Internet Working Group Y. Jiang
H. Li
Internet Draft Huawei

Intended status: Informational

Expires: December 2013 June 27, 2013

An Architecture of Service Chaining
draft-jiang-service-chaining-arch-00.txt

Status of this Memo

Copyright Notice

Abstract

As network virtualization is opening the gate to much more innovative services for service providers, service chaining provides a flexible way of service provisioning and facilitates their deployment.

This document provides a general abstract architecture for service chaining. It is a flexible and scalable architecture which can fulfill requirements of service chaining. Some solutions based on this architecture are also discussed with their advantages and disadvantages. This architecture can be used as a guideline and also a criterion for the design of service chaining.

Table of Contents

1. Conventions used in this document

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

2. Terminology

Service Flow: packets/frames with specific service characteristics (e.g., packets matching a specific tuple of fields in Ethernet, IP, TCP, HTTP headers and etc.) or determined by some service policies (such as access port and etc.)

Service Classifier (SCLA), an entity which can classify incoming packets/frames into different service flows based on their service characteristics or some policies.

Service Processing Entity (SPE): a logical entity which can provide one or more service processing functions for packets/frames such as firewall, DPI (Deep Packet Inspection), LI (Lawful Intercept) and etc. Usually these processing functions are computation intensive. This entity may also provide packet/frame encapsulation/decapsulation capability.

Service Chain: one or more service processing functions in a specific order which are chained to provide a composite service, and packets/frames from one or more service flow should follow.

Service Chaining Domain: a domain where packet is forwarded using service chaining mechanism.

Service Forwarding Entity (SFE): a logical entity which forwards packets/frames to one or more SPEs in a same service chain. Optionally, it provides mapping, insertion and removal of header(s) in packets/frames. Note service forwarding path may not be the shortest path to its destination.

Service Chaining: a mechanism of building service chains and forwarding packets/frames of service flows through them.

3. Introduction

With the maturity of hardware/software in network virtualization, it is possible for service providers to provide more innovative services.

Service chaining provides a flexible way to construct services, e.g., it is easy to insert/remove, and upgrade service processing functions for a service in this framework. It is thus possible to define very complex services over heterogeneous networks in a consistent way with the help of service chaining.

Several drafts have already been proposed for service chaining with different approaches ([NLFC], [NSH] and [L3VPNSC]) and more solutions are expected to emerge in the near future.

This document provides architecture and abstraction of the critical components for service chaining, and it is hoped to shed some lights on the design of service chaining. Possible solutions of service chaining are also outlined and compared in this document so that they can be gauged under the same criterion.

4. Service Chaining Architecture

In general, a service chaining architecture can be abstracted to consist of following components:

- At least one Service Classifier (SCLA), which classifies incoming packets/frames into different service flows based on their service characteristics or some policies;

- One or more Service Forwarding Entities (SFEs), each of which is attached with one or more Service Processing Entities (SPEs).

Service chaining architecture with a single SFE is demonstrated in Figure 1. Upon their entry into a service chaining domain, packets/frames are classified by SCLA into different service flows. Then packets/frames from a specific service flow are forwarded by SFE into one or more SPEs in a service chain in correct order. Upon completion of all service processing, they exit the service chaining domain. It should be noted that Service Classifier, Service Forwarding Entity, and even Service Processing Entity can be implemented in a single network element.

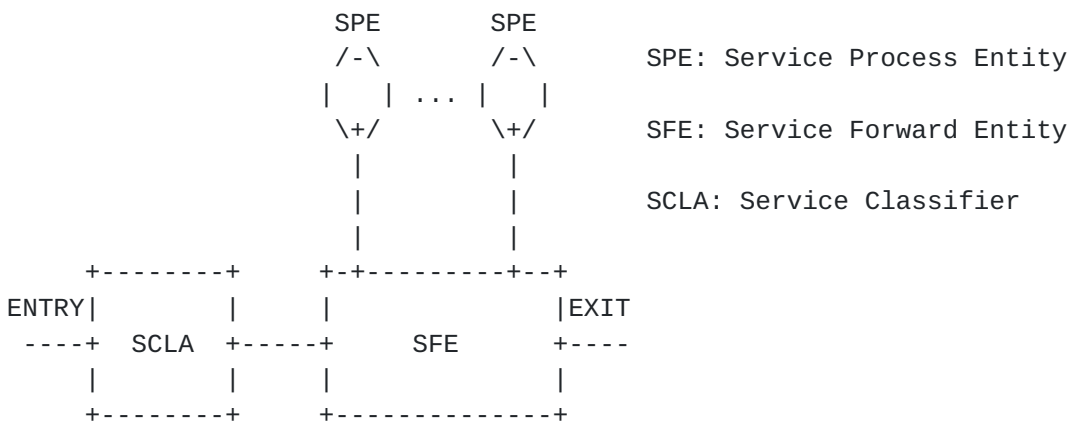

Figure 1 Service Chain with a single SFE

Service chaining architecture with multiple SFEs is further demonstrated in Figure 2. In addition to the previous discussion, multiple SFEs are interconnected by a physical/logical link or a

network path which may be an Ethernet or IP/MPLS underlay network. Ethernet, IP/MPLS or other tunneling technologies such as those being in progress in NVO3 workgroup can be used for this underlay network.

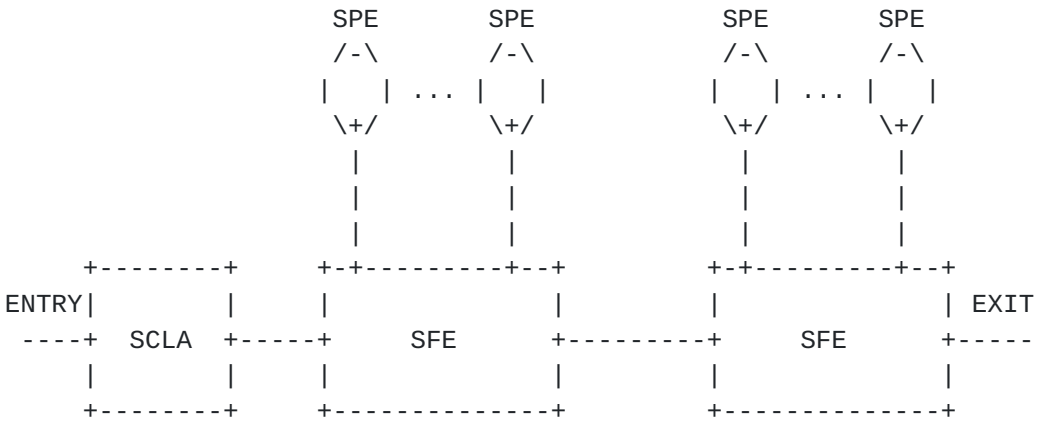

Figure 2 Service Chain with Multiple SFEs

5. Service Chaining Topology

SCLA, SPE and SFE can be implemented as a standalone network entity, or in a network entity combined with other functions (such as one SFE plus one or more SPEs, one SCLA plus one SFE, or one SCLA plus one SFE and one or more SPEs). Furthermore, they may be implemented as software modules running in a DC, in clouds or as standalone physical equipments.

Following service chaining topologies can be supported by this architecture:

-Daisy chain

This is a type of service chain in the shape of a daisy, as demonstrated in Figure 1, where multiple SPEs are attached to a single SFE, and SPEs cannot send packets/frames directly to each other but via one shared SFE.

-Lily chain

This is a type of service chain in the shape of a lily. It can be regarded as a simplified form of Figure 2, where only a single SPE is attached to each SFE, and the SFEs are connected from one to another in sequence.

-Hybrid chain

Part of a hybrid chain may be Daisy chain and other parts may be Lily chain, thus it can be readily combined by the previous two.

6. Service Chaining Construction

Attachment of SPEs onto an SFE could be pre-configured on the SFE; or updated by some auto discovery and registering procedure when an SPE first attaches to it.

6.1. Service chaining Controller

A service chaining controller which centrally manages service chains (e.g., set up, remove, and monitor service chains) can be implemented together with an SDN controller or a network orchestrator. OSS may also be used as a platform to provide this kind of control function.

7. Availability and Scalability of Service Chaining

Two options for load balancing are possible:

-Load balancing on flows

It is very convenient to provide load balancing and make it scalable in this architecture: construct multiple service chains which provide the same set of service processing functions; the SCLA classifies a service flow into sub-flows, and each sub-flow is directed into a different service chain.

-Load balancing on SPEs/paths

It is also possible to provide load balancing in finer granularity. Such as, provide multiple SPEs with the same service processing function if a compute bottleneck is expected or found for this service processing function; provide multiple paths between a pair of SFEs if shortage of bandwidth is expected or found for the single path between them. The SFE should be able to load balance a service flow over these SPEs or paths with some pre-determined algorithms.

High availability of a service chain will be discussed in a next version.

8. Service Chaining Solution Considerations

A solution must be able to classify packets/frames into different service flows. This is done by the Service Classifier (a packet/frame is classified with a tuple of fields in one or more headers or local policy). After the classification, a Service Identification (SID) may be applied to the packet/frame so that no further classification is needed in subsequent SFEs. SID can be mapped to traditional header fields such as a VLAN or an MPLS label (hence VLAN mapping or MPLS label swap is needed), or carried in a new service header (hence an extra header is inserted).

A solution must be able to forward packets/frames across all service processing entities in a service chain in a correct sequence. This is done by the service forwarding entity with the help of SID.

A solution should support the exchange of information between service processing entity and the service forwarding entity. For example, the processing result of a packet in a service processing entity can be encoded in a service header in the packets. But for long term service states (which can influence the processing of multiple packets/frames), they can be locally stored and/or signaled to service forwarding entity or other service processing entities by some specific control channels (e.g., OpenFlow protocol).

Possible solutions for service chaining are outlined in this section. Note this may not be a complete picture, and it is expected that more solutions will emerge in the future.

8.1. Ethernet compatible solution

Ethernet technology (IEEE 802.1Q, 802.1ad, and etc.) or DC technology can be used to support service chaining. SCLA maps service ID of a service flow to a VLAN (e.g., C-VLAN and S-VLAN) ID or a VXLAN ID in a frame. SFE then forwards the frame with the VLAN ID to appropriate SPEs for service processing (may need to change VLAN ID for each SPE).

The issue with this solution: its service chaining layer is entangled with its transport layer, thus planning and provisioning of service chains are complex, and it may not be a scalable solution to be used in a large network since VLAN ID or VXLAN ID space is shared by both service ID and tunnel ID.

8.2. IP/MPLS compatible solution

MPLS can be used to support service chaining - only an MPLS label is needed to represent SID of a service flow. Service ID of a service flow is mapped to an MPLS label in the packets by the SCLA. SFE then forwards frames with the label to appropriate SPEs for service processing (may need to swap the label for each SPE).

[L3VPNSC] demonstrates how service topology specific Route Targets can be introduced into BGP/MPLS VPN to support automatic signaling of service chaining.

Similarly, L2VPN may be crafted to support service chaining.

8.3. Solution of Network-Located Function Chaining (NLFC)

[NLFC] proposes a solution of using NLFC Map (an ordered list of NLF identifiers) for service chaining, where a packet carries a map index of its service chain, then NLF nodes forward the packet by looking up the map index in locally stored NLFC Policy Table for its next NLF node.

8.4. Solution with a standalone service header

A standalone service header can provide an independent layer of protocol data unit, so that service specific information such as SID can be carried over all kinds of underlay networks with no need of mapping.

[NSH] proposes a new Network Service Header for service chaining, which consists of a Base Header and a Context Header.

9. Security Considerations

This document proposes architecture for the service chaining, outlines several types of its topology, and discusses possible approaches in this domain, thus no security issue is raised at present. It was expected that further solutions for these requirements will deal with security considerations specifically.

10. IANA Considerations

No IANA action is needed for this document.

11. References

11.1. Normative References

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

11.2. Informative References

[NLFC] Boucadair, M., and Jacquenet, C., "Differentiated Network-Located Function Chaining Framework", draft-boucadair-network-function-chaining-00, June 2013.

[NSH] Quinn, P., and Fernando, R., and et al, "Network Service Header", draft-quinn-nsh-00, June 2013.

[L3VPNSC] Fernando, R., Rao, D., and et al, "Virtual Topologies for Service Chaining in BGP IP VPNs", draft-rfernando-l3vpn-service-chaining-01, June 2013.

12. Acknowledgments

TBD.

Authors' Addresses

Yuanlong Jiang
Huawei Technologies Co., Ltd.
Bantian, Longgang district
Shenzhen 518129, China
Email: jiangyuanlong@huawei.com

Hongyu Li
Huawei Technologies Co., Ltd.
Bantian, Longgang district
Shenzhen 518129, China
Email: hongyu.lihongyu@huawei.com