INTAREA WG CJ. Bernardos
Internet-Draft UC3M
Intended status: Experimental A. Mourad
Expires: February 27, 2020 InterDigital
August 26, 2019

IPv6-based discovery and association of Virtualization Infrastructure Manager (VIM) and Network Function Virtualization Orchestrator (NFVO)
draft-bernardos-intarea-vim-discovery-02

Abstract

Virtualized resources do not need to be limited to those available in traditional data centers, where the infrastructure is stable, static, typically homogeneous and managed by a single admin entity. Computational capabilities are becoming more and more ubiquitous, with terminal devices getting extremely powerful, as well as other types of devices that are close to the end users at the edge (e.g., vehicular onboard devices for infotainment, micro data centers deployed at the edge, etc.). It is envisioned that these devices would be able to offer storage, computing and networking resources to nearby network infrastructure, devices and things (the fog paradigm). These resources can be used to host functions, for example to offload/complement other resources available at traditional data centers, but also to reduce the end-to-end latency or to provide access to specialized information (e.g., context available at the edge) or hardware.

This document describes mechanisms allowing dynamic discovery of virtualization resources and orchestrators in IPv6-based networks. In the current version, mechanisms based on piggybacking options on IPv6 neighbor discovery are explored. New IPv6 neighbor discovery options are defined. Additional mechanisms will be explored in future releases of this document.

Copyright Notice

Table of Contents

1. Introduction

The telecommunications sector is experiencing a major revolution that will shape the way networks and services are designed and deployed for the next decade. We are witnessing an explosion in the number of applications and services demanded by users, which are now really capable of accessing them on the move. In order to cope with such a demand, some network operators are looking at the cloud computing paradigm, which enables a potential reduction of the overall costs by outsourcing communication services from specific hardware in the operator's core to server farms scattered in data centers. These services have different characteristics if compared with conventional IT services that have to be taken into account in this cloudification process. Also the transport network is affected in that it is evolving to a more sophisticated form of IP architecture with trends like separation of control and data plane traffic, and more fine-grained forwarding of packets (beyond looking at the destination IP address) in the network to fulfill new business and service goals.

Virtualization of functions also 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.

Network function virtualization (NFV) [etsi_nfv_whitepaper] and software defined networking (SDN) [onf_sdn_architecture] are changing the way the telecommunications sector will deploy, extend and operate their networks. The ETSI NFV Industry Specification Group (ISG) is developing the baseline NFV architecture, under some assumptions to make this development easier. One of these assumptions is that the resources used to run the virtualized functions are well known in

advance by the management and orchestration entities, as well as stable. This document goes beyond this assumption [RFC8568], by describing mechanisms allowing dynamic discovery of virtualization resources and orchestrators in IPv6-based networks.

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 ETSI NFV ISG, the ONF and the IETF:

NFV Infrastructure (NFVI): totality of all hardware and software components which build up the environment in which VNFs are deployed

NFV Management and Orchestration (NFV-MANO): functions collectively provided by NFVO, VNFM, and VIM.

NFV Orchestrator (NFVO): functional block that manages the Network Service (NS) lifecycle and coordinates the management of NS lifecycle, VNF lifecycle (supported by the VNFM) and NFVI resources (supported by the VIM) to ensure an optimized allocation of the necessary resources and connectivity.

Virtualized Infrastructure Manager (VIM): functional block that is responsible for controlling and managing the NFVI compute, storage and network resources, usually within one operator's Infrastructure Domain.

Virtualized Network Function (VNF): implementation of a Network Function that can be deployed on a Network Function Virtualisation Infrastructure (NFVI).

Virtualized Network Function Manager (VNFM): functional block that is responsible for the lifecycle management of VNF.

3. Network Function Virtualization

The ETSI ISG NFV is a working group which, since 2012, aims to evolve quasi-standard IT virtualization technology to consolidate many network equipment types into industry standard high volume servers, switches, and storage. It enables implementing network functions in software that can run on a range of industry standard server hardware and can be moved to, or loaded in, various locations in the network as required, without the need to install new equipment. The ETSI NFV is one of the predominant NFV reference framework and architectural footprints [nfv_sota_research_challenges]. The ETSI NFV framework architecture framework is composed of three domains (Figure 1):

- Virtualized Network Function, running over the NFVI.

- NFV Infrastructure (NFVI), including the diversity of physical resources and how these can be virtualized. NFVI supports the execution of the VNFs.

- NFV Management and Orchestration, which covers the orchestration and life-cycle management of physical and/or software resources that support the infrastructure virtualization, and the life-cycle management of VNFs. NFV Management and Orchestration focuses on all virtualization specific management tasks necessary in the NFV framework.

```
+-------------------------------------------+  +---------------+
|   Virtualized Network Functions (VNFs)    |  |               |
|  -------   -------   -------   -------    |  |               |
|  |     |   |     |   |     |   |     |    |  |               |
|  | VNF |   | VNF |   | VNF |   | VNF |    |  |               |
|  |     |   |     |   |     |   |     |    |  |               |
|  -------   -------   -------   -------    |  |               |
+-------------------------------------------+  |               |
                                               |               |
+-------------------------------------------+  |               |
|         NFV Infrastructure (NFVI)         |  |      NFV      |
| -----------    -----------    ----------- |  |  Management   |
| | Virtual |    | Virtual |    | Virtual | |  |      and      |
| | Compute |    | Storage |    | Network | |  | Orchestration |
| -----------    -----------    ----------- |  |               |
| +---------------------------------------+ |  |               |
| |         Virtualization Layer          | |  |               |
| +---------------------------------------+ |  |               |
| +---------------------------------------+ |  |               |
| | -----------  -----------  ----------- | |  |               |
| | | Compute |  | Storage |  | Network | | |  |               |
| | -----------  -----------  ----------- | |  |               |
| |          Hardware resources           | |  |               |
| +---------------------------------------+ |  |               |
+-------------------------------------------+  +---------------+
```

Figure 1: ETSI NFV framework

The NFV architectural framework identifies functional blocks and the main reference points between such blocks. Some of these are already present in current deployments, whilst others might be necessary additions in order to support the virtualization process and consequent operation. The functional blocks are (Figure 2):

- Virtualized Network Function (VNF).

- Element Management (EM).

- NFV Infrastructure, including: Hardware and virtualized resources, and Virtualization Layer.

- Virtualized Infrastructure Manager(s) (VIM).

- NFV Orchestrator.

- VNF Manager(s).

- Service, VNF and Infrastructure Description.

o Operations and Business Support Systems (OSS/BSS).

```
                                               +--------------------+
   +-------------------------------------------+  | ----------------   |
   |                 OSS/BSS                   |  | | NFV          |   |
   +-------------------------------------------+  | | Orchestrator +-- |
                                                  | ---+------------ | |
   +-------------------------------------------+  |    |             | |
   |  ---------     ---------     ---------    |  |    |             | |
   |  | EM 1  |     | EM 2  |     | EM 3  |    |  |    |             | |
   |  ----+----     ----+----     ----+----    |  | ---+----------   | |
   |      |             |             |        |--|-|    VNF     |   | |
   |  ----+----     ----+----     ----+----    |  | | manager(s) |   | |
   |  | VNF 1 |     | VNF 2 |     | VNF 3 |    |  | ---+----------   | |
   |  ----+----     ----+----     ----+----    |  |    |             | |
   +------|-------------|-------------|--------+  |    |             | |
          |             |             |           |    |             | |
   +------+-------------+-------------+--------+  |    |             | |
   |         NFV Infrastructure (NFVI)         |  |    |             | |
   | -----------    -----------    ----------- |  |    |             | |
   | | Virtual |    | Virtual |    | Virtual | |  |    |             | |
   | | Compute |    | Storage |    | Network | |  |    |             | |
   | -----------    -----------    ----------- |  | ---+------       | |
   | +-------------------------------------+ |  | |        |       | |
   | |         Virtualization Layer          | |--|-| VIM(s) +-------- |
   | +-------------------------------------+ |  | |        |         |
   | +-------------------------------------+ |  | ----------         |
   | | -----------  -----------  ----------- | |  |                    |
   | | | Compute |  | Storage |  | Network | | |  |                    |
   | | | hardware|  | hardware|  | hardware| | |  |                    |
   | | -----------  -----------  ----------- | |  |                    |
   | |          Hardware resources           | |  |  NFV Management    |
   | +-------------------------------------+ |  | and Orchestration  |
   +-------------------------------------------+  +--------------------+
```

Figure 2: ETSI NFV reference architecture

4. Fog Virtualization Overview

Virtualization is invading all domains of the E2E 5G network, including the access, as a mean to achieve the necessary flexibility in support of the E2E slicing concept. The ETSI NFV framework is the cornerstone for making virtualization such a promising technology that can be matured in time for 5G. Typically, virtualization has been mostly envisaged in the core network, where sophisticated data centers and clouds provided the right substrate. And mostly, the framework focused on virtualizing network functions, so called VNFs (virtualized network functions), which were somewhat limited to

functions that are delay tolerant, typically from the core and aggregation transport.

As the community has recently been developing the 5G applications and their technical requirements, it has become clear that certain applications would require very low latency which is extremely challenging and stressing for the network to deliver through a pure centralized architecture. The need to provide networking, computing, and storage capabilities closer to the users has therefore emerged, 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.

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. The challenges and opportunities for such extensions of the intelligent edge remain an exciting area of future research.

Figure 3 shows a diagram representing the fog virtualization concept. The fog is composed by virtual resources on top of heterogeneous resources available at the edge and even further in the RAN and end-user devices. These resources are therefore owned by different stakeholders who collaboratively form a single hosting environment for the VNFs to run. As an example, virtual resources provided to the fog might be running on eNBs, APs, at micro data centers deployed in shopping malls, cars, trains, etc. The fog is connected to data centers deeper into the network architecture (at the edge ir the core). On the top part of the figure, an example of user and control plane VNFs is shown. User plane VNFs are represented as "fx", and control ones as "ctrlx". Depending on the functionality implemented by these VNFs and the service requirements, these VNFs would be mapped (i.e., instantiated) differently to the physical resouces (as described in [I-D.aranda-sfc-dp-mobile]).

```
            --------                        ---------   ---------
   control  | ctrl1 |.......................| ctrl2 |...| ctrl3 |
   plane    --------                        ---------   ---------
   - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
                                ------         ------     ------
                               .| f3 |.........| f5 |.....| f6 |
          ------       ------ . ------         ------     ------
    user  | f1 |.......| f2 |.                    .
   plane  ------       ------ . ------            .
                               .| f4 |.............
                                ------
   - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
         +--------------------------------+  +-------------------+
         | -------   --------   --------  |  |      ----------   |
         | |     |   |      |   |      |  |  |    ---------- |   |
         | | @UE |   | @car |   | @eNB |  |  |  ---------- | |   |
         | -------   --------   --------  |  |  |  Data  | | |   |
         |                                |  |  | Center | | -   |
         | -------- Heterogeneous ------- |  |  |  (DC)  |-      |
    phy  | |      |   computing   |     | |  |  ----------       |
   infra | |@train|    devices    | @AP | |==|      ----------   |
         | --------    forming    ------- |  |    ---------- |   |
         |             the fog            |  |  ---------- | |   |
         | ---------        ------------  |  |  |  Data  | | |   |
         | |       |        |          |  |  |  | Center | | -   |
         | | @mall |        | @localDC |  |  |  |  (DC)  |-      |
         | ---------        ------------  |  |  ----------       |
         |              FOG               |  |       CLOUD       |
         +--------------------------------+  +-------------------+
         <--------- fog and edge ----------------->
                                     <--- edge & central cloud --->
```

Figure 3: Fog virtualization

5. Problem statement

Virtualized resources do not need to be limited to those available in traditional data centers, where the infrastructure is stable, static, typically homogeneous and managed by a single admin entity. Computational capabilities are becoming more and more ubiquitous, with terminal devices getting extremely powerful, as well as other types of devices that are close to the end users at the edge (e.g., vehicular onboard devices for infotainment, micro data centers deployed at the edge, etc.). It is envisioned that these devices would be able to offer storage, computing and networking resources to nearby network infrastructure, devices and things (the fog paradigm). These resources can be used to host functions, for example to offload/complement other resources available at traditional data

centers, but also to reduce the end-to-end latency or to provide access to specialized information (e.g., context available at the edge) or hardware.

In this draft, we consider that a mobile terminal may: (i) provide resources for others to be used, by integrating them into an existing virtualization infrastructure (either fixed or mobile); and/or (ii) consume resources offered by others, by integrating them into the set of resources under the management of the given mobile terminal. WE look at how to enable virtualization infrastructures to dynamically integrate resources that are mobile and volatile (because either the terminal hosting the resources is mobile/volatile or the terminal controlling them is mobile/volatile). Since the fog resources are volatile, i.e. may dynamically appear and disappear, and may be mobile, i.e. may move from one place to another, mechanisms to discover and advertise virtualized fog resources are required.

Taking the ETSI NFV architecture (see Section 3) as a baseline for the virtualization of the fog nodes, the discovery of a virtualization resource can be done either through (i) the discovery of NFVI from a VIM; or through (ii) the discovery of VIMs and associated NFVI from an NFVO. In this draft, we focus on the alternative (ii), that is, the discovery of the VIMs and NFVI1 from an NFVO. Both mobile VIM+NFVI, and mobile NFVO are in the scope of the document.

The relationship between an NFVO and the resources it is capable to orchestrate through a VIM is statically defined according to the current ETSI NFV specifications [etsi_nfv_002] [etsi_nfv_ifa_005]. The interface Or-Vi (between NFVO and VIM) [etsi_nfv_ifa_005] does not include any discovery and automatic registration of (mobile) VIMs from a (mobile) NFVO. Therefore, currently there is no standardized mechanism defined for such a discovery and registration specified by ETSI or any other SDO. This is the gap addressed by this draft.

We cover two different scenarios:

- o A mobile terminal (hosting mobile resources) joins a network where there is an existing virtualization infrastructure. The mobile terminal hosts both some kind of NFVI (resources) plus a VIM (in charge of managing those resources and providing an appropriate interfaces for others to use and control them).

- o A mobile terminal (looking for available resources) joins a network where there are virtualization resources available. The mobile terminal hosts a NFVO, capable of integrating and controlling others' virtual resources.

6. Advertisement and discovery of mobile resources (VIM+NFVI)

This document describes IPv6 extensions to allow discovery of virtualization resources, in the form of a VIM + associated NFVI. Examples of scenarios where this is useful are shown in Figure 4 and Figure 5, including also a high-level view of the solution.

```
                                                    __
                  ___________                     _(  )_
    ----------  _(           )_    -----------  _(      )_
    | device |-(_  VIM--NFVI  _)   | network |-(_  NFVO  _)
    ----------   (___________)     -----------   (_    _)
        |                               |          (__)
       XXX (1. attachment)              |
        |                               |
        +---2. Advertisement----------->|
        |                               |
        |<......(3. VIM Registration)..>|
        |                               |
```

Figure 4: VIM+NFVI advertisement

Figure 4 shows an scenario in which a mobile device with available resources (NFVI, and associated VIM) attaches to a network (step 1). Then, it advertises (step 2) that it has virtualization resources (and their characteristics, such as the type of VIM) that could be eventually used. An NFVO sitting in the network can then decide to register the VIM for later use (step 3). This document specifies some options for step 2 based on IP signaling. Step 3 is implementation dependent and very much VIM-NFVO specific.

Similarly, Figure 5 shows a scenario with a mobile NFVO. A mobile device with an embedded NFVO attaches to a network (step 1). Then, it queries the network (step 2) to learn if there are virtualization resources available. If so, the network conveys that information (step 3). The NFVO can then decide to register the VIM for later use (step 4). This document specifies some options for steps 2 and 3 based on IP signaling. Step 4 is implementation dependent and very much VIM-NFVO specific.

```
                                                      ___________
                                                    _(           )_
                    ______                        _(      +-NFVI   )_
    ------------  _(      )_       -----------  _(       /           )_
    | terminal |-(_  NFVO  _)      | network |-(_  VIM(s)---NFVI      _)
    ------------   (______)        -----------   (_      \          _)
        |                               |          (_     +-NFVI  _)
       XXX (1. attachment)              |            (___________)
        |                               |
        +---2. Request----------------->|
        |                               |
        |<-----------3. Advertisement---|
        |                               |
        |<..(4. VIM Registration)......>|
        |                               |
```

Figure 5: VIM+NFVI discovery

6.1. IPv6 ND-based discovery

This section describes a solution based on IPv6 Neighbor Discovery [RFC4861]. The solution is based on defining a new set of options to convey information about available virtualization resources, including optional attributes. In such a way, it is possible to discover VIM+NFVI resources available at:

- A mobile device connecting to the network, such as a smartphone or a device embedded in a vehicle. This device might have some available resources that other mobile devices, or the network infrastructure can opportunistically use.

- The network infrastructure, e.g., at the edge, like micro-data centers deployed at the very edge of the network. Mobile devices can use these available resources to computationally offload some tasks that require low latency and/or information that is only available at the edge (such as radio related information).

The discovery of available resources (VIM+NFVI) is based on a combination of proactive and reactive advertisement. IPv6 Neighbor Discovery (ND) [RFC4861] is a very good approach to convey this information as, (i) it is widely deployed, (ii) it is very lightweight and easy to implement, (iii) it allows dynamic updates due to network topology updates (e.g., a device connecting/ disconnecting from a network), and (iv) it is independent on the network access technology.

The basic operation of ND-based VIM+NFVI discovery consists in the advertisement of virtual resources in IPv6 ND messages from the

device hosting those virtual resources. This can be done, for example by a mobile host sending unsolicited Neighbor Advertisement (NA) messages (or in response to a Neighbor Solicitation, NS) including the new VIM+NFVI options -- as shown in Figure 6 -- or even including them in Router Solicitations. Another example would be the network infrastructure advertising available resources by including VIM+NFVI options in Router Advertisement (RA) or Neighbor Advertisement messages -- as shown in Figure 7.

```
                                                    __
                  ___________                     _(  )_
    ----------  _(           )_    -----------  _(      )_
    | device |-(_  VIM--NFVI  _)   | network |-(_  NFVO  _)
    ----------   (___________)     -----------   (_    _)
        |                               |          (__)
        +--Unsolicited Neigh. Advert.-->|
        |  (incld. VIM+NFVI opt.)       |
        |                               |
```

Figure 6: Example of VIM+NFVI advertisement via unsolicited NA

```
                                                      ___________
                                                    _(           )_
                    ______                        _(      +-NFVI   )_
    ------------  _(      )_       -----------  _(       /           )_
    | terminal |-(_  NFVO  _)      | network |-(_  VIM(s)---NFVI      _)
    ------------   (______)        -----------   (_      \          _)
        |                               |          (_     +-NFVI  _)
        |<--------Router Advertisement--+            (___________)
        |        (incld. VIM+NFVI opt.) |
        |                               |
```

Figure 7: Example of VIM+NFVI advertisement via RA

6.2. VIM+NFVI options

New ND VIM+NFVI options are defined to be used with Neigbor Solicitation, Neighbor Advertisement, Router Solicitation and Router Advertisement options. The presence of any of these options is used to signal the availability of VIM+NFVI. These options are used to convey information of associated attributes, like:

- o Available Virtualized Compute Resources.

- o Available Virtualized Storage Resources.

- o Available Virtualized Networking Resources.

o Type of virtualization e.g., full virtualization, para virtualization, hybrid virtualization.

o Available hypervisor e.g., bare metal or hosted hypervisor.

o Supported virtual machine images or container format.

o Power profile, e.g., battery or mains powered, battery capacity, charge status, etc.

o Volatility profile, e.g., expected availability.

o Type of VIM and version.

o Protocol APIs supported by the VIM.

o URI of the VIM.

The format of these options is described next. Note that this list is just an example and that additional options could be added.

6.2.1. Available Virtualized Compute Resources

The format of this option is shown below. This option should be padded when necessary to ensure that they end on their natural 64-bit boundaries, as specified in [RFC4861].

```
 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        |    Length     |N|        Reserved0            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    cpuArch    | numVirtualCpu |       virtualCpuClock         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  accelCapab   | vCpuOP| vMemOP|        virtualMemSize         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Reserved1                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Type

   To be assigned by IANA.

Length

   2

N

1-bit NUMA supported flag. When set, indicates that the memory allocation can be cognisant of the relevant process/core allocation.

Reserved0

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

cpuArch

8-bit identifier indicating the type CPU architecture type. Examples are: 1 (x86), 2 (ARM).

numVirtualCpu

8-bit unsigned integer. Indicates the number of virtual CPUs.

virtualCpuClock

16-bit unsigned integer. Indicates the Minimum virtual CPU clock rate (in MHz).

accelCapab

8-bit mask indicating the acceleration capabilities. Examples are: 1 (crypto), 2 (GPU).

vCpuOP

8-bit unsigned integer. Indicates the CPU core oversubscription policy, e.g. the relation of virtual CPU cores to physical CPU cores/threads. A value of 0 indicates that no concrete policy is defined.

vMemOP

8-bit unsigned integer. Indicates the memory core oversubscription policy in terms of virtual memory to physical memory on the platform. A value of 0 indicates that no concrete policy is defined.

Reserved1

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

6.2.2. Available Virtualized Storage Resources

The format of this option is shown below. This option should be padded when necessary to ensure that they end on their natural 64-bit boundaries, as specified in [RFC4861].

```
 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        |    Length     |         sizeOfStorage         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                            Reserved                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Type

To be assigned by IANA.

Length

1

sizeOfStorage

16-bit unsigned integer. Indicates the Size of virtualised storage resource (in GB).

Reserved

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

6.2.3. Available Virtualized Networking Resources

The format of this option is shown below. This option should be padded when necessary to ensure that they end on their natural 64-bit boundaries, as specified in [RFC4861].

```
 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        |    Length     |            bandwidth          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  networkType  |                  Reserved                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Type

To be assigned by IANA.

Length

1

bandwidth

16-bit unsigned integer. Indicates the minimum network bandwidth (in Mbps).

networkType

8-bit unsigned identifier. Indicates the type of network that maps to the virtualised network. Examples are: 1 (local), 2 (vlan), 3 (vxlan), 4(gre).

Reserved

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

6.2.4. Type of virtualization

The format of this option is shown below. This option should be padded when necessary to ensure that they end on their natural 64-bit boundaries, as specified in [RFC4861].

```
 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        |     Length    |   virtType    |   hypervisor  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                            Reserved                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Type

To be assigned by IANA.

Length

1

virtType

8-bit identifier indicating the type of virtualization. Examples are: 1 (full virtualization), 2 (para virtualization), 3 (hybrid virtualization).

hypervisor

8-bit identifier indicating the type of hypervisor (if applicable). Examples are: 0 (not applicable), 1 (type 1), 2 (type 2).

Reserved

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

6.2.5. Power profile

The format of this option is shown below. This option should be padded when necessary to ensure that they end on their natural 64-bit boundaries, as specified in [RFC4861].

```
 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        |     Length    |B|C|   BatStat |   Reserved0   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           Reserved1                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Type

To be assigned by IANA.

Length

1

B

1-bit Battery-powered flag. When set, indicates that the sending device is battery powered.

C

1-bit Charging flag. If the B flag is set to 0, this MUST be set to 0. When set, indicates that the battery is charging.

BatStat

6-bit integer indicating the charge of the charge of the Battery. If the B flag is set to 0, this MUST be set to 0. A value of 64 indicates that the battery is full.

Reserved0

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

Reserved1

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

6.2.6. Volatility profile

The format of this option is shown below. This option should be padded when necessary to ensure that they end on their natural 64-bit boundaries, as specified in [RFC4861].

```
 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        |    Length     |      ExpectedAvailability     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                            Reserved                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Type

To be assigned by IANA.

Length

1

ExpectedAvailability

16-bit integer indicating the expected availability (in units of seconds). This is an estimation from the sender. How this is set is implementation dependent.

Reserved

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

6.2.7. URI of the VIM

The format of this option is shown below. This option should be padded when necessary to ensure that they end on their natural 64-bit boundaries, as specified in [RFC4861].

```
 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        |    Length     |            Reserved           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                             VimUri                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Type

To be assigned by IANA.

Length

1

Reserved

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

VimUri

A variable-length encoded string containing the URI of the VIM.

7. IANA Considerations

TBD.

8. Security Considerations

TBD.

9. Acknowledgments

The work in this draft will be further developed and explored under the framework of the H2020 5G-CORAL project (Grant 761586).

10. References

10.1. 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>.

[RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, DOI 10.17487/RFC4861, September 2007, <https://www.rfc-editor.org/info/rfc4861>.

10.2. Informative References

[etsi_nfv_002] ""Network Functions Virtualization (NFV); Architectural Framework," ETSI GS NFV 002 v1.1.1", October 2013.

[etsi_nfv_ifa_005] ""Network Functions Virtualisation (NFV) Release 2; Management and Orchestration; Or-Vi reference point - Interface and Information Model Specification," ETSI GS NFV-IFA 005 V2.3.1", August 2017.

[etsi_nfv_whitepaper] "Network Functions Virtualisation (NFV). White Paper 2", October 2014.

[I-D.aranda-sfc-dp-mobile] Gutierrez, P., Gramaglia, M., Lopez, D., and W. Haeffner, "Service Function Chaining Dataplane Elements in Mobile Networks", draft-aranda-sfc-dp-mobile-04 (work in progress), June 2017.

[nfv_sota_research_challenges] Mijumbi, R., Serrat, J., Gorricho, J-L., Bouten, N., De Turck, F., and R. Boutaba, "Network Function Virtualization: State-of-the-art and Research Challenges", IEEE Communications Surveys & Tutorials Volume: 18, Issue: 1, September 2015.

[onf_sdn_architecture] "SDN Architecture (Issue 1.1), ONF TR-521", February 2016.

[RFC8568] Bernardos, CJ., Rahman, A., Zuniga, JC., Contreras, LM., Aranda, P., and P. Lynch, "Network Virtualization Research Challenges", RFC 8568, DOI 10.17487/RFC8568, April 2019, <https://www.rfc-editor.org/info/rfc8568>.

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/