SFC WG CJ. Bernardos
Internet-Draft UC3M
Intended status: Experimental A. Mourad
Expires: 22 September 2022 InterDigital
21 March 2022

SFC function mobility with Mobile IPv6
draft-bernardos-dmm-sfc-mobility-04

Abstract

Service function chaining (SFC) allows the instantiation of an ordered set of service functions and subsequent "steering" of traffic through them. In order to set up and maintain SFC instances, a control plane is required, which typically is centralized. In certain environments, such as fog computing ones, such centralized control might not be feasible, calling for distributed SFC control solutions. This document specifies Mobile IPv6 extensions to enable function migration in SFC.

Status of This Memo

Copyright Notice

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).

Services are typically formed as a composition of SFs (VNFs), with each SF providing a specific function of the whole service. Services also referred to as Network Services (NS), according to ETSI 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.).

The concept of fog computing has emerged driven by the Internet of Things (IoT) due to the need of handling the data generated from the end-user devices. The term fog is referred to any networked computational resource in the continuum between things and cloud. A fog node may therefore be an infrastructure network node such as an eNodeB or gNodeB, an edge server, a customer premises equipment (CPE), or even a user equipment (UE) terminal node such as a laptop, a smartphone, or a computing unit on-board a vehicle, robot or drone.

In fog computing, the functions composing an SFC are hosted on resources that are inherently heterogeneous, volatile and mobile [I-D.bernardos-sfc-fog-ran]. This means that resources might appear and disappear, and the connectivity characteristics between these resources may also change dynamically. These scenarios call for distributed SFC control solutions, where there are SFC pseudo controllers, enabling autonomous SFC self-orchestration capabilities. The concept of SFC pseudo controller (P-CTRL) is described in [I-D.bernardos-sfc-distributed-control], as well different procedures for their discovery and initialization.

This document specifies Mobile IPv6 extensions to enable function migration in SFC.

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].

The following terms used in this document are defined by the IETF in [RFC7665]:

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.

The following terms are used in this document:

SFC Pseudo Controller (P-CTRL): logical entity [I-D.bernardos-sfc-distributed-control], complementing the SFC controller/orchestrator found in current architectures and deployments. It is service specific, meaning that it is defined and meaningful in the context of a given network service. Compared to existing SFC controllers/orchestrators, which manage multiple SFCs instantiated over a common infrastructure, pseudo controllers are constrained to service specific lifecycle management.

SFC Central Controller (C-CTRL): central control plane logical entity in charge of configuring and managing the SFC components [RFC7665].

3. Function mobility signaling extending Mobile IPv6

This section describes Mobile IPv6 (MIPv6) extensions to perform function migration/mobility. This is an example of NS lifecycle management operation: the update of the location of a given function. We refer to this as function mobility, though it might involve or not the actual migration of the function.

```
+----------+  +-----+  +----------+ +----------+ +-----------+   +-------+
| node A   |  | C   |  | node B   | | node D   | |    3GPP   |   | SFC   |
|P-CTRL F1|  | F3  |  |P-CTRL F2| |P-CTRL F3| |ctrl plane|   |C-CTRL|
+--+----+-+  +-----+  +--+----+-+ +--+----+-+ +-----------+   +-------+
   |    |       |       |    |      |    |         |            |
   |  F1@A<->F2@B<->F3@D SFC network service         |            |
   |    |<-·-·-·-·-·-·-·-·->|<-·-·-·-·->|         |            |
   |    |       |       |    |      |    |         |            |
   |    |       |    Node B moves out of           |            |
   |    |       |   the coverage of node D         |            |
   |    |       |       |    |      |    |         |            |
   | 0. Service specific OAM monitoring  |         |            |
   |<-·>|<-·-·->|<-·-·-·-·-·>|      |    |         |            |
   |<-·-·-·-·-·-·-·-·-·-·-·-·-·-·-·-·-·-·-·-·-·-·->|            |
```

```
      |    |       |       |    |      |    |         |            |
    P-CTRL@A detects D disconnection   |    |         |            |
   and decides to place F3 at node C   |    |         |            |
      |    |       |       |    |      |    |         |            |
      | 1a. SPU[NS_ID,(F3,C)]   |      |    |         |            |
      |-.-.-.-.-.-.-.-.-.-.-.-.>|      |    |         |            |
      |          1b. SPA[NS_ID] |      |    |         |            |
      |<-.-.-.-.-.-.-.-.-.-.-.-.|      |    |         |            |
      | 1c. SPU[NS_ID,(F3,C),(F2,B),(F1,A)] |         |            |
      |-.-.-.-.-.>|    |    |      |    |         |            |
      | 1d. SPA[NS_ID]     |    |      |    |         |            |
      |<-.-.-.-.-.|    |    |      |    |         |            |
      |    |       |       |    |      |    |         |            |
      | 2. Updated F1@A<->F2@B<->F3@C SFC network service          |
      |    |<-.-.-.-.-.-.-.-.-.>|      |    |         |            |
      |    |       |<-.-.-.-.->|      |    |         |            |
      |    |       |       |    |      |    |         |            |
      |  3a. SPU[NS_ID,(F3,C),(F2,B),(F1,A)]          |            |
      |-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.->|
      |    |       |       |    |      |    |      3b. SPA[NS_ID] |
      |<-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-|
      |  3c. SPU[NS_ID,(F3,C)]  |      |    |         |            |
      |-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.>|         |            |
      |    |       |       | 3d. SPA[NS_ID] |         |            |
      |<-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-|
      |    |       |       |    |      |    |         |            |
```


Figure 1: SFC mobility signaling

We next describe the signaling extensions with an example. For the sake of this example we assume that the function which location is updated is already available at the new target node (if not, it has to be previously migrated using any of the solutions available in the state-of-the-art). The different steps are described next:

* (The network service F1--F2--F3 is already instantiated and running. The only SFC P-CTRL active at this point is running at node A, and there is a candidate one at node B.)

* UE node B is moving out of the coverage of gNB node D.

1. This movement is detected by the active (designated) pseudo

controller running at node A, thanks to local (service specific OAM) monitoring.

2. The active pseudo controller sends mobility signaling to all affected nodes, in this case node B (it has to update the network service path due to the F3 location update) and node C (as it starts being part of the SFC, hosting F3). The signaling messages are new mobility messages: Service Path Update (SPU) and Service Path Acknowledgement (SPA), which contain: (i) the identifier of the network service (NS_ID), and (ii) the updated elements of the network service path: (ID, updated location). The SPA acknowledges that the procedure has been performed correctly.

3. The network service F1--F2--F3 is updated so it now runs at A, B and C.

4. Whenever connectivity with nodes D and the centralized SFC controller is back, the pseudo controller also informs about the updated SFC path, sending SPU messages, which are acknowledged with SPA messages.

Note that this is an example of NS lifecycle management (function mobility) by a SFC pseudo controller, but that other operations are also possible, such as (non-limiting examples): scaling up/down, scaling in/out, termination, etc.

4. Mobile IPv6 extensions for SFC function mobility

4.1. Service Path Update

The Service Path Update (SPU) message is used by a CTRL to notify nodes in an SFC (e.g., SFF) of an update of the service path.

The Service Path Update uses the MH Type value TBD. When this value is indicated in the MH Type field, the format of the Message Data field in the Mobility Header is as follows:

```
    0                   1                   2                   3
  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
                                 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                                 |          Sequence #           |
```

```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|A|          Reserved           |           Lifetime            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
.                                                               .
.                        Mobility Options                       .
.                                                               .
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Sequence #

A 16-bit unsigned integer used by the receiving node to sequence Binding Updates and by the sending node to match a returned Service Path Acknowledgement with this Service Path Update.

Acknowledge (A)

The Acknowledge (A) bit is set by the sending mobile node to request a Service Path Acknowledgement be returned upon receipt of the Service Path Update.

Reserved

This field is unused for now. The value MUST be initialized to 0 by the sender and MUST be ignored by the receiver.

Lifetime

16-bit unsigned integer. The number of time units remaining before the service path MUST be considered expired. A value of zero indicates that the Service Path MUST be deleted. A value of 0xFFFF indicates an infinite lifetime for the Service Path. One time unit is 4 seconds.

Mobility Options

Variable-length field of such length that the complete Mobility

Header is an integer multiple of 8 octets long. This field contains zero or more TLV-encoded mobility options. The receiver MUST ignore and skip any options that it does not understand.

The following options are valid in a Service Path Update:

- Network Service ID.

- SFC node.

4.2. Service Path Acknowledgement

The Service Path Acknowledgement (SPA) message is used by a CTRL to acknowledge a received SPU.

The Service Path Acknowledge uses the MH Type value TBD. When this value is indicated in the MH Type field, the format of the Message Data field in the Mobility Header is as follows:

```
    0                   1                   2                   3
  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
                                 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                                 |          Sequence #           |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |            Reserved           |           Lifetime            |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                                                               |
 .                                                               .
 .                        Mobility Options                       .
 .                                                               .
 |                                                               |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Sequence #

A 16-bit unsigned integer used to match the returned Service Path Acknowledgement with the Service Path Update.

Reserved

This field is unused for now. The value MUST be initialized to 0 by the sender and MUST be ignored by the receiver.

Lifetime

16-bit unsigned integer. The number of time units remaining before the service path MUST be considered expired. A value of zero indicates that the Service Path MUST be deleted. A value of 0xFFFF indicates an infinite lifetime for the Service Path. One time unit is 4 seconds.

Mobility Options

Variable-length field of such length that the complete Mobility Header is an integer multiple of 8 octets long. This field contains zero or more TLV-encoded mobility options. The receiver MUST ignore and skip any options that it does not understand.

The following options are valid in a Service Path Acknowledgement:

- Network Service ID.

4.3. New Mobility options

4.3.1. Network Service ID

The Network Service ID option has the following format:

```
  0                   1                   2                   3
 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
                                +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                                |   Type = TBA  | Option Length |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Service Path Identifier (SPI)        | Service Index |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+                                                               +
|                                                               |
+                      Network Service ID                       +
|                                                               |
+                                                               +
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Option Type

TBA by IANA.

Option Length

8-bit unsigned integer. Length of the option, in octets, excluding the Option Type and Option Length fields.

Service Path Identifier (SPI)

Uniquely identifies a Service Function Path (SFP). Participating nodes MUST use this identifier for SFP selection. The initial Classifier MUST set the appropriate SPI for a given classification result.

Service Index (SI)

Provides location within the SFP.

Network Service ID

Variable length field that identifies the network service.

4.3.2. SFC node

The SFC node option has the following format:

```
 0                   1                   2                   3
 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
                                +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                                |   Type = TBA  | Option Length |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      Function ID Length       |        Node ID Length         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+                                                               +
|                                                               |
+                          Function ID                          +
|                                                               |
+                                                               +
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+                                                               +
|                                                               |
+                            Node ID                            +
|                                                               |
```

```
   +                                                               +
   |                                                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Option Type

TBA by IANA.

Option Length

8-bit unsigned integer. Length of the option, in octets, excluding the Option Type and Option Length fields.

Function ID Length

8-bit unsigned integer. Length of the Function ID field, in octets.

Node ID Length

8-bit unsigned integer. Length of the Node ID field, in octets.

Function ID

Variable length field that identifies the function.

Node ID

Variable length field that identifies the node.

There might be multiple SFC node options in a Service Function Update message, following the options the same order of the SFC/NS.

5. IANA Considerations

TBD.

6. Security Considerations

TBD.

7. Acknowledgments

The work in this draft has been partially supported by the H2020 5Growth (Grant 856709) and 5G-DIVE projects (Grant 859881).

8. References

8.1. Normative References

[I-D.bernardos-sfc-distributed-control]
Bernardos, C. J. and A. Mourad, "Distributed SFC control for fog environments", Work in Progress, Internet-Draft, draft-bernardos-sfc-distributed-control-05, 27 January 2022, <https://www.ietf.org/archive/id/draft-bernardos-sfc-distributed-control-05.txt>.

[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. J. and A. Mourad, "Service Function Chaining Use Cases in Fog RAN", Work in Progress, Internet-Draft, draft-bernardos-sfc-fog-ran-10, 22 October 2021, <https://www.ietf.org/archive/id/draft-bernardos-sfc-fog-ran-10.txt>.

[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
28911 Leganes, Madrid

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/