                 Problem Statement for Service Function Chaining

Abstract

   This document provides an overview of the issues associated with the
   deployment of service functions (such as firewalls, load balancers,
   etc.) in large-scale environments. The term "service function
   chaining" is used to describe the definition and instantiation of an
   ordered list of instances of such service functions, and the
   subsequent "steering" of traffic flows through those service
   functions.

   The set of enabled service function chains reflects operator service
   offerings and is designed in conjunction with application delivery
   and service and network policy.

   This document also identifies several key areas that the Service
   Function Chaining (SFC) working group will investigate to guide its
   architectural and protocol work and associated documents.

Status of This Memo

   This document is not an Internet Standards Track specification; it is
   published for informational purposes.

   This document is a product of the Internet Engineering Task Force
   (IETF). It represents the consensus of the IETF community. It has
   received public review and has been approved for publication by the
   Internet Engineering Steering Group (IESG). Not all documents
   approved by the IESG are a candidate for any level of Internet
   Standard; see Section 2 of RFC 5741.

   Information about the current status of this document, any errata,
   and how to provide feedback on it may be obtained at
   http://www.rfc-editor.org/info/rfc7498.

Copyright Notice

Table of Contents

1. Introduction

   The delivery of end-to-end services often requires various service
   functions including traditional network service functions (for
   example, firewalls and server load balancers), as well as
   application-specific features such as HTTP header manipulation.
   Service functions may be delivered within the context of an isolated
   user (e.g., a tenant) or shared amongst many users or user groups.

   Current deployment models for service functions are often tightly
   coupled to network topology and physical resources, thus resulting in
   relatively rigid and static deployments. The static nature of such
   deployments greatly reduces and, in many cases, limits the ability of
   an operator to introduce new or modify existing services and/or
   service functions. Furthermore there is a cascading effect: changing
   one or more elements of a service function chain often affects other
   elements in the chain and/or the network elements used to construct
   the chain.

   This issue is particular acute in elastic service environments that
   require relatively rapid creation, destruction, or movement of
   physical 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, post facto modification, and the movement of
   service functions and application workloads in the existing network.
   The service insertion model must also retain 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 (SFC), which is often
   referred to simply as "service chaining" (in this document, the terms
   will be used interchangeably). Section 3 of this document highlights
   three key areas of WG focus for investigating solutions that address
   the current problems. The document highlights three key areas of WG
   focus for addressing the issues highlighted in this document that
   will form the basis for the possible WG solutions that address the
   current problems.

1.1. Definition of Terms

   Classification: Locally instantiated matching of traffic flows
      against policy for subsequent application of the required set of
      network service functions. The policy may be customer, network,
      or service specific.

   Network Overlay: A logical network built, via virtual links or
      packet encapsulation, over an existing network (the underlay).

   Network Service: An offering provided by an operator that is
      delivered using one or more service functions. This may also be
      referred to as a composite service. The term "service" is used to
      denote a "network service" in the context of this document.

      Note: Beyond this document, the term "service" is overloaded with
      varying definitions. For example, to some a service is an
      offering composed of several elements within the operator's
      network, whereas for others a service, or more specifically a
      network service, is a discrete element such as a firewall.
      Traditionally, such services (in the latter sense) host a set of
      service functions and have a network locator where the service is
      hosted.

   Service Function: A function that is responsible for specific
      treatment of received packets. A service function can act at
      various layers of a protocol stack (e.g., at the network layer or
      other OSI layers). As a logical component, a service function can
      be realized as a virtual element or be embedded in a physical
      network element. One or more service functions can be embedded in
      the same network element. Multiple occurrences of the service
      function can exist in the same administrative domain.

      A non-exhaustive list of service functions includes: firewalls,
      WAN and application acceleration, Deep Packet Inspection (DPI),
      server load balancers, NAT44 [RFC3022], NAT64 [RFC6146], HTTP
      header enrichment functions, and TCP optimizers.

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

   Service Function Chain (SFC): A service function chain defines an
      ordered or partially ordered set of abstract service functions
      (SFs) and ordering constraints that must be applied to packets,
      frames, and/or flows selected as a result of classification. An
      example of an abstract service function is a firewall. The
      implied order may not be a linear progression as the architecture
      allows for SFCs that copy to more than one branch, and also allows
      for cases where there is flexibility in the order in which service
      functions need to be applied. The term "service chain" is often
      used as shorthand for "service function chain".

   Service Overlay: An overlay network created for the purpose of
      forwarding data to required service functions.

   Service Topology: The service overlay connectivity forms a service
      topology.

2. Problem Space

   The following points describe aspects of existing service deployments
   that are problematic and that the SFC working group aims to address.

2.1. Topological Dependencies

   Network service deployments are often coupled to network topology,
   whether it be physical, virtualized, or a hybrid of the two. For
   example, use of a firewall requires that traffic flow through the
   firewall, which means placing the firewall on the network path (often
   via creation of VLANs) or architecting the network topology to steer
   traffic through the firewall. Such dependency imposes constraints on
   service delivery, potentially inhibiting the network operator from
   optimally utilizing service resources, and reduces flexibility. This
   limits scale, capacity, and redundancy across network resources.

   These topologies serve only to "insert" the service function (i.e.,
   ensure that traffic traverses 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 Layer
   2 segments and/or IP subnets).

   As more service functions are required -- often with strict ordering
   -- topology changes are needed in "front" and "behind" 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.

   The topological coupling limits placement and selection of service
   functions: service functions are "fixed" in place by topology.
   Therefore, placement and service function selection that take into
   account network topology information such as load, new links, or
   traffic engineering are often not possible.

   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 additional
   interfaces to provide an alternate topology.

2.2. Configuration Complexity

   A direct consequence of topological dependencies is the complexity of
   the entire configuration, specifically in deploying service function
   chains. Simple actions such as changing the order of the service
   functions in a service function chain require changes to the logical
   and/or physical topology. However, network operators are hesitant to
   make changes to the network once services are installed, configured,
   and deployed in production environments for fear of misconfiguration
   and consequent downtime. All of this leads to very static service
   delivery deployments. 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.

2.3. Constrained High Availability

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

   An effect of topological dependency is that the availability of
   service functions is constrained.

2.4. Consistent Ordering of Service Functions

   Service functions are typically independent; service function_1
   (SF1)...service function_n (SFn) 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 service functions that are used
   to deliver a given service. Furthermore, altering the order of a
   deployed chain is complex and cumbersome.

2.5. Application of Service Policy

   Service functions rely on topology information such as VLANs or
   packet classification/reclassification 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. Topology-centric information often
   does not convey adequate information to the service functions,
   forcing functions to individually perform more granular
   classification. In other words, the topology information is not
   granular enough, and its semantics is often overloaded.

2.6. Transport Dependence

   Service functions can and will be deployed in networks with a range
   of network transports, including network under and overlays, such as
   Ethernet, Generic Routing Encapsulation (GRE), Virtual eXtensible
   Local Area Network (VXLAN), MPLS, etc. The coupling of service
   functions to topology may require service functions to support many
   transport encapsulations or for a transport gateway function to be
   present.

2.7. Elastic Service Delivery

   Given that the current state of the art for adding/removing service
   functions largely centers around VLANs and routing changes, rapid
   changes to the deployed service capacity (increasing or decreasing)
   can be hard to realize due to the risk and complexity of VLANs and/or
   routing modifications.

2.8. Traffic Selection Criteria

   Traffic selection is coarse; that is, all traffic on a particular
   segment traverses all service functions whether or not the traffic
   requires service enforcement. 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 which
   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 coarse and inflexible.

2.9. Limited End-to-End Service Visibility

   Troubleshooting service-related issues is a complex process that
   involves both network-specific and service-specific expertise. This
   is especially the case when service function chains span multiple
   data centers or cross administrative boundaries. 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.

2.10. Classification/Reclassification per Service Function

   Classification occurs at each service function, independent from
   previously applied service functions since there are limited
   mechanisms to share the detailed classification information between
   services. The classification functionality often differs between
   service functions, and service functions may not leverage the
   classification results from other service functions.

2.11. Symmetric Traffic Flows

   Service function chains may be unidirectional or bidirectional
   depending on the state requirements of the service functions. In a
   unidirectional chain, traffic is passed through a set of service
   functions in one forwarding direction only. Bidirectional chains
   require traffic to be passed through a set of service functions in
   both forwarding directions. Many common service functions such as
   DPI and firewalls often require bidirectional chaining in order to
   ensure flow state is consistent.

   Existing service deployment models provide a static approach to
   realizing forward and reverse associations of service function
   chains, most often requiring complex configuration of each network
   device throughout the SFC. In other words, the same complex network
   configuration must be in place for both "directions" of the traffic,
   effectively doubling the configuration and associated testing.
   Further, if partial symmetry is required (i.e., only some of the
   services in the chain required symmetry), the network configuration
   complexity increases since the operator must ensure that the
   exceptions -- the services that do not need the symmetry flow -- are
   handled correctly via unique configuration to account for their
   requirements.

2.12. Multi-vendor Service Functions

   Deploying service functions from multiple vendors often requires per-
   vendor expertise (insertion models differ, common attributes are few,
   and inter-vendor service functions do not share information), hence
   standards are needed to ensure interoperability.

3. Service Function Chaining

   Service function chaining aims to address the aforementioned problems
   associated with service deployment. Concretely, the SFC working
   group will investigate solutions that address the following elements.

3.1. Service Overlay

   Service function chaining utilizes a service-specific overlay that
   creates the service topology. The service overlay provides service
   function connectivity, built "on top" of the existing network
   topology. It allows operators to use whatever overlay or underlay
   they prefer to create a path between service functions 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.

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

3.2. 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 services offered.
Initial classification determines the service function chain required
to process the traffic. Subsequent classification can be used within
a given service function chain to alter the sequence of service
functions applied. Symmetric classification ensures that forward and
reverse chains are in place. Similarly, asymmetric -- relative to
required service function -- chains can be achieved via service
classification.

3.3. SFC Encapsulation

The SFC encapsulation enables the creation of a service chain in the
data plane and can convey information about the chain such as chain
identification and OAM status.

The SFC encapsulation also carries data-plane metadata that provides
the ability to exchange information between logical classification
points and service functions (and vice versa) and between service
functions. Metadata is not used as forwarding information to deliver
packets along the service overlay.

Metadata can include the result of antecedent classification and/or
information from external sources. Service functions utilize
metadata, as required, for localized policy decisions.

In addition to sharing of information, the use of metadata addresses
several of the issues raised in Section 2, most notably by decoupling
policy from the network topology, and by removing the need for
classification (and reclassification) per service function as
described in Section 2.10.

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

4. Security Considerations

   Although this problem statement does not introduce any protocols,
   when considering service function chaining, the three main areas
   begin investigated (see Section 3) by the WG have security aspects
   that warrant consideration.

   Service Overlay: The service overlay will be constructed using
      existing transport protocols (e.g., MPLS, VXLAN) and as such is
      subject to the security specifics of the transport selected. If
      an operator requires authenticity and/or confidentiality in the
      service overlay, a transport (e.g., IPsec) that provides such
      functionally can be used.

   Classification: Since classification is used to select the
      appropriate service overlay and the required service encapsulation
      details, classification policy must be both accurate and trusted.
      Conveying the policy to an SFC edge node (a node that forms the
      logical boundary of an SFC domain) may be done via a multitude of
      methods depending on an operator's existing provisioning practices
      and security posture.

      Additionally, traffic entering the SFC domain and being classified
      may be encrypted, thus limiting the granularity of classification.
      The use of pervasive encryption varies based on type of traffic,
      environment, and level of operator control. For instance, a large
      enterprise can mandate how encryption is used by its users,
      whereas a broadband provider likely does not have the ability to
      do so.

      The use of encrypted traffic, however, does not obviate the need
      for SFC (nor the problems associated with current deployment
      models described herein); rather, when encrypted traffic must be
      classified, the granularity of such classification must adapt. In
      such cases, service overlay selection might occur using outer
      (i.e., unencrypted) header information (in the presence of
      encryption) or external information about the packets.

   SFC Encapsulation: As described in Section 3, the SFC encapsulation
      carries information about the SFC and data-plane metadata.
      Depending on the environment and security posture, the SFC
      encapsulation might need to be authenticated and/or encrypted.
      The use of an appropriate overlay transport (as described above)
      can provide data-plane confidentiality and authenticity.

The exchange of SFC encapsulation data such as metadata must
originate from trusted source(s). Also, if needed, authentication
and confidentiality protection should be provided during the
exchange to the various SFC nodes.

SFC and Multi-tenancy: If tenant isolation is required in an SFC
deployment, an appropriate network transport overlay that provides
adequate isolation and identification can be used. Additionally,
tenancy might be used in the selection of the appropriate service
chain; however, as stated, the network overlay is still required
to provide transport isolation. SF deployment and how specific
SFs might or might not be allocated per tenant are outside the
scope of this document.

The SFC Architecture document [SFC-ARCH] presents a more complete
review of the security implications of a complete SFC architecture.

5. Informative References

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

[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,
            <http://www.rfc-editor.org/info/rfc6146>.

[SFC-ARCH]
            Halpern, J. and C. Pignataro, "Service Function Chaining
            (SFC) Architecture", Work in Progress, draft-ietf-sfc-
            architecture-07, March 2015.

Acknowledgments

Contributors

   The following people are active contributors to this document and
   have provided review, content and concepts (listed alphabetically by
   surname):

   Puneet Agarwal
   Broadcom
   EMail: pagarwal@broadcom.com

   Mohamed Boucadair
   France Telecom
   EMail: mohamed.boucadair@orange.com

   Abhishek Chauhan
   Citrix
   EMail: Abhishek.Chauhan@citrix.com

   Uri Elzur
   Intel
   EMail: uri.elzur@intel.com

   Kevin Glavin
   Riverbed
   EMail: Kevin.Glavin@riverbed.com

   Ken Gray
   Cisco Systems
   EMail: kegray@cisco.com

   Jim Guichard
   Cisco Systems
   EMail:jguichar@cisco.com

   Christian Jacquenet
   France Telecom
   EMail: christian.jacquenet@orange.com

   Surendra Kumar
   Cisco Systems
   EMail: smkumar@cisco.com

   Nic Leymann
   Deutsche Telekom
   EMail: n.leymann@telekom.de

   Darrel Lewis
   Cisco Systems
   EMail: darlewis@cisco.com

   Rajeev Manur
   Broadcom
   EMail:rmanur@broadcom.com

   Brad McConnell
   Rackspace
   EMail: bmcconne@rackspace.com

   Carlos Pignataro
   Cisco Systems
   EMail: cpignata@cisco.com

   Michael Smith
   Cisco Systems
   EMail: michsmit@cisco.com

   Navindra Yadav
   Cisco Systems
   EMail: nyadav@cisco.com

Authors' Addresses

   Paul Quinn (editor)
   Cisco Systems, Inc.

   EMail: paulq@cisco.com

   Thomas Nadeau (editor)
   Brocade

   EMail: tnadeau@lucidvision.com