USE CASE FOR HANDLING DYNAMIC CHAINING AND SERVICE INDIRECTION
              draft-purkayastha-sfc-service-indirection-00

Abstract

   Many stringent requirements are imposed on today's network, such as
   low latency, high availability and reliability in order to support
   several use cases such as IoT, Gaming, Content distribution, Robotics
   etc. Networks need to be flexible and dynamic in terms of allocation
   of services and resources. Network Operators should be able to
   reconfigure the composition of a service and steer users towards new
   service end points as user move or resource availability changes.
   SFC allows network operators to easily create and reconfigure service
   function chains dynamically in response to changing network
   requirements. We discuss a use case where Service Function Chain can
   adapt or self-organize as demanded by the network condition without
   requiring SPI re-classification. This can be achieved, for example,
   by decoupling the service consumer and service endpoint by a new
   service function proposed in this draft. We describe few
   requirements for this service function to enable dynamic switching
   between consumer and end point.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

   The requirements on today's networks are very diverse, enabling
   multiple use cases such as IoT, Content Distribution, Gaming, Network
   functions such as Cloud RAN. Every use case imposes certain
   requirements on the network. These requirements vary from one
   extreme to other and often they are in a divergent direction.
   Network operator and service providers are pushing many functions
   towards the edge of the network in order to be closer to the users.
   This reduces latency and backhaul traffic, as user request can be
   processed locally.

   It becomes more challenging for the network when user mobility as
   well as non-deterministic availability of compute and storage
   resources are considered. The impact is felt most in the edge of the
   network because as the users move, their point of attachment changes
   frequently, which results in (at least partially) relocating the
   service as well as the service endpoint. Furthermore, network
   functions are pushed more and more towards the edge, where compute

and storage resources are constrained and availability is non-
deterministic. Also, storage resources may need to be moved where
the user concentration is more in case of content delivery
applications.

Take the following video orchestration service example from ETSI MEC
Requirements document [ETSI_MEC]. The proposed use case of edge
video orchestration suggests a scenario where visual content can be
produced and consumed at the same location close to consumers in a
densely populated and clearly limited area. Such a case could be a
sports event or concert where a remarkable number of consumers are
using their handheld devices to access user select tailored content.
The overall video experience is combined from multiple sources, such
as local recording devices, which may be fixed as well as mobile, and
master video from central production server. The user is given an
opportunity to select tailored views from a set of local video
sources.

In such a dynamic network environment, the capability to dynamically
compose new services from available services as well as move a
service instance in response to user mobility or resource
availability is desirable. SFC allows network operators as well as
service providers to compose new services by chaining individual
service functions towards the composed new service. In a dynamic
network environment where service functions move frequently because
of user movement, load balancing or resource modification, service
function chains and the service end points need to be created and
recreated frequently. SFC, as defined in IETF, is capable of
modifying the service chain dynamically in response to network
conditions.

In this document we address this dynamicity by introducing a special
Service Function, called SRR (service request routing). We describe
the problems associated with today's network and Layer 3 based
approach to handle dynamicity in the network. We then discuss how
such new Service Function with certain capabilities can handle the
dynamicity better than these conventional methods.

2. NSH and Re-classification

   [RFC7498] captures the problems associated with existing service
   deployments that are problematic. High level problems are listed
   below.

   o Network topology: Network service deployment is tightly coupled
      with network topology thus reducing the flexibility in service
      delivery. It adds complexity in deploying network service when

certain traffic types may need some service and other traffic types do not need the same service.

o Configuration complexity is the direct result of dependency on network topology.

o Limited availability of services

o Altering the order of a deployed chain is complex and cumbersome

o Coupling of service functions to topology may require service functions to support many transport encapsulations or for a transport gateway function to be present.

o In a dynamic environment like the Edge of a network service delivery, routing changes fast. It may be difficult to deliver service dynamically due to the risk and complexity of VLANs and/or routing modifications.

These factors provide motivation for a simplified and flexible service insertion model that addresses many of the current shortcomings and provides new, much needed functionality to enable service deployments in modern network environments. Service chaining accomplishes this by considering service functions as resources, with associated attributes, available for scheduled consumption. Selective traffic, subject to policy, may then be "steered" to the requisite service resources, along with any "extra" information referred to as metadata. This metadata is used for policy enforcement.

A basic form of service chaining may be realized using existing transport encapsulations. This method of chaining relies upon the tunneling of selected data between service functions. Although this form of service chaining achieves some level of abstraction from the underlying topology, it does not truly create a service plane. NSH [I-D.ietf-sfc-nsh] is a distinct identifiable plane that can be used across all transports to create a service chain and exchange metadata along the chain.

2.1. Dynamic service chain creation using NSH

We revisit the dynamic service chain creation capability of NSH. NSH defines a new service plane protocol [I-D.ietf-sfc-nsh]. A Network Service Header (NSH) contains service path information and optionally metadata that are added to a packet or frame and used to create a service plane. A control plane is required in order to exchange NSH values with participating nodes, and to provision the same nodes with requisite information such as service path ID to overlay mapping.

The Network Service Header has three parts, Base header, Service Path
Header and Context Header. NSH Service Path Header is a 4-byte
service path header follows the base header and defines two fields
used to construct a service path:

o Service path identifier (SPI)

o Service index (SI)

The following figure depicts the service path header.

```
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Service Path ID                                | Service Index |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
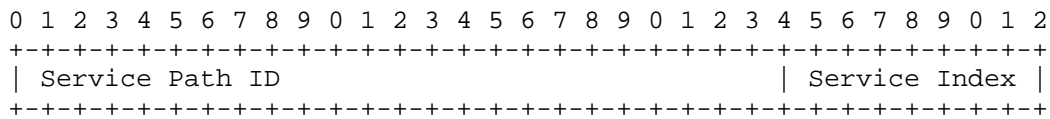
Figure 1: NSH Path Header

The service path identifier (SPI) is used to identify the service
path that interconnects the needed service functions. It allows
nodes to utilize the identifier to select the appropriate network
transport protocol and forwarding techniques. The service index (SI)
identifies the location of a packet within a service path. As
packets traverse a service path, the SI is decremented post-service.

SPI represents the service path and altering the path identifier
results in a change of a service path. A change in SPI value is a
result of re-classification. It means a node in the service path
determined, based on policy, that the initial classification was
incorrect or incomplete. If the updated classification results in
the necessity of a new service path, the node updates the SPI and SI
fields accordingly. The new identifier is then used to select the
appropriate overlay topology. This allows service functions to alter
the path of a packet without having to participate in the network
topology and its associated control plane(s). The method to
determine that an existing classification is incorrect and how to
determine the new classification is not defined.

3. Challenges with dynamic indirection

The emerging trend in today's network is to deploy network functions,
services and applications at the edge of the network to support
latency requirements, computational offload, traffic optimization
etc. As users are moving, application or services being used by
users, may need to be moved closer to the user's new location. This
implies another instance of the service function may need to be
instantiated close to the user's new location. It may result in re-

establishing service path from the newly instantiated service
function to other service instances. It is also possible that the
newly instantiated service function may be redirected to a new
service end point (e.g. Application Server) for various reasons,
such as incomplete content, proximity to data store, load balancing
etc. In another scenario, a single instance of the service function
may not handle all users. A single service function may be
instantiated more than once to balance user load. As the number of
instances increase and along with mobility, the complexity of service
routing increases. It is anticipated that there may be a constant
action of function chaining, re-chaining occurring in the network.

The challenge of dynamic indirection may be better described by
analyzing the working of CDNs, which dynamically (re-)direct user-
initiated requests towards the most appropriate content instance.
This task becomes more difficult if granularity of the instance
placement increases. For instance, in case of a CDN being realized
close to end users, specifically in edge of the network, the specific
content instance might need to be selected dynamically. After
initial selection, the instance may change during service execution.

In a conventional network, an instance of a service is found and
selected using DNS. The subsequent service request is then routed
through the network between the client and the service. If the user
is doing a DNS lookup to access content served by a CDN then the DNS
service will maintain a list of IP addresses that can be returned for
a given domain name and will try to return an IP address of a node
geographically close to the client. Should the service provider want
to replace an instance of their service with another one at a
different IP address (and potentially a different physical location
for various reasons such as load balancing, reliability etc.) then
the DNS tables must be updated, i.e., the service needs to be
(re-)registered quickly. This is done by updating the local
authoritative DNS server which then propagates the new mapping to DNS
services across the world. DNS propagation can take up to 48 hours
so fast and dynamic switching from one service instance to another is
not possible in conventional networks. When relying on many
surrogate service endpoints to exist in the edge network, there is a
clear issue of certain resources not being available in one surrogate
instance while existing in another so that changes in redirection
might be desirable, while also changes in local load drive the need
for such change in redirection.

The other issue in conventional network lies with mobility management
procedure. These procedures use an anchor point, which terminates a
session at the network edge. As user moves around, traffic is
redirected from the anchor point to the new point of attachment.
Relying on typical mobility management approaches found in IP

networks, usually leads to inefficient 'triangular' routing of
requests through this common 'anchor' point. This triangular routing
increases the latency in reaching the new service function or service
end points as users move.

Traffic steering is a common procedure in managed networks,
particularly at the edge, due to desired subscriber-centric traffic
policies (e.g., related to pricing structures), resource requirements
(e.g., related to using particular paths in the network) or mobility
(e.g., users moving in a cellular network). Today's methods for
traffic steering include anchor-based mobility management as well as
traffic classification, for instance, in packet gateways of cellular
systems (using, e.g., deep packet inspection as well as port and
address classification). While the former leads to inefficient
'triangular' traffic forwarding, the latter often requires additional
state in the forwarders to differentiate traffic from one user to
another.

The analysis of CDN network shows that dynamic indirection is a
necessary requirement, which needs to be supported by the networks.
The goal for this indirection is to provide user applications lowest
possible latency. But as discussed above, relying on today's
technique, does not help in guaranteeing same latency to user
applications. On the other hand, there is a high possibility that
latency may increase if we rely on Layer 3 based service redirection
techniques.

SFC handles indirection through the use of SPI. A packet needs to be
reclassified and the intermediate node changes the SPI. Following
are the typical steps that happens in order to implement the
indirection.

o A packet arrives at a particular node

o The node contacts the policy manager

o Identifies the current classification is incorrect

o Reclassifies the packet, i.e. change the SPI

o Inserts the packet in the pipe, possibly towards the SFF

The indirection mechanism in SFC involves certain steps to process
policy information and change the SPI in the packet header, making it
suitable to handle dynamic indirection requirements. Our proposed SF
in this document provides an additional method to handle dynamic
indirection of service requests, not relying on the reclassification

mechanism. Combining these two techniques may provide flexibility
and improvement over single method.

4. Desired Features

   In order to route the service requests to service end points in a
   dynamic manner, we identify the following desirable features:

   o Fast switching from one service instance to another by not relying
      on DNS for service location resolution. Instead of DNS, the
      function should be able to identify the path, which will allow to
      reach the service end point.

   o Direct path mobility, where the path between the requester and the
      responding service can be determined as being optimal (e.g.,
      shortest path or direct path to a selected instance), is needed to
      avoid the use of anchor points and reduce service-level latency

   o Indirect service requests at the network level, transparent to the
      requesting client and without the involvement of the DNS. End
      user is not aware of the decision made by the SF.

   o New methods for forwarding, such as path-based forwarding, direct
      path routing in mobility cases, path pinning for traffic steering
      and simplified service-specific peering towards the Internet.

5. Service Request Routing (SRR) Service Function

   The following diagram shows the application of the new proposed SRR
   service function in an example of media clients connecting to media
   servers. There may be more than one media functions to support CDN
   like architecture, Surrogate servers to handle mobility and load
   balancing.

```
                                              +--------+
                                              |        |
              |------------------------|----------------+  SRR   +
              |                        |                |        |
              |                        |                +---/|\--+
              |                        |                     |
  +---\|/--+    +---------+    +--\|/--+    +------+    +----+---+
  |       |    |         |    |       |    |      |    |        |
  + Client +-->+   IP    +-->+ Media +-->+ SRR  +-->+ Media  +
  |       |    | Routing |    | Fn1   |    |      |    | Fn2    |
  +-------+    +---------+    +-------+    +------+    +--------+
```
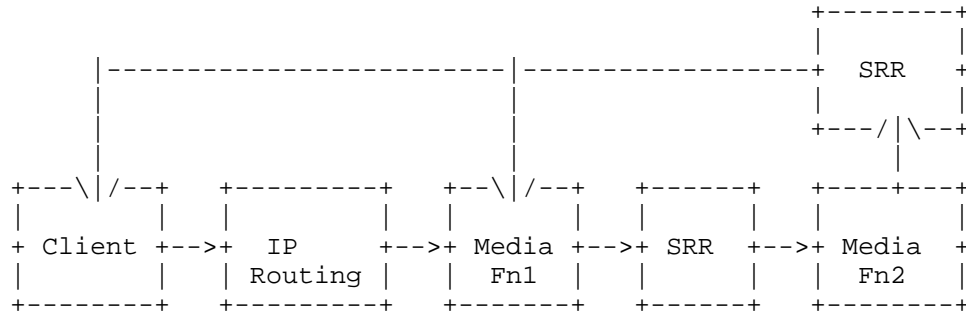
                  Figure 2: Use of SRR function

   The clients are connected to media functions through frontend routed
   network, e.g., relying on standard IP routing, while media functions
   are chained via the new proposed service request routing (SRR)
   function. Alternatively, we also envision to utilize the SRR
   function directly between client SF and media function SF, as
   outlined in the figure below

```
                                              +--------+
                                              |        |
              |------------------------|----------------+  SRR   +
              |                        |                |        |
              |                        |                +---/|\--+
              |                        |                     |
  +---\|/--+    +---------+    +--\|/--+    +------+    +----+---+
  |       |    |         |    |       |    |      |    |        |
  + Client +-->+   SRR   +-->+ Media +-->+ SRR  +-->+ Media  +
  |       |    |         |    | Fn1   |    |      |    | Fn2    |
  +-------+    +---------+    +-------+    +------+    +--------+
```
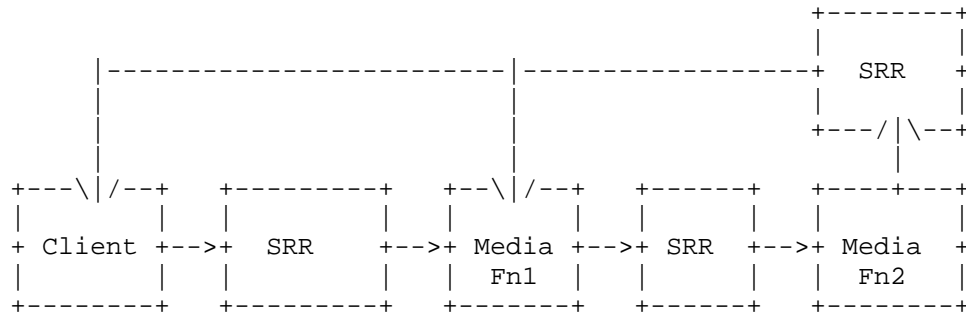
             Figure 3: SRR function between Client and Media Function

   The SRR service function decouples clients from media functions.
   This brings in flexibility in routing service requests from client to
   service end points. In the edge network, where users are moving and
   service end points may also change, having flexibility to decide and
   steer service requests directly helps in guaranteeing the same
   latency to user applications. Clearly, that is achieved by reducing
   the switching time from SF to another. As service end point changes,
   the routing functions makes instantaneous decision to route the
   request to the appropriate media server.

The possible improvements of using SRR within an SFC framework are
listed below:

o Fast (between 10 and 20ms) switching times from one service
   instance to another by not relying on the DNS for service
   discovery and directly routing service requests at the level of
   the transport network.

o The capability to indirect service requests at the network level
   will help in reducing latency, when service end points change.
   E.g. when a service request is being sent to one surrogate
   instance but results in a HTTP 404 or 5xx error response, the
   original request is redirected to another alternative surrogate
   with minimal latency, i.e., right at the destination of said
   failed service request. Nesting these operations effectively
   leads to a net-level 'search' among all available surrogate
   instances until the search is exhausted (with a negative result)
   or the resource is found.

o New methods for forwarding, such as path-based forwarding, will
   enable direct path routing in mobility cases, path pinning for
   traffic steering and simplified service-specific peering towards
   the Internet. Such capability would allow for localizing traffic,
   reduce latency and costs.

6. Next Steps

   Does the WG see value in supporting the requirements for SFC to
   enable routing of service requests between service consumers and
   service endpoints in a dynamic manner as outlined in Section 4?

7. IANA Considerations

   This document requests no IANA actions.

8. Security Considerations

   TBD.

9. Informative References

   [ETSI_MEC]
             ETSI, "Mobile Edge Computing (MEC), Technical
             Requirements", GS MEC 002 1.1.1, March 2016,
             <http://www.etsi.org/deliver/etsi_gs/
             MEC/001_099/002/01.01.01_60/gs_MEC002v010101p.pdf>.

   [I-D.ietf-sfc-nsh]
              Quinn, P. and U. Elzur, "Network Service Header", draft-
              ietf-sfc-nsh-13 (work in progress), June 2017.

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

Authors' Addresses

   Debashish Purkayastha
   InterDigital Communications, LLC
   Conchoken
   USA

   Email: Debashish.Purkayastha@InterDigital.com

   Akbar Rahman
   InterDigital Communications, LLC
   Montreal
   Canada

   Email: Akbar.Rahman@InterDigital.com

   Dirk Trossen
   InterDigital Communications, LLC
   64 Great Eastern Street, 1st Floor
   London EC2A 3QR
   United Kingdom

   Email: Dirk.Trossen@InterDigital.com
   URI: http://www.InterDigital.com/