

Network Working Group
Internet-Draft
Intended status: Informational
Expires: February 27, 2014

P. Quinn
J. Guichard
S. Kumar
Cisco Systems, Inc.
P. Agarwal
R. Manur
Broadcom
A. Chauhan
Citrix
N. Leymann
Deutsche Telekom
M. Boucadair
C. Jacquenet
France Telecom
M. Smith
N. Yadav
Insieme Networks
T. Nadeau
K. Gray
Juniper Networks
B. McConnell
Rackspace
K. Glavin
Riverbed
August 26, 2013

Network Service Chaining Problem Statement
[draft-quinn-nsc-problem-statement-03.txt](#)

Abstract

This document provides an overview of the issues associated with the deployment of services functions (such as firewalls, load balancers) in large-scale environments. The term service function chaining is used to describe the deployment of such service functions, and the ability of a network operator to specify an ordered list of service functions that should be applied to a deterministic set of traffic flows. Such service function chains require integration of service policy alongside the deployment of applications, while allowing for the optimal utilization of network resources.

Status of this Memo

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

Internet-Drafts are working documents of the Internet Engineering

Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on February 27, 2014.

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

- [1. Introduction](#) [4](#)
- [1.1. Definition of Terms](#) [4](#)
- [2. Problem Areas](#) [6](#)
- [3. Service Function Chaining](#) [9](#)
- [4. Service Function Chaining Use Cases](#) [11](#)
- [4.1. Enterprise Data Center Service Chaining](#) [11](#)
- [4.2. Mobility Service Chaining](#) [11](#)
- [5. Related IETF Work](#) [12](#)
- [6. Summary](#) [13](#)
- [7. Security Considerations](#) [14](#)
- [8. Acknowledgments](#) [15](#)
- [9. References](#) [16](#)
- [9.1. Normative References](#) [16](#)
- [9.2. Informative References](#) [16](#)
- [Authors' Addresses](#) [17](#)

1. Introduction

Services that are composed of service functions require more flexible service function deployment models than those typically available in networks today. Such services may utilize traditional network service functions (for example firewalls and server load balancers), as well as higher layer applications and features. Services may be delivered within a specific context so that isolated user groups attached to a common network may be formed. Such user groups may require unique capabilities with the ability to tailor service characteristics on a per-tenant/per-subscriber/per-VPN basis that must not affect other user groups

Current service function deployment models are relatively static in that they are bound to fixed network topologies and resources. At present, these deployments are not easily manipulated (i.e.: moved, created or destroyed) even when virtualized elements are deployed. This poses a problem in highly elastic service environments that require relatively rapid creation, destruction or movement of real or virtual service functions or network elements. Additionally, the transition to virtual platforms requires an agile service insertion model that supports elastic and very granular service delivery, and post-facto modification; supports the movement of service functions and application workloads in the existing network, all the while retaining the network and service policies and the ability to easily bind service policy to granular information such as per-subscriber state.

This document outlines the problems encountered with existing service deployment models for service function chaining (often referred to simply as service chaining; in this document the terms will be used interchangeably), as well as the problems of service chain creation/deletion, policy integration with service chains, and policy enforcement within the network infrastructure.

1.1. Definition of Terms

Classification: Locally instantiated policy and customer/network/service profile matching of traffic flows for identification of appropriate outbound forwarding actions.

Network Overlay: Logical network built on top of existing network (the underlay). Packets are encapsulated or tunneled to create the overlay network topology.

Service Chain: A service chain defines the required functions and associated order (service-function1 --> service-function 2) that must be applied to packets and/or frames. A service chain does not specify the network location or specific instance of service functions (e.g. firewall1 vs. firewall2).

Service Function: A network or application based packet treatment, application, compute or storage resource, used singularly or in concert with other service functions within a service chain to enable a service offered by a network operator.

A non-exhaustive list of Service Functions includes: firewalls, WAN and application acceleration, Deep Packet Inspection (DPI), server load balancers, NAT44 [[RFC3022](#)], NAT64 [[RFC6146](#)], HOST_ID injection, HTTP Header Enrichment functions, TCP optimizer, etc.

The generic term "L4-L7 services" is often used to describe many service functions.

Service Node: Physical or virtual element providing one or more service functions.

Network Service: An externally visible service offered by a network operator; a service may consist of a single service function or a composite built from several service functions executed in one or more pre-determined sequences and delivered by one or more service nodes.

2. Problem Areas

The following points describe aspects of existing service deployment that are problematic, and are being addressed by the network service chaining effort.

1. Topological Dependencies: Network service deployments are often coupled to the physical network topology creating constraints on service delivery and potentially inhibiting the network operator from optimally utilizing service resources. This limits scale, capacity, and redundancy across network resources.

These topologies serve only to "insert" the service function (i.e. ensure that traffic traverse a service function); they are not required from a native packet delivery perspective. For example, firewalls often require an "in" and "out" layer-2 segment and adding a new firewall requires changing the topology (i.e. adding new L2 segments).

As more service functions are required - often with strict ordering - topology changes are needed before and after each service function resulting in complex network changes and device configuration. In such topologies, all traffic, whether a service function needs to be applied or not, often passes through the same strict order.

A common example is web servers using a server load balancer as the default gateway. When the web service responds to non-load balanced traffic (e.g. administrative or backup operations) all traffic from the server must traverse the load balancer forcing network administrators to create complex routing schemes or create additional interfaces to provide an alternate topology.

2. Configuration complexity: A direct consequence of topological dependencies is the complexity of the entire configuration, specifically in deploying service chains. Simple actions such as changing the order of the service functions in a service chain require changes to the topology. Changes to the topology are avoided by the network operator once installed, configured and deployed in production environments fearing misconfiguration and downtime. All of this leads to very static service delivery models. Furthermore, the speed at which these topological changes can be made is not rapid or dynamic enough as it often requires manual intervention, or use of slow provisioning systems.

The service itself can contribute to complexity: it may require an intricate combination of very different capabilities,

regardless of the underlying topology. QoS-based, resilient VPN service offerings are a typical example of such complexity.

3. **Constrained High Availability:** An effect of topological dependency is constrained service function high availability. Worse, when modified, inadvertent non-high availability can result.

Since traffic reaches service functions based on network topology, alternate, or redundant service functions must be placed in the same topology as the primary service.

4. **Consistent Ordering of Service Functions:** Service functions are typically independent; service function₁ (SF1)...service function_n (SF_n) are unrelated and there is no notion at the service layer that SF1 occurs before SF2. However, to an administrator many service functions have a strict ordering that must be in place, yet the administrator has no consistent way to impose and verify the ordering of the functions that used to deliver a given service.
5. **Service Chain Construction:** Service chains today are most typically built through manual configuration processes. These are slow and error prone. With the advent of newer service deployment models the control / management planes will provide not only connectivity state, but will also be increasingly utilized for the formation of services. Such a control / management plane could be centrally controlled and managed, or be distributed between a subset of end-systems.
6. **Application of Service Policy:** Service functions rely on topology information such as VLANs or packet (re) classification to determine service policy selection, i.e. the service function specific action taken. Topology information is increasingly less viable due to scaling, tenancy and complexity reasons. The topological information is often stale, providing the operator with inaccurate placement that can result in suboptimal resource utilization. Per-service function packet classification is inefficient and prone to errors, duplicating functionality across service functions. Furthermore packet classification is often too coarse lacking the ability to determine class of traffic with enough detail.
7. **Transport Dependence:** Service functions can and will be deployed in networks with a range of transports, including under and overlays. The coupling of service functions to topology requires service functions to support many transports or for a transport gateway function to be present.

8. **Elastic Service Delivery:** Given the current state of the art for adding/removing service functions largely centers around VLANs and routing changes, rapid changes to the service layer can be hard to realize due to the risk and complexity of such changes.
9. **Traffic Selection Criteria:** Traffic selection is coarse, that is, all traffic on a particular segment traverse service functions whether the traffic requires service enforcement or not. This lack of traffic selection is largely due to the topological nature of service deployment since the forwarding topology dictates how (and what) data traverses service function(s). In some deployments, more granular traffic selection is achieved using policy routing or access control filtering. This results in operationally complex configurations and is still relatively inflexible.
10. **Limited End-to-End Service Visibility:** Troubleshooting service related issues is a complex process that involve network and service expertise. This is especially the case when service chains span multiple DCs, or across administrative boundaries such as externally consumable service chain components. Furthermore, the physical and virtual environments (network and service), can be highly divergent in terms of topology and that topological variance adds to these challenges.
11. **Per-Service (re)Classification:** Classification occurs at each service, independent from previously applied service functions. These unrelated classification events consume resources per service. More importantly, the classification functionality often differs per service function and service function cannot leverage the results from other deployed network or service.
12. **Symmetric Traffic Flows:** Service chains may be unidirectional or bidirectional; unidirectional is one where traffic is passed through a set of service functions in one forwarding direction only. Bidirectional is one where traffic is passed through a set of service functions in both forwarding directions. Existing service deployment models provide a static approach to realizing forward and reverse service chain association most often requiring complex configuration of each network device throughout the forwarding path.
13. **Multi-vendor Service Functions:** Deploying service functions from multiple vendors often requires per-vendor expertise: insertion models differ, there are limited common attributes and inter-vendor service functions do not share information.

3. Service Function Chaining

Service chaining provides a framework to address the aforementioned problems associated with service deployments:

1. **Service Overlay:** Service chaining utilizes a service specific overlay that creates the service topology: the overlay creates a path between service nodes. The service overlay is independent of the network topology and allows operators to use whatever overlay or underlay they prefer and to locate service functions in the network as needed.

Within the service topology, service functions can be viewed as resources for consumption and an arbitrary topology constructed to connect those resources in a required order. Adding new service functions to the topology is easily accomplished, and no underlying network changes are required. Furthermore, additional service instances, for redundancy or load distribution, can be added or removed to the service topology as required.

Lastly, the service overlay can provide service specific information needed for troubleshooting service-related issues.

2. **Generic Service Control Plane (GSCP):** GSCP provides information about the available service functions on a network. The information provided by the control plane includes service network location (for topology creation), service type (e.g. firewall, load balancer, etc.) and, optionally, administrative information about the service functions such as load, capacity and operating status. GSCP allows for the formulation of service chains and disseminates the service chains to the network.
3. **Service Classification:** Classification is used to select which traffic enters a service overlay. The granularity of the classification varies based on device capabilities, customer requirements, and service functionality. Initial classification is used to start the service chain. Subsequent classification can be used within a given service chain to alter the sequence of service functions applied. Symmetric classification ensures that forward and reverse chains are in place.
4. **Dataplane Metadata:** Dataplane metadata provides the ability to exchange information between the network and service functions, service functions and service functions and service functions and the network. Metadata can include the result of antecedent classification, information from external sources or forwarding related data. For example, service functions utilize metadata, as required, for localized policy decision. A common approach to

service metadata creates a common foundation for interoperability between service functions, regardless of vendor.

4. Service Function Chaining Use Cases

The following sections provide high level overviews of several common service chaining deployments.

4.1. Enterprise Data Center Service Chaining

TBD

4.2. Mobility Service Chaining

TBD

5. Related IETF Work

The following subsections discuss related IETF work and are provided for reference. This section is not exhaustive, rather it provides an overview of the various initiatives and how they relate to network service chaining.

1. L3VPN[L3VPN]: The L3VPN working group is responsible for defining, specifying and extending BGP/MPLS IP VPNs solutions. Although BGP/MPLS IP VPNs can be used as transport for service chaining deployments, the service chaining WG focuses on the service specific protocols, not the general case of VPNs. Furthermore, BGP/MPLS IP VPNs do not address the requirements for service chaining.
2. LISP[LISP]: LISP provides locator and ID separation. LISP can be used as an L3 overlay to transport service chaining data but does not address the specific service chaining problems highlighted in this document.
3. NV03[NV03]: The NV03 working group is chartered with creation of problem statement and requirements documents for multi-tenant network overlays. NV03 WG does not address service chaining protocols.
4. ALTO[ALTO]: The Application Layer Traffic Optimization Working Group is chartered to provide topological information at a higher abstraction layer, which can be based upon network policy, and with application-relevant service functions located in it. The mechanism for ALTO obtaining the topology can vary and policy can apply to what is provided or abstracted. This work could be leveraged and extended to address the need for services discovery.
5. I2RS[I2RS]: The Interface to the Routing System Working Group is chartered to investigate the rapid programming of a device's routing system, as well as the service of a generalized, multi-layered network topology. This work could be leveraged and extended to address some of the needs for service chaining in the topology and device programming areas.

6. Summary

This document highlights problems associated with network service deployment today and identifies several key areas that will be addressed by the service chaining working group. Furthermore, this document identifies four components that are the basis for service chaining. These components will form the areas of focus for the working group.

7. Security Considerations

Security considerations are not addressed in this problem statement only document. Given the scope of service chaining, and the implications on data and control planes, security considerations are clearly important and will be addressed in the specific protocol and deployment documents created by the service chaining working group.

8. Acknowledgments

The authors would like to thank David Ward, Rex Fernando and Jim French for their contributions.

9. References

9.1. Normative References

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

9.2. Informative References

- [ALTO] "Application-Layer Traffic Optimization (alto)", <<http://datatracker.ietf.org/wg/alto/>>.
- [I2RS] "Interface to the Routing System (i2rs)", <<http://datatracker.ietf.org/wg/i2rs/>>.
- [L3VPN] "Layer 3 Virtual Private Networks (l3vpn)", <<http://datatracker.ietf.org/wg/l3vpn/>>.
- [LISP] "Locator/ID Separation Protocol (lisp)", <<http://datatracker.ietf.org/wg/lisp/>>.
- [NV03] "Network Virtualization Overlays (nvo3)", <<http://datatracker.ietf.org/wg/nvo3/>>.
- [RFC3022] Srisuresh, P. and K. Egevang, "Traditional IP Network Address Translator (Traditional NAT)", [RFC 3022](#), January 2001.
- [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.

Authors' Addresses

Paul Quinn
Cisco Systems, Inc.

Email: paulq@cisco.com

Jim Guichard
Cisco Systems, Inc.

Email: jguichar@cisco.com

Surendra Kumar
Cisco Systems, Inc.

Email: smkumar@cisco.com

Puneet Agarwal
Broadcom

Email: pagarwal@broadcom.com

Rajeev Manur
Broadcom

Email: rmanur@broadcom.com

Abhishek Chauhan
Citrix

Email: Abhishek.Chauhan@citrix.com

Nic Leymann
Deutsche Telekom

Email: n.leymann@telekom.de

Mohamed Boucadair
France Telecom

Email: mohamed.boucadair@orange.com

Christian Jacquenet
France Telecom

Email: christian.jacquenet@orange.com

Michael Smith
Insieme Networks

Email: michsmit@insiemenetworks.com

Navindra Yadav
Insieme Networks

Email: nyadav@insiemenetworks.com

Thomas Nadeau
Juniper Networks

Email: tnadeau@juniper.net

Ken Gray
Juniper Networks

Email: kgray@juniper.net

Brad McConnell
Rackspace

Email: bmconne@rackspace.com

Kevin Glavin
Riverbed

Email: Kevin.Glavin@riverbed.com