

Network Working Group
Internet-Draft
Intended status: Informational
Expires: September 2, 2018

D. Purkayastha
A. Rahman
D. Trossen
InterDigital Communications, LLC
Z. Despotovic
R. Khalili
Huawei
March 1, 2018

Alternative Handling of Dynamic Chaining and Service Indirection
draft-purkayastha-sfc-service-indirection-02

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

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 <https://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 September 2, 2018.

Copyright Notice

Copyright (c) 2018 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 (<https://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 | 2 |
| 2. Use Case Description | 3 |
| 2.1. Data Center | 3 |
| 2.2. Third party cloud service provider | 4 |
| 2.3. ETSI MEC USE CASE | 5 |
| 2.4. 3GPP | 6 |
| 2.5. Use Case Analysis | 6 |
| 3. NSH and Re-classification | 8 |
| 3.1. Dynamic service chain creation using NSH | 9 |
| 4. Challenges with dynamic indirection | 10 |
| 5. HTTP as a transport | 12 |
| 6. Service Request Routing (SRR) Service Function | 14 |
| 6.1. Overview | 14 |
| 6.2. Details of SRR Function | 16 |
| 7. Protocol Consideration | 21 |
| 8. Next Steps | 21 |
| 9. IANA Considerations | 21 |
| 10. Security Considerations | 22 |
| 11. Informative References | 22 |
| Authors' Addresses | 23 |

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

We describe few use cases in the next section and derive the requirement for composing new services and service path in a dynamic edge network. 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. Use Case Description

2.1. Data Center

The data center use case draft [I-D.ietf-sfc-dc-use-cases] describes an East West traffic use case. This is the predominant traffic in data centers today. Server virtualization has led to the new paradigm where virtual machines can migrate from one server to another across the data center. This explosion in east-west traffic is leading to newer data center network fabric architectures that provide consistent latencies from one point in the fabric to another.

SFCs applied in an enterprise or service provider data center can be broadly categorized into two types:

- o Access SFCs
- o Application SFCs

Access SFCs are focused on servicing traffic entering and leaving the data center while Application SFCs are focused on servicing traffic destined to applications. Service providers deploy a single "Access SFC" and multiple "Application SFCs" for each tenant. Enterprise

data center operators on the other hand may not have a need for Access SFCs depending on the size and requirements of the enterprise.

In carrier networks, operators may deploy multiple data centers dispersed geographically. Each data center may host different types of service functions. For example, latency sensitive or high usage service functions are deployed in regional data centers while other latency tolerant, low usage service functions are deployed in global or central data centers. In such deployments, SFCs may span multiple data centers and enable operators to deploy services in a flexible and inexpensive way.

It is clear that within the data center as well as in inter data center scenarios, users are serviced by multiple SFs distributed inside as well as outside a location. In this scenario, it is clear that Service function chains should be able to reselect, redirect traffic very fast. The draft identifies that Static service chains do not allow for modifying the SFCs as they require the ability to add SNs or remove SNs to scale up and down the service capacity. Likewise the ability to dynamically pick one among the many SN instance is not available.

2.2. Third party cloud service provider

This use case is related to an emerging business model, where computational resources for edge cloud service are provided by alternative facility providers that are non-traditional network operators. This is due to the situation for many specific localized use cases, where network operators may not have necessary real estate available. They may even not be willing to spend on CAPEX and OPEX for said point-of-presence, because there is no clear path for sustainable cost recovery [UKNIC].

The industry is witnessing the emergence of real estate owners such as building asset or management companies, cell tower owners, railway companies or other facility owners willing to deploy edge cloud resources. The facility provider, e.g. cell tower owner or building management company, deploys edge computing resources throughout their installation in the country. They have their own operation and management software, which is capable of resource deployment, scale up or scale down resources, deploy edge applications from third party service providers. They are capable of offering service to more than one network operator at a specific location, thus acting as a "neutral host". The facility provider, which owns cloud resources and provides application services, is referred to as "Third party Edge Owner (TEO)".

There is more than one stakeholder in this ecosystem, E.g. Network Service Provider, Real estate owner, Cloud capability (compute and storage resource) provider, Application/service provider. An entity can assume more than one role. From network operators point of view there may be "Cloud provider" or "Cloud service provider" depending on the roles assumed by external entity.

"Cloud Providers" provide cloud resources (compute and storage) to network operators. Network operators rent those resources and manage MEC host by themselves. Network operator can set up application traffic rules, so that traffic can be processed, by that host.

"Cloud Service Providers" not only make resources available to network operators or service providers, but also provides management and hosting service. They can host edge applications on behalf of application service providers and sets up user plane traffic to be steered towards the edge application.

Cloud Service Providers, as well as many organizations that need to share and analyze a quickly growing amount of data, such as retailers, manufacturers, telcos, financial services firms, and many more, are turning to localized Micro Data Centers(MDC) installed on the factory floor, in the telco central office, the back of a retail outlet, etc. The solution applies to a broad base of applications that require low latency, high bandwidth, or both.

As Micro Data centers are deployed at the edge of the network, common deployment options are:

- o Micro Data Centers are deployed on L2 in the edge of the network
- o Instead of single internet Point Of Presence (POP) deployment, multiple internet POP deployment is desirable to localize data
- o Service is composed out of these multiple POP deployment of MDC, where data exchange and collaboration is expected among these MDCs
- o Due to mobility, changes in network condition (e.g. congestion, load), service composition may change frequently to support promised quality of experience

2.3. ETSI MEC USE CASE

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.

2.4. 3GPP

3GPP Rel. 15 introduces the notion of the service-based interface (SBI) as an alternative to the traditional call pattern invocation of network functions. This introduction targets the support for replication, e.g., driven by virtualized functions, as well as supporting alternative interactions, e.g., for different vertical market specific control planes, by making the discovery as well as composition of new interactions more flexible.

We believe that SFC is a suitable framework for the interconnection of such network functions through the new SBI. One of the aforementioned driving forces, namely the replication of functions aligns with our thinking in this draft in that indirections to new vertical instances need to be dynamic in reacting to the appearance of new virtual instances or to changes in policies for the selection of specific instances by specific calling entities.

2.5. Use Case Analysis

SFC allows network operators as well as service providers to compose new services by chaining individual service functions.

In a dynamic network environment, like the edge of a network, the capability to dynamically compose new services from available services as well as move a service instance is desirable. Dynamic composition and relocation of services may be attributed to:

- o Congestion in the network: Due to constrained network resources, increase in the network load may create congestion in the network, resulting in a congested Service Function Path. Service functions may detect congestion and reconfigure the Service Function Path to avoid it.
- o In response to latency: in a dynamic network environment and with the need for ultra-low latency communication, instantiation of new service function endpoints might be the only remedy to combat the increase of latency caused, e.g., by increased load on a previous endpoint or mobility of the user and therefore increasing the 'distance' to the service function endpoint. Keeping the service

function endpoint 'close' to the user allows for reducing latency, segregating communication in localized islands of service interaction.

- o In response to user mobility: 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
- o Resource availability.: Availability of compute and storage resources varies with network load, number and type of applications running etc. In the edge of the network, due to sudden increase of users, compute load may increase. In this situation applications, running on the compute resources may be moved to another location where more resources are available.

In SFC, there is a notion of logical chaining of SFs and chaining of actual physical locations, known as Rendered Service Path (RSP). RSP provides a static binding of SFs to their physical location. In order to create a chain in dynamic fashion, late binding of SFs and physical location may be desired. SFC is capable of modifying the service chain to certain extent in response to network conditions, but not a complete solution has been described

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

- o Capability to trigger service chain reconfiguration based on network information such as congestion indication, mobility, degradation of user experience etc. Service Functions should be able to process such network information, identify which section of the chain needs to be reconfigured and take action
- o Fast switching from one service instance to another by not relying on the 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 further 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.

3. NSH and Re-classification

[RFC7498] captures the problems associated with existing service deployments that are problematic. The problems are described below at a high level:

- 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 [RFC8300] is a distinct identifiable plane that can be used across all transports to create a service chain and exchange metadata along the chain.

Fundamentally, however, the notion of "services" in SFC is tied into specific service function endpoints, which lie along a well-defined service function path (SFP) where the path is defined through lower layer transport encapsulations. If any such service function endpoint changes, the service chain needs to be adjusted; a procedure we outline in the following sub-section.

3.1.1. Dynamic service chain creation using NSH

We revisit the dynamic service chain creation capability of NSH. NSH defines a new service plane protocol [RFC8300]. 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.

4. 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 due to latency or load constraints. 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; even in more localized scenarios, the propagation of DNS updates might still be insufficient. 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. With the emergence of container-based virtualization platforms, service function endpoints can be established in a matter of seconds and we therefore believe that the 'reachability' of such said service instance, i.e., the possibility of route service requests to it from a client that was previously served elsewhere, must follow a similar timeline, i.e., a few seconds or even less.

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.

5. HTTP as a transport

With the extensive use of "web technology", "distributed services" and availability of heterogeneous network, HTTP has effectively transitioned into the common transport for name-based E2E communication across the web. In the context of SFC and SF, HTTP requests and response are considered as the "Service Request (SR)". This use case describes how these SRs are directed towards correct SF in a fast and dynamic way. The routing and indirection of SRs are abstracted at HTTP level, instead of the traditional approach where routing decision for a service request is made at Layer 3.

If we abstract HTTP as a transport, HTTP requests, such as GET, PUT and POST can be routed based on the URI associated with the request, with the URI being simply the name of a resource or the invocation point for a service transaction. Based on the name of the resource requested, the appropriate HTTP request can be routed to the suitable service endpoint. If Service Functions (SF) could be identified using URI or name, HTTP requests to an SF would be routed or directed using name based routing. With that, the redirection to the most suitable service instance is purely done based on named services with HTTP being a specific (application layer) transport service.

The ongoing EU H2020 efforts like FLAME [H2020FLAME] are driven by city-scale many-POP deployments of compute infrastructure, all SDN-connected and OpenStack managed. Localized media use cases drive the need for name-based (HTTP as the main transport protocol here) service instances being chained with the relationship between specific virtual instances being controlled at the underlying routing/switching level.

The notion of 'HTTP as-a transport', utilizing URLs as addressing scheme, can be used to create SFP as shown in Fig 2., i.e., 192.168.x.x -> www.example.com -> 192.168.x.x -> www.example2.com -> 192.168.x.x -> ... -> www.exampleN.com. It is this 'name-based' relationship that we see possibly realized through specific replicated instances, where in turn the routing towards those specific instances is realized by the SRR.

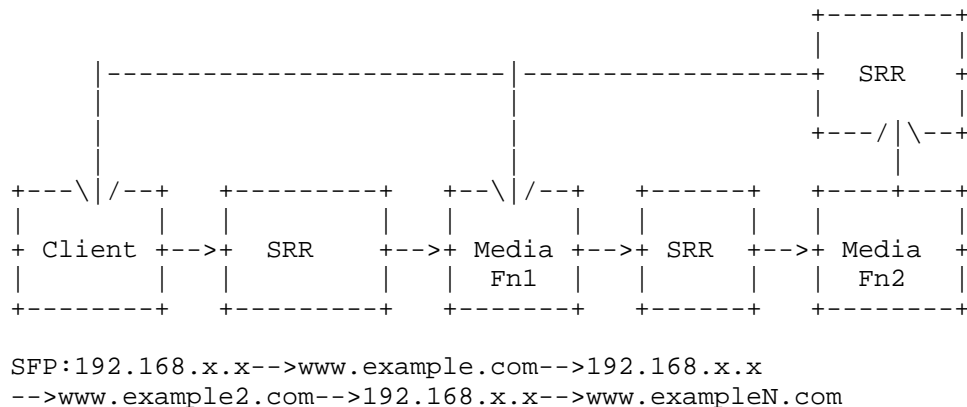


Figure 2: SFP with new HTTP-based Transport option

In a pure SFC architectural framework, Classifier function may interact with SRR to obtain an SE (Service Encapsulation). E.g. the

Classifier function may look into the network locator map in Fig 2 and determine the next SF is www.example.com. It provides this information to SRR to obtain the next hop information. SRR returns the SE for next hop, which can be a "bitfield" information that is being used in the overlay routing for this part of the SFP. The Classifier function uses this SE to route the incoming packet directly at the transport network level.

6. Service Request Routing (SRR) Service Function

6.1. Overview

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.

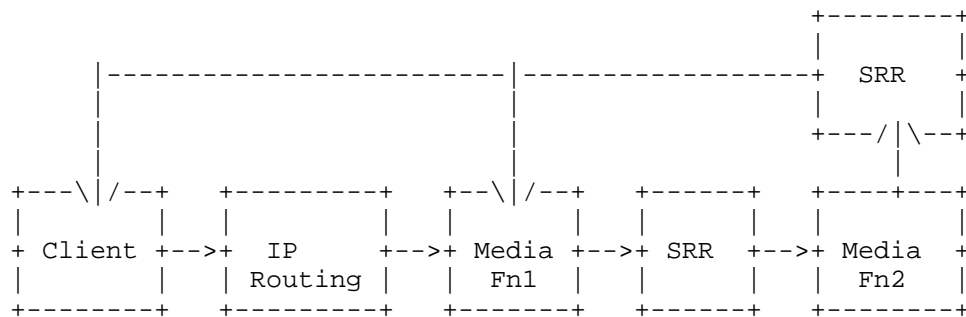


Figure 3: General SFC with SRR Flexible Chaining, initiated via IP Routed Client Connection

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

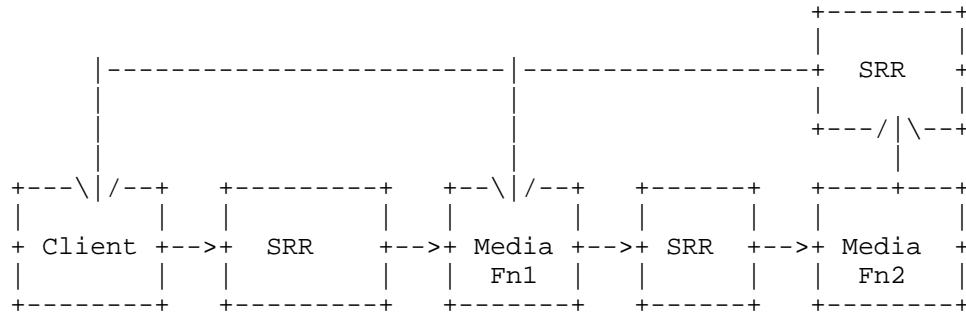


Figure 4: General SFC with SRR Flexible Chaining, initiated via SRR Chained Client

For our considerations, we assume that each SF is realized by at least one or more service function endpoints (SFEs). Hence, instead of looking at "chaining" as a concept that connects specific SFEs along a well-defined SFP, we propose to look at "chaining" at the level of "named" service functions rather than their specific endpoint instances. With this in mind, the SRR service function lifts the relationship between the connecting SFs to the level of "logical" service functions rather than their specific realizing endpoints. Instead of relying on dynamic re-chaining in case of any dynamically changing relationship between specific SFEs, the SRR provides the selection of suitable SFEs while maintaining the logical relationship between the SFs. In Section 6.3, we will present the necessary extensions to the SFP concept to support this higher abstraction of "chaining" via "named" logical SFs. The SRR introduces the flexibility in routing service requests from client to specific SFEs. 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 SRR introduces the flexibility in routing service requests from client to specific SFEs in response to conditions such as congestion in the network, user mobility etc. 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. The edge of the network maybe congested due to limited network resources. The SRR may be able to determine network congestion and quickly route service requests to other Service End point, which is not experiencing congestion. In

addition, application-layer control functions might utilize latency measurements to ensure that suitable service instances are being created during runtime of the scenario such as to ensure that service function endpoints are available 'nearby' (possibly) moving so as to keep a desired latency under a desired value.

Clearly, that is achieved by reducing the switching time from one SF endpoint to another. As the 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.

6.2. Details of SRR Function

Assuming such introduction of an HTTP-level transport notion, the SRR function can be decomposed further as shown in Fig 5.

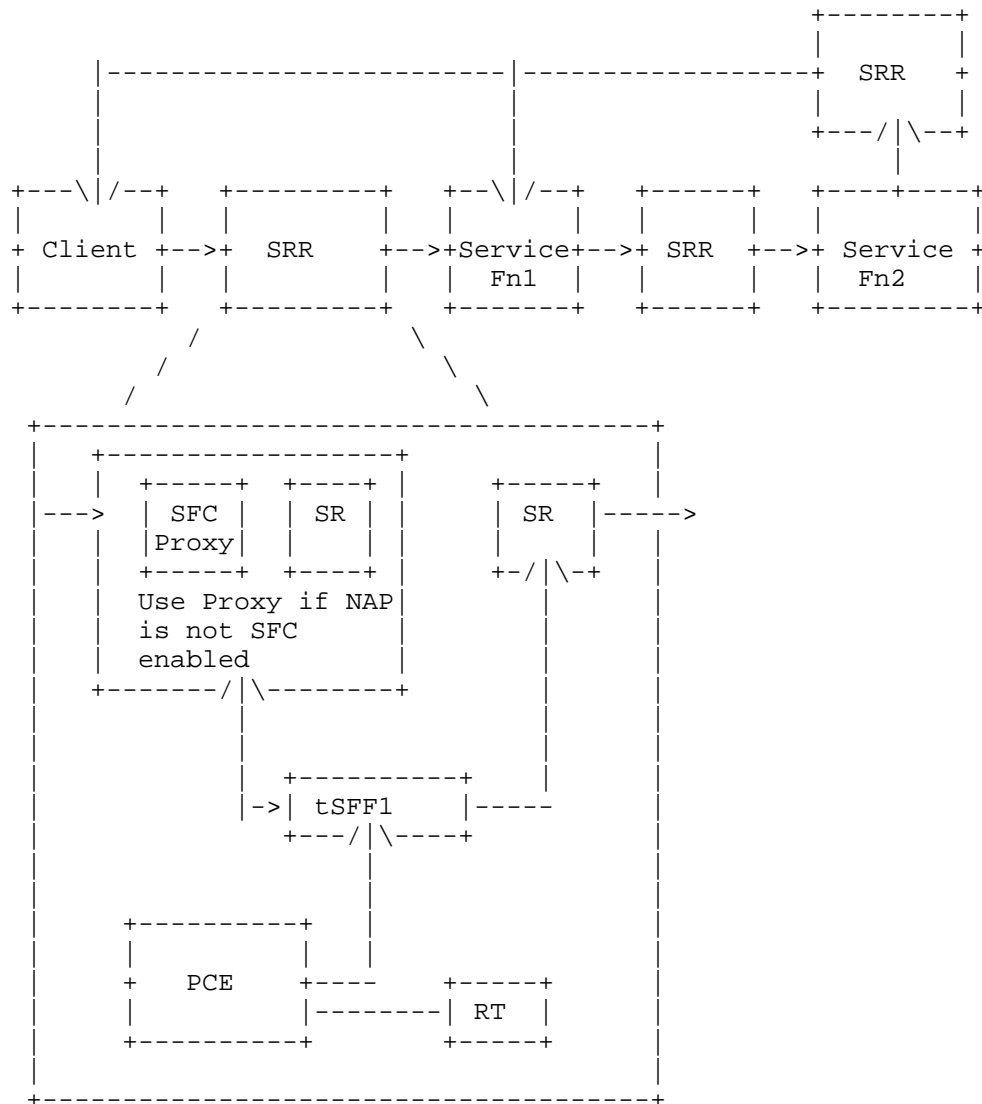


Figure 5: SRR decomposition

Another option for the two functions routing via the SRR could be entirely link-local, i.e., there's another simple tSFF2 between client and SRR as well as SF1 and SRR that is simply a link-local transport. The following figure describes this alternate option.

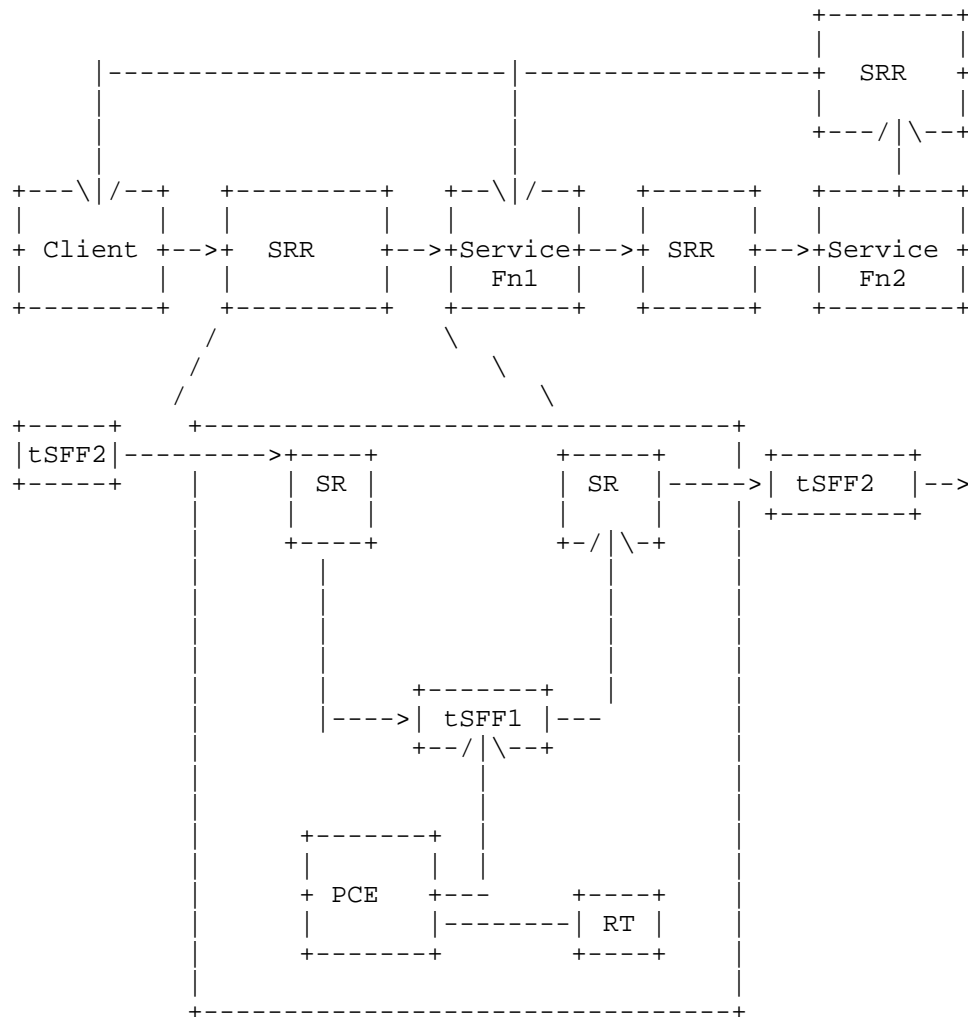


Figure 6: SRR decomposition using link-local client/function communication

The SRR function may be composed of the following functions:

- o Service Router(SR) at the ingress, terminates on the client side Layer 3 and above protocols, such as TCP

- o Service Router(SR) at the egress, terminates any transport protocol on the outgoing (server) side
- o PCE, Path Computation Element function is responsible for selecting the correct next SF, also possibly realizing path policy enforcement. The result of the selection is a path identifier which is delivered to the ingress SR upon initial path computation request (i.e., when sending a request to a specific URL on the SFP for the first time). The path identifier is utilized for any future request for a given URL-based SF. In case of another SF instance becoming available, indicated to the PCE through a registration procedure, the PCE will instruct all ingress SRs to invalidate path identifiers to the specific URL of the SF, resulting in an initial path computation request at the next SF request forwarding. Through this, the newly registered SF instance might be utilized if the policy-governed path computation will select said SF instance.
- o Reclassification Trigger Handler (RT) : Network measurement information, such as latency, packet loss or network congestion, etc. could be processed by the handler. This may trigger reconfiguration of the specific service function endpoint chain over which the SFC is being executed. The handler forwards the information about the chain reconfiguration to PCE.
- o Transport-derived SFF (tSFF1): the communication between ingress/egress SRs as well as SRs to PCE is realized via a transport-derived SFF. We outline here three possible tSFFs
 - * SDN-based: This option utilizes path-based forwarding through SDN-based wildcard matching fields, supported with OF1.2+[Reed2016]. It can be embedded into slicing approach of underlying transport infrastructure by leaving typical slicing fields available (e.g., VLAN tags). The forwarding utilizes the Ethernet frame format at Layer 2, representing the topological links of a specific forwarding path in the transport network as unique bits in a fixed size bit array. For the latter, the approach utilizes the IPv6 source and destination fields for storing the bit array information (in a simple version for this forwarding, this limits the topology to 256 links but extensions schemes are possible, which are left out of this document at this stage). As mentioned, the SDN forwarding decision action is a simple wildcard matching, supported with OF1.2+, with the wildcard representing the unique bit of a switch-specific output port. With that, the switch needs to consider as many forwarding rules as switch local output ports - see [Reed2016] for more information. Fig. xx illustrate this forwarding solution, including the ability

to create ad-hoc multicast relations by simply ORing individual bitarrays representing unicast paths.

- * Another approach is outlined in [I-D.ietf-bier-use-cases] where the SFF is suggested to be realized via a BIER overlay, in turn realized over a BIER-compliant underlay, such as MPLS. BIER utilizes a similar bit array approach for representing a forwarding path in the overlay network but unlike [Reed2016], the bit fields indicate the egress BIER-compliant router that the packet is supposed to reach.
- * As yet another alternative, the tSFF may utilize a flow aggregation approach, outlined in [Khalili2016], called edge switch classification (ESC). In this approach, a path from an ingress to egress SR is described as a so-called edge classification vector (ECV), which combines information on the aggregated flow (following [Khalili2016]) and the switch-local endpoint. The representation has similar bitarray characteristics as the previous two approaches
- o NOTE: with the ingress and egress SRs terminating SF Layer 3 connections and the utilization of bitarray-based tSFFs, the transmission of packets can effectively take place as an ad-hoc Layer multicast while the SFC itself is denoted as an n-times unicast SFC. As an example, consider the chaining of a set of n clients to a single video server. Each sub-SFC from an individual client to the video server will semantically result in a unicast response from the server back to the client (e.g., carrying the video chunk for a MPEG DASH-based video stream). When combining the sub-SFCs to the single SFC with n times unicast relations to the server, the SRR will deliver the responses from the server via one or more multicast responses to one or more clients. The size of the individual multicast groups will depend on the synchronicity of the client requests (and therefore on the synchronicity of the server responses). Note that the multicast relations here are ad-hoc created by ORing the bitarrays representing the specific clients to which the responses are meant to be sent. This is illustrated in the figure below. The HTTP multicast use case is being presented in the BIER use case draft [I-D.ietf-bier-use-cases] albeit without specific a SFC relation.

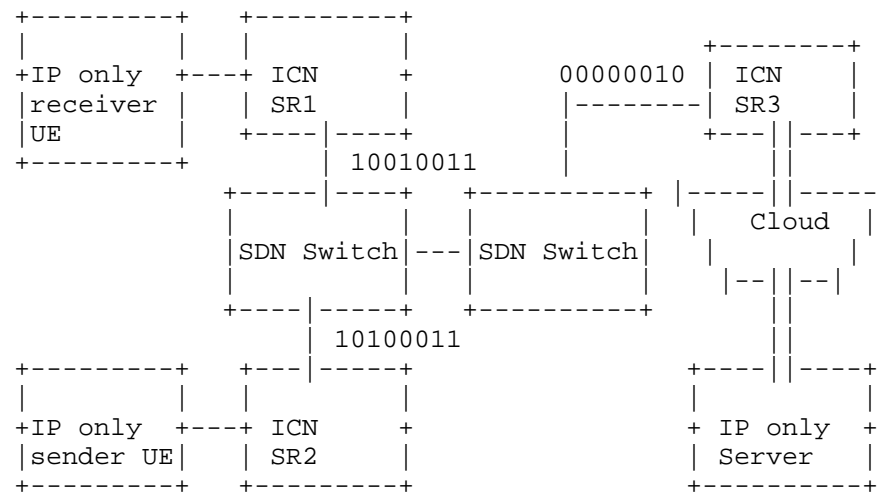


Figure 7: Illustration of Bitfield-based Forwarding using SDN

7. Protocol Consideration

For the operations outlined in the previous section, we foresee the following protocol changes are required:

- o SR-to-SR protocol for HTTP: HTTP based message exchange between client and server SRs
- o SR-PCE protocol: Used for path computation, obtaining routing information as well as provide path updates
- o Registration protocol: Used to register FQDN service endpoints

8. Next Steps

Feedback from the SFC WG on the validity of this solution and its scope within the SFC WG. If such alternative to the re-classification for service indirection is seen beneficial as well as fitting with the charter of the WG, the next steps would be to update the draft to outline potential protocol solutions required for the realization of such SRR SF.

9. IANA Considerations

This document requests no IANA actions.

10. Security Considerations

TBD.

11. 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>.

[H2020FLAME]

EU, "EU H2020 FLAME PROJECT", , March 2016, <<https://www.ict-flame.eu/>>.

[I-D.ietf-bier-use-cases]

Kumar, N., Asati, R., Chen, M., Xu, X., Dolganow, A., Przygienda, T., Gulko, A., Robinson, D., Arya, V., and C. Bestler, "BIER Use Cases", draft-ietf-bier-use-cases-06 (work in progress), January 2018.

[I-D.ietf-sfc-dc-use-cases]

Kumar, S., Tufail, M., Majee, S., Captari, C., and S. Homma, "Service Function Chaining Use Cases In Data Centers", draft-ietf-sfc-dc-use-cases-06 (work in progress), February 2017.

[Khalili2016]

Khalili, R., Poe, W., Despotovic, Z., and A. Hecker, "Reducing State of SDN Switches in Mobile Core Networks by Flow Rule Aggregation", ICCCN, August, 2016.

[Reed2016]

Reed, M., Al-Naday, M., Thomas, N., Trossen, D., and S. Spirou, "Reducing State of SDN Switches in Mobile Core Networks by Flow Rule Aggregation", ICC 2016, 2016.

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

[RFC8300] Quinn, P., Ed., Elzur, U., Ed., and C. Pignataro, Ed., "Network Service Header (NSH)", RFC 8300, DOI 10.17487/RFC8300, January 2018, <<https://www.rfc-editor.org/info/rfc8300>>.

[UKNIC] UK NIC, "5G Infrastructure Requirements in the UK", Final Report 3.0, December 2016,
<https://www.gov.uk/government/uploads/system/uploads/attachment_data/file/577940/5G_Infrastructure_requirements_for_the_UK_-_LS_Telcom_report_for_the_NIC.pdf>.

Authors' Addresses

Debashish Purkayastha
InterDigital Communications, LLC
Conshohocken
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/>

Zoran Despotovic
Huawei

Email: Zoran.Despotovic@huawei.com
URI: <http://www.huawei.com/>

Ramin Khalili
Huawei

Email: Ramin.khalili@huawei.com
URI: <http://www.huawei.com/>