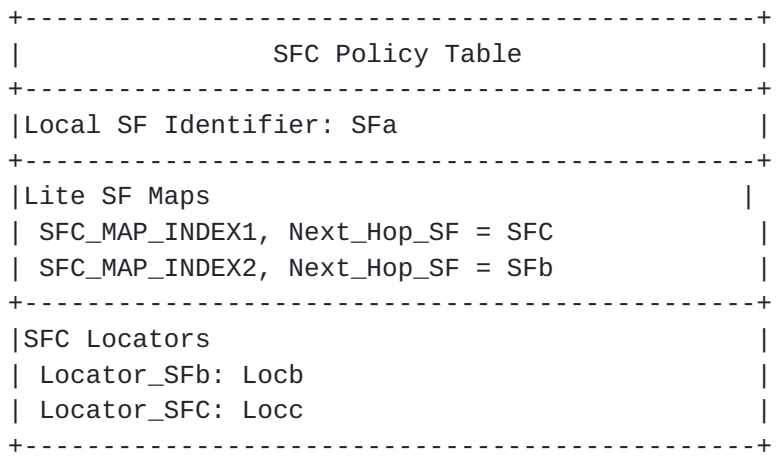


Figure 4: Lightweight SFC Policy Table.

9.8. Liveness Detection Of SFs By The PDP

The ability of the PDP to check the liveness of each SF invoked in a service chain has several advantages, including:

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

In order to determine the liveness of any particular SF Node, standard protocols such as ICMP or BFD (both single-hop [RFC5881] and multi-hop [RFC5883]) may be utilized between the PDP and the SF Nodes.

Because an SF Node can be responsive from a reachability standpoint (e.g., IP level) while the function its 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 (e.g., [RFC6849], [I-D.tsou-softwire-bfd-ds-lite]).

For more sophisticated load-balancing support, protocols that allow for both liveness determination and the transfer of application-specific data, such as SNMP and NETCONF may be utilized between the PDP and the SF Nodes.

10. IANA Considerations

Required IANA actions will be discussed in future versions of the document.

11. Security Considerations

Means to protect SFC Boundary Nodes and SF Nodes against various forms of DDoS attacks MUST be supported. For example, mutual PDP and SF node authentication should be supported. Means to protect SF nodes against malformed, poorly configured (deliberately or not) SFC Policy Tables should be supported.

SFC Boundary Nodes MUST strip any existing SF Map Index when handling an incoming packet. A list of authorized SF Map Indexes are configured in the SFC elements.

NETCONF-related security considerations are discussed in [RFC6146].

Means to prevent SF loops should be supported.

Nodes involved in the same SFC-enabled domain MUST be provisioned with the same SFC Policy Table. Possible table inconsistencies may result in forwarding errors.

12. Contributors

The following individuals contributed to the document:

Parviz Yegani
Juniper Networks
1194 N. Mathilda Ave.
Sunnyvale, CA 94089
USA
Email: pyegani@juniper.net

Paul Quinn

Cisco Systems, Inc.
USA
Email: paulq@cisco.com

13. Acknowledgments

Many thanks to D. Abgrall, D. Minodier, Y. Le Goff, and D. Cheng for their review and comments.

14. References

14.1. Normative References

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

[RFC6241] Enns, R., Bjorklund, M., Schoenwaelder, J., and A. Bierman, "Network Configuration Protocol (NETCONF)", RFC 6241, June 2011.

14.2. Informative References

[I-D.boucadair-chaining-requirements]
Boucadair, M., Jacquenet, C., Jiang, Y., Hongyu, L., and R. Parker, "Requirements for Service Function Chaining", draft-boucadair-chaining-requirements-01 (work in progress), August 2013.

[I-D.quinn-nsc-problem-statement]
Quinn, P., Guichard, J., Surendra, S., Agarwal, P., Manur, R., Chauhan, A., Leymann, N., Boucadair, M., Jacquenet, C., Smith, M., Yadav, N., Nadeau, T., Gray, K., McConnell, B., and K. Kevin, "Network Service Chaining Problem Statement", draft-quinn-nsc-problem-statement-03 (work in progress), August 2013.

[I-D.sin-sdnrg-sdn-approach]
Boucadair, M. and C. Jacquenet, "Software-Defined Networking: A Service Provider's Perspective", draft-sin-sdnrg-sdn-approach-03 (work in progress), June 2013.

[I-D.tsou-softwire-bfd-ds-lite]
Tsou, T., Li, B., Zhou, C., Schoenwaelder, J., Penno, R., and M. Boucadair, "DS-Lite Failure Detection and Failover", draft-tsou-softwire-bfd-ds-lite-05 (work in progress), June 2013.

[RFC2474] Nichols, K., Blake, S., Baker, F., and D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers", RFC 2474, December 1998.

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

[RFC2753] Yavatkar, R., Pendarakis, D., and R. Guerin, "A Framework for Policy-based Admission Control", RFC 2753, January 2000.

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

[RFC3022] Srisuresh, P. and K. Egevang, "Traditional IP Network Address Translator (Traditional NAT)", RFC 3022, January 2001.

[RFC5881] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD) for IPv4 and IPv6 (Single Hop)", RFC 5881, June 2010.

[RFC5883] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD) for Multihop Paths", RFC 5883, June 2010.

[RFC6092] Woodyatt, J., "Recommended Simple Security Capabilities in Customer Premises Equipment (CPE) for Providing Residential IPv6 Internet Service", RFC 6092, January 2011.

[RFC6146] Bagnulo, M., Matthews, P., and I. van Beijnum, "Stateful NAT64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers", RFC 6146, April 2011.

[RFC6296] Wasserman, M. and F. Baker, "IPv6-to-IPv6 Network Prefix Translation", RFC 6296, June 2011.

[RFC6333] Durand, A., Droms, R., Woodyatt, J., and Y. Lee, "Dual-Stack Lite Broadband Deployments Following IPv4 Exhaustion", RFC 6333, August 2011.

[RFC6459] Korhonen, J., Soininen, J., Patil, B., Savolainen, T., Bajko, G., and K. Iisakkila, "IPv6 in 3rd Generation Partnership Project (3GPP) Evolved Packet System (EPS)", RFC 6459, January 2012.

[RFC6849] Kaplan, H., Hedayat, K., Venna, N., Jones, P., and N. Stratton, "An Extension to the Session Description Protocol (SDP) and Real-time Transport Protocol (RTP) for Media Loopback", RFC 6849, February 2013.

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

Ron Parker
Affirmed Networks
Acton, MA
USA

EMail: Ron_Parker@affirmednetworks.com

Diego R. Lopez
Telefonica I+D
Don Ramon de la Cruz, 82
Madrid 28006
Spain

Phone: +34 913 129 041
EMail: diego@tid.es

Jim Guichard
Cisco Systems, Inc.
USA

EMail: jguichar@cisco.com

Carlos Pignataro
Cisco Systems, Inc.
USA

EMail: cpignata@cisco.com