Service Function discovery in fog environments
draft-bernardos-sfc-discovery-04

Abstract

   Service function chaining (SFC) allows the instantiation of an
   ordered set of service functions and subsequent "steering" of traffic
   through them. Service functions provide an specific treatment of
   received packets, therefore they need to be known so they can be used
   in a given service composition via SFC. This document discusses the
   need for service function discovery mechanisms and propose some
   solutions for sfc-aware nodes to discover available service functions
   in fog environments.

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 12, 2020.

Table of Contents

1. Introduction

Virtualization of functions provides operators with tools to deploy
new services much faster, as compared to the traditional use of
monolithic and tightly integrated dedicated machinery. As a natural
next step, mobile network operators need to re-think how to evolve
their existing network infrastructures and how to deploy new ones to
address the challenges posed by the increasing customers' demands, as
well as by the huge competition among operators. All these changes
are triggering the need for a modification in the way operators and
infrastructure providers operate their networks, as they need to
significantly reduce the costs incurred in deploying a new service
and operating it. Some of the mechanisms that are being considered
and already adopted by operators include: sharing of network
infrastructure to reduce costs, virtualization of core servers
running in data centers as a way of supporting their load-aware
elastic dimensioning, and dynamic energy policies to reduce the
monthly electricity bill. However, this has proved to be tough to
put in practice, and not enough. Indeed, it is not easy to deploy
new mechanisms in a running operational network due to the high
dependency on proprietary (and sometime obscure) protocols and
interfaces, which are complex to manage and often require configuring
multiple devices in a decentralized way.

Service Functions are widely deployed and essential in many networks.
These Service Functions provide a range of features such as security,

WAN acceleration, and server load balancing. Service Functions may
be instantiated at different points in the network infrastructure
such as data center, the WAN, the RAN, and even on mobile nodes.

Service functions (SFs), also referred to as VNFs, or just functions,
are hosted on compute, storage and networking resources. The hosting
environment of a function is called Service Function Provider or
NFVI-PoP (using ETSI NFV terminology).

With the arrival of virtualization, the deployment model for service
function is evolving to one where the traffic is steered through the
functions wherever they are deployed (functions do not need to be
deployed in the traffic path anymore). For a given service, the
abstracted view of the required service functions and the order in
which they are to be applied is called a Service Function Chain
(SFC). An SFC is instantiated through selection of specific service
function instances on specific network nodes to form a service graph:
this is called a Service Function Path (SFP). The service functions
may be applied at any layer within the network protocol stack
(network layer, transport layer, application layer, etc.).

A mobile terminal can benefit from using service function chaining at
the edge/fog to enhance existing applications or to enable new ones.
In order to do so, discovery of available service functions is
required. This document focuses on this aspect.

2. Terminology

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

While [RFC2119] describes interpretations of these key words in terms
of protocol specifications and implementations, they are used in this
document to describe requirements for the SFC mechanisms to
efficiently enable fog RAN.

The following terms used in this document are defined by the IETF in
[RFC7665] and [I-D.ietf-bess-nsh-bgp-control-plane]:

   Service Function (SF): a function that is responsible for specific
   treatment of received packets (e.g., firewall, load balancer).

   Service Function Chain (SFC): for a given service, the abstracted
   view of the required service functions and the order in which they
   are to be applied. This is somehow equivalent to the Network
   Function Forwarding Graph (NF-FG) at ETSI.

Service Function Forwarder (SFF): A service function forwarder is
responsible for forwarding traffic to one or more connected
service functions according to information carried in the SFC
encapsulation, as well as handling traffic coming back from the
SF.

SFI: SF instance.

Service Function Path (SFP): the selection of specific service
function instances on specific network nodes to form a service
graph through which an SFC is instantiated.

A Service Function Type (SFT) that is the category of Service
Function that is provided (such as "firewall").

3. Problem statement

[RFC7665] describes an architecture for the specification, creation,
and ongoing maintenance of Service Function Chains (SFCs) in a
network. It includes architectural concepts, principles, and
components used in the construction of composite services through
deployment of SFCs. In this architecture, a key element is the
service function (SF), which is a function that is responsible for
specific treatment of received packets (e.g., a firewall).

So far, how the SFs are discovered and composed has been out of the
scope of discussions in IETF. There is however a need to define
mechanisms that allow SF discovery in fog environments
[I-D.bernardos-sfc-fog-ran]. Note that the mechanisms described in
this document address fog environments. There are other mechanisms
described, like [I-D.ietf-bess-nsh-bgp-control-plane], that cover
generic SF discovery in more traditional environments. Some of the
solutions described in the present document might be of applicable to
other scenarios as well.

3.1. Discovery of SF in a multi-provider fog/edge environment

The need to provide networking, computing, and storage capabilities
closer to the users has recently emerged, due to the demands from 5G
applications of very low latency, leading to what is known today as
the concept of intelligent edge. ETSI has been the first to address
this need recently by developing the framework of mobile edge
computing (MEC). Such an intelligent edge could not be envisaged
without virtualization. Beyond applications, it raises a clear
opportunity for networking functions to execute at the edge
benefiting from inherent low latencies. Being in close proximity to
the access, the edge becomes an attractive place for hosting
different functions, saving bandwidth in their respective domains and

offering local breakout options where required. Whilst it is
appreciated the particular challenge for the intelligent edge concept
in dealing with mobile users, the edge virtualization substrate has
been largely assumed to be fixed or stationary. Although little
developed, the intelligent edge concept is being extended further to
scenarios where for example the edge computing substrate is on the
move, e.g., on-board a car or a train, or that it is distributed
further down the edge, even integrating resources from different
stakeholders, into what is known as the fog.

Service composition is a powerful tool which can provide significant
benefits when applied in a softwarized network environment. While it
is being explored in the core part of networks to compose services
using DPIs (Deep Packet Inspections), firewalls, parental control,
video accelerators, etc., its applicability to the RAN (Radio Access
Network), and in particular to the edge and the fog, has not been
explored yet.

Running functions (standalone functions or service function chains)
at the edge of the network has clear advantages. For example, it
enables offloading functions from the end-user terminal so that it
can become more efficient in terms of cost and energy consumption.

A mobile terminal can benefit from using service function chaining at
the edge/fog to enhance existing applications or to enable new ones.
Some examples of such applications are: privacy enhancement by local
anchoring, opportunistic local breakout, assisted encryption, video
transcoding, personal firewalling, etc. The mobile terminal might
look for function hosting opportunities at the edge for various
reasons such as:

o to increase battery life in critical situations by offloading
   energy demanding operations (e.g., video transcoding, augmented
   reality) to the edge/cloud;

o to reduce communications latency (e.g., by using local breakout at
   the edge for selected applications demanding low latency);

o to enable new functions (e.g., privacy improvements, personal
   firewalling) which demand additional intelligence/resources at the
   network;

o to benefit from context information available at the edge (e.g.,
   enrich networking decisions by executing functions at the edge
   using RAN information);

Several key challenges need to be addressed to enable controlled
service function chaining for a mobile terminal, and one of them is

the discovery of the functions available for use at the Fog/Edge/
Cloud.

4. Network-based SF discovery

In this section we describe several mechanisms for a mobile SFC-aware
node to discover what SFs are available in the network. Different
alternatives (protocol containers) are considered to enable the
mobile node to obtain the following information per SF available:

o Service Function Type, identifying the category of SF provided.

o SFC-aware: Yes/No. Indicates if the SF is SFC-aware.

o Route Distinguisher (RD): IP address indicating the location of
   the SF(I).

o Pricing/costs details.

o Migration capabilities of the SF: whether a given function can be
   moved to another provider (potentially including information about
   compatible providers topologically close).

o Mobility of the device hosting the SF, with e.g. the following
   sub-options:

      Level: no, low, high; or a corresponding scale (e.g., 1 to 10).

      Current geographical area (e.g., GPS coordinates, post code).

      Target moving area (e.g., GPS coordinates, post code).

o Power source of the device hosting the SF, with e.g. the following
   sub-options:

      Battery: Yes/No. If Yes, the following sub-options could be
      defined:

      Capacity of the battery (e.g., mmWh).

      Charge status (e.g., %).

      Lifetime (e.g., minutes).

Figure 1 shows the generic mechanism for SF discovery, with network
support. In this scenario, SFs (which might belong to different
administrative domains) are previously registered at the network,
which can then reply to requests sent from mobile nodes that have

just attached to the network. A request might optionally include the
SFs of interest for the terminal, instead of a request for all known
SFs.

The network might also send periodic advertisements in addition to
responses to solicited requests. These responses/advertisements
include the information about known SFs (or only about the ones
queried by the terminal), which can then be used by the terminal to
decide whether to use (some of) them in a certain SFC. How the
mobile terminal then configures this SFC is not covered in this
document.

```
                                                 _____
                                               _(           )_
                                             _(   SF1    SF2   )_
     ------------                 -----------  _(        SF3         )_
    | terminal |                 | network |-(_   SF4          SF5    _)
     ------------                 -----------   (_    SF6    SF7     _)
        |                             |           (_      SF8      _)
     XXX (1. attachment)             |             (_____)
        |                            |
      +---2. Request (optional)------>|
        |                            |
        |<--------3. Response/Advert.---|
        |             (SF1,SF2...,SF8)   |
        |                            |
```
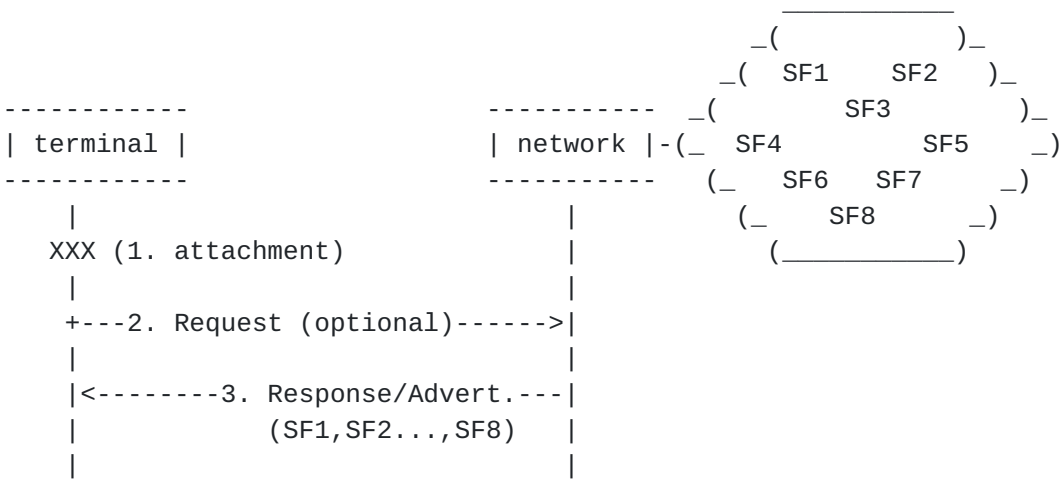
                    Figure 1: SF (network) discovery

In addition to the discovery of SFs at the infrastructure, mobile
terminals can also host SF(I)s, and therefore they also need to be
discovered. A similar approach can be followed, as showin in
Figure 2.

```
                    ------------
                    |   SF3    |
      ------------  | terminal |   ------------
      | SF1  SF2 |  ------------   | SF4  SF5 |
      | terminal |       |        | terminal |
      ------------       |         ------------
           |             |              |
        +--1. Request->+-------------->|
        |    (SF1,SF2) |               |
        |<---------------2. Response--+
        |            (SF4,SF5)         |
        |<-2. Response-+               |
        |       (SF3)  |               |
        |              |               |
```
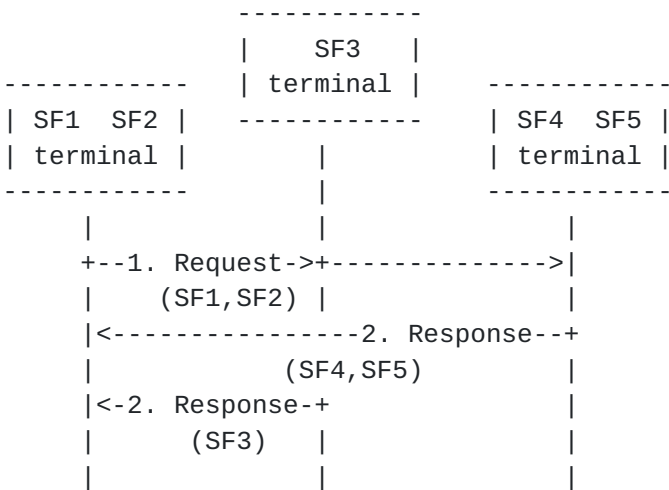
                  Figure 2: SF (mobiles) discovery

   SFs might belong to different administrative domains. This might
   require the use of additional security and authentication mechanisms.
   Policies can be used (both in single and multi-domain scenarios) to
   adapt/limit the type and number of SFs that are advertised, depending
   on the relationship of the requester and the advertiser.

   Next sections describe different protocol alternatives for this SF
   discovery in fog environments.

4.1. ICMPv6-based SF discovery

   TBD.

4.2. DHCPv6-based SF discovery

   TBD.

5. IANA Considerations

   N/A.

6. Security Considerations

   TBD.

7. Acknowledgments

   The work in this draft will be further developed and explored under
   the framework of the H2020 5G-DIVE project (Grant 859881).

8. References

8.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate
           Requirement Levels", BCP 14, RFC 2119,
           DOI 10.17487/RFC2119, March 1997,
           <https://www.rfc-editor.org/info/rfc2119>.

8.2. Informative References

[I-D.bernardos-sfc-fog-ran]
           Bernardos, C., Rahman, A., and A. Mourad, "Service
           Function Chaining Use Cases in Fog RAN", draft-bernardos-
           sfc-fog-ran-06 (work in progress), September 2019.

[I-D.ietf-bess-nsh-bgp-control-plane]
           Farrel, A., Drake, J., Rosen, E., Uttaro, J., and L.
           Jalil, "BGP Control Plane for NSH SFC", draft-ietf-bess-
           nsh-bgp-control-plane-13 (work in progress), December
           2019.

[RFC7665] Halpern, J., Ed. and C. Pignataro, Ed., "Service Function
           Chaining (SFC) Architecture", RFC 7665,
           DOI 10.17487/RFC7665, October 2015,
           <https://www.rfc-editor.org/info/rfc7665>.

Authors' Addresses

   Carlos J. Bernardos
   Universidad Carlos III de Madrid
   Av. Universidad, 30
   Leganes, Madrid 28911
   Spain

   Phone: +34 91624 6236
   Email: cjbc@it.uc3m.es
   URI: http://www.it.uc3m.es/cjbc/

   Alain Mourad
   InterDigital Europe

   Email: Alain.Mourad@InterDigital.com
   URI: http://www.InterDigital.com/