Internet Research Task Force (IRTF)
Request for Comments: 8568
Category: Informational
ISSN: 2070-1721

CJ. Bernardos
UC3M
A. Rahman
InterDigital
JC. Zuniga
SIGFOX
LM. Contreras
TID
P. Aranda
UC3M
P. Lynch
Keysight Technologies

April 2019

Network Virtualization Research Challenges

Abstract

This document describes open research challenges for network virtualization. Network virtualization is following a similar path as previously taken by cloud computing. Specifically, cloud computing popularized migration of computing functions (e.g., applications) and storage from local, dedicated, physical resources to remote virtual functions accessible through the Internet. In a similar manner, network virtualization is encouraging migration of networking functions from dedicated physical hardware nodes to a virtualized pool of resources. However, network virtualization can be considered to be a more complex problem than cloud computing as it not only involves virtualization of computing and storage functions but also involves abstraction of the network itself. This document describes current research and engineering challenges in network virtualization including the guarantee of quality of service, performance improvement, support for multiple domains, network slicing, service composition, device virtualization, privacy and security, separation of control concerns, network function placement, and testing. In addition, some proposals are made for new activities in the IETF and IRTF that could address some of these challenges. This document is a product of the Network Function Virtualization Research Group (NFVRG).

Status of This Memo

This document is not an Internet Standards Track specification; it is published for informational purposes.

This document is a product of the Internet Research Task Force (IRTF). The IRTF publishes the results of Internet-related research and development activities. These results might not be suitable for deployment. This RFC represents the consensus of the Network Function Virtualization Research Group of the Internet Research Task Force (IRTF). Documents approved for publication by the IRSG are not candidates for any level of Internet Standard; see Section 2 of RFC 7841.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at https://www.rfc-editor.org/info/rfc8568.

Table of Contents

1. Introduction and Scope

The telecommunications sector is experiencing a major revolution that will shape the way networks and services are designed and deployed for the next few decades. In order to cope with continuously increasing demand and cost, network operators are taking lessons from the IT paradigm of cloud computing. This new approach of virtualizing network functions will enable multi-fold advantages by moving communication services from bespoke hardware in the operator's core network to Commercial Off-The-Shelf (COTS) equipment distributed across data centers.

Some of the network virtualization mechanisms that are being considered include the following: sharing of network infrastructure to reduce costs, virtualization of core and edge servers/services running in data centers as a way of supporting their load-aware elastic dimensioning, and dynamic energy policies to reduce the electricity consumption.

This document presents research and engineering challenges in network virtualization that need to be addressed in order to achieve these goals, spanning from pure research and engineering/standards space. The objective of this memo is to document the technical challenges and corresponding current approaches and to expose requirements that should be addressed by future research and standards work.

This document represents the consensus of the Network Function Virtualization Research Group (NFVRG). It has been reviewed by the RG members active in the specific areas of work covered by the document.

2. Terminology

The following terms used in this document are defined by the ETSI Network Function Virtualization (NFV) Industrial Study Group (ISG) [etsi_gs_nfv_003], the Open Networking Foundation (ONF) [onf_tr_521], and the IETF [RFC7426] [RFC7665]:

Application Plane: The collection of applications and services that program network behavior.

Control Plane (CP): The collection of functions responsible for controlling one or more network devices. The CP instructs network devices with respect to how to process and forward packets. The control plane interacts primarily with the forwarding plane and, to a lesser extent, with the operational plane.

Forwarding Plane (FP): The collection of resources across all network devices responsible for forwarding traffic.

Management Plane (MP): The collection of functions responsible for monitoring, configuring, and maintaining one or more network devices or parts of network devices. The management plane is mostly related to the operational plane (it is related less to the forwarding plane).

NFV Infrastructure (NFVI): Totality of all hardware and software components that 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) life cycle and coordinates the management of NS life cycle, VNF life cycle (supported by the VNFM) and NFVI resources (supported by the VIM) to ensure an optimized allocation of the necessary resources and connectivity.

Operational Plane (OP): The collection of resources responsible for managing the overall operation of individual network devices.

Physical Network Function (PNF): Physical implementation of a network function in a monolithic realization.

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

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

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

Virtualized Network Function Manager (VNFM): Functional block that is responsible for the life-cycle management of VNF.

3. Background

This section briefly describes some basic background technologies as well as other Standards Developing Organizations (SDOs) and open-source initiatives working on network virtualization or related topics.

3.1. Network Function Virtualization

The ETSI ISG Network Function Virtualization (NFV) is a working group that, since 2012, has aimed 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 is composed of three domains (Figure 1):

- Virtualized Network Function, running over the NFVI.
- 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.

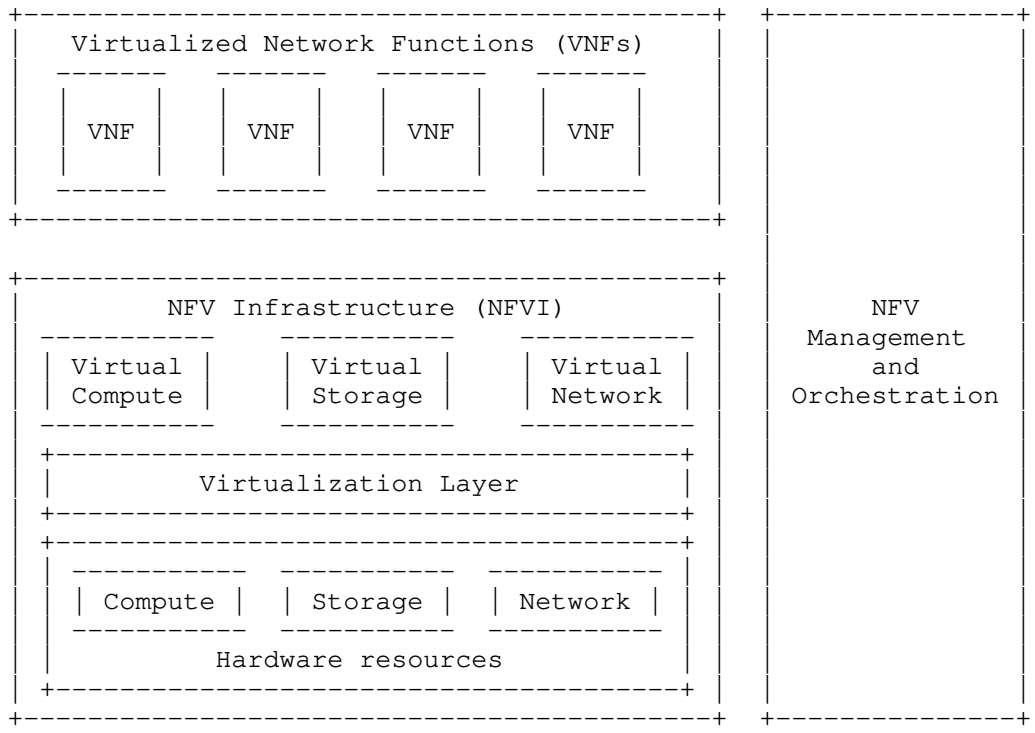


Figure 1: ETSI NFV Framework

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 as well as the Virtualization Layer.
- Virtualized Infrastructure Manager(s) (VIM)
- NFV Orchestrator
- VNF Manager(s)
- Service, VNF and Infrastructure Description

- Operational Support Systems and Business Support Systems (OSS and BSS)

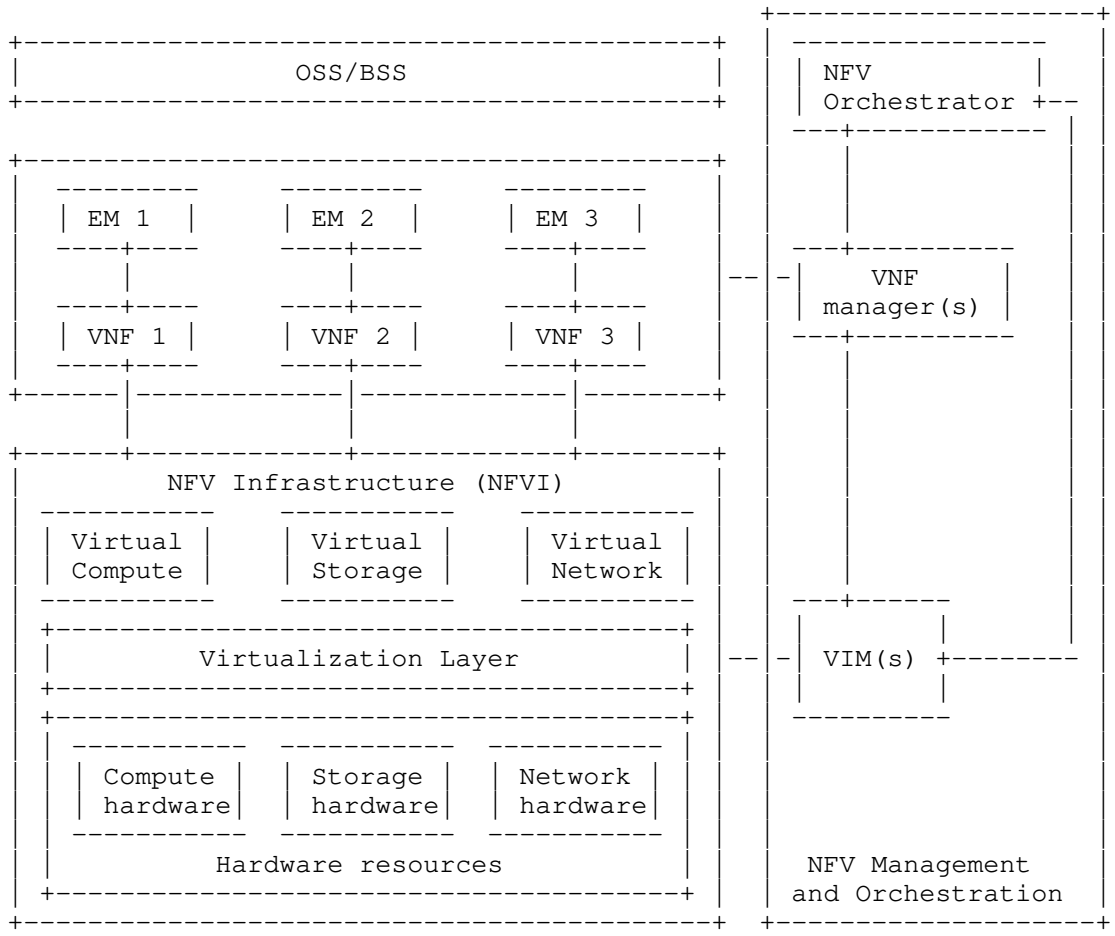


Figure 2: ETSI NFV Reference Architecture

3.2. Software-Defined Networking

The Software-Defined Networking (SDN) paradigm pushes the intelligence currently residing in the network elements to a central controller implementing the network functionality through software. In contrast to traditional approaches, in which the network's control plane is distributed throughout all network devices, with SDN, the control plane is logically centralized. In this way, the deployment of new characteristics in the network no longer requires complex and costly changes in equipment or firmware updates, but only a change in the software running in the controller. The main advantage of this approach is the flexibility it provides operators to manage their network, i.e., an operator can easily change its policies on how traffic is distributed throughout the network.

One of the most well-known protocols for the SDN control plane between the central controller and the networking elements is the OpenFlow Protocol (OFP), which is maintained and extended by the Open Network Foundation (ONF) <https://www.opennetworking.org/>. Originally, this protocol was developed specifically for IEEE 802.1 switches conforming to the ONF OpenFlow Switch specification [OpenFlow]. As the benefits of the SDN paradigm have reached a wider audience, its application has been extended to more complex scenarios such as wireless and mobile networks. Within this area of work, the ONF is actively developing new OFP extensions addressing three key scenarios: (i) wireless backhaul, (ii) cellular Evolved Packet Core (EPC), and (iii) unified access and management across enterprise wireless and fixed networks.

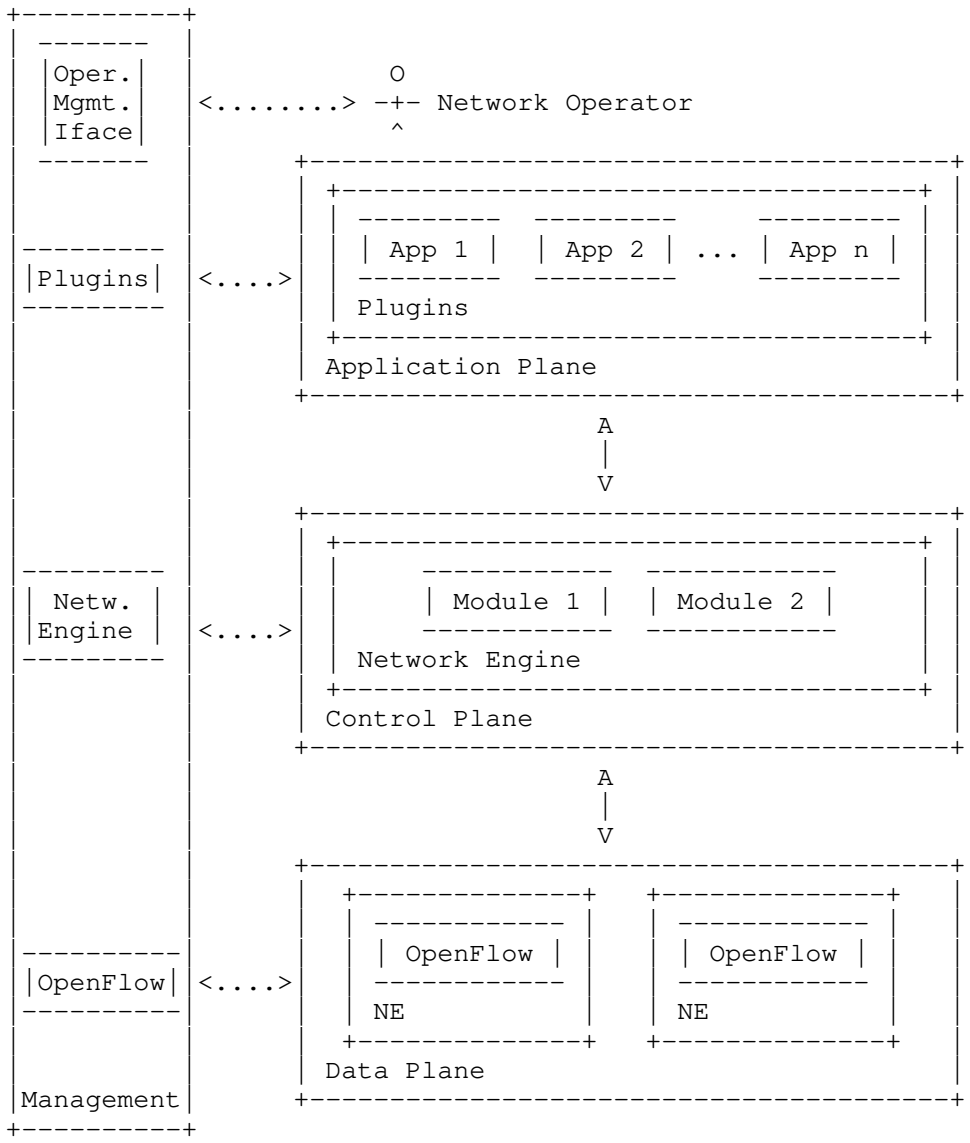


Figure 3: High-Level SDN ONF Architecture

Figure 3 shows the blocks and the functional interfaces of the ONF architecture, which comprises three planes: data, controller, and application. The data plane comprehends several Network Entities (NEs), which expose their capabilities toward the control plane via a Southbound API. The control plane includes several cooperating modules devoted to the creation and maintenance of an abstracted

resource model of the underlying network. Such a model is exposed to the applications via a Northbound API where the application plane comprises several applications/services, each of which has exclusive control of a set of exposed resources.

The management plane spans its functionality across all planes performing the initial configuration of the network elements in the data plane, the assignment of the SDN controller and the resources under its responsibility. In the control plane, the management needs to configure the policies defining the scope of the control given to the SDN applications, to monitor the performance of the system and to configure the parameters required by the SDN controller modules. In the application plane, the management plane configures the parameters of the applications and the service-level agreements. In addition to these interactions, the management plane exposes several functions to network operators that can easily and quickly configure and tune the network at each layer.

In RFC 7426 [RFC7426], the IRTF Software-Defined Networking Research Group (SDNRG) documented a layer model of an SDN architecture. This was due to the following controversial discussion topics (among others). What exactly is SDN? What is the layer structure of the SDN architecture? How do layers interface with each other?

Figure 4 reproduces the figure included in RFC 7426 [RFC7426] to summarize the SDN architecture abstractions in the form of a detailed, high-level schematic. In a particular implementation, planes can be collocated with other planes or can be physically separated.

In SDN, a controller manipulates controlled entities via an interface. Interfaces, when local, are mostly API invocations through some library or system call. However, such interfaces may be extended via some protocol definition, which may use local interprocess communication (IPC) or a protocol that could also act remotely; the protocol may be defined as an open standard or in a proprietary manner.

SDN expands multiple planes: forwarding, operational, control, management, and application. All planes mentioned above are connected via interfaces. Additionally, RFC 7426 [RFC7426] considers four abstraction layers: the Device and resource Abstraction Layer (DAL), the Control Abstraction Layer (CAL), the Management Abstraction Layer (MAL), and the Network Services Abstraction Layer (NSAL).

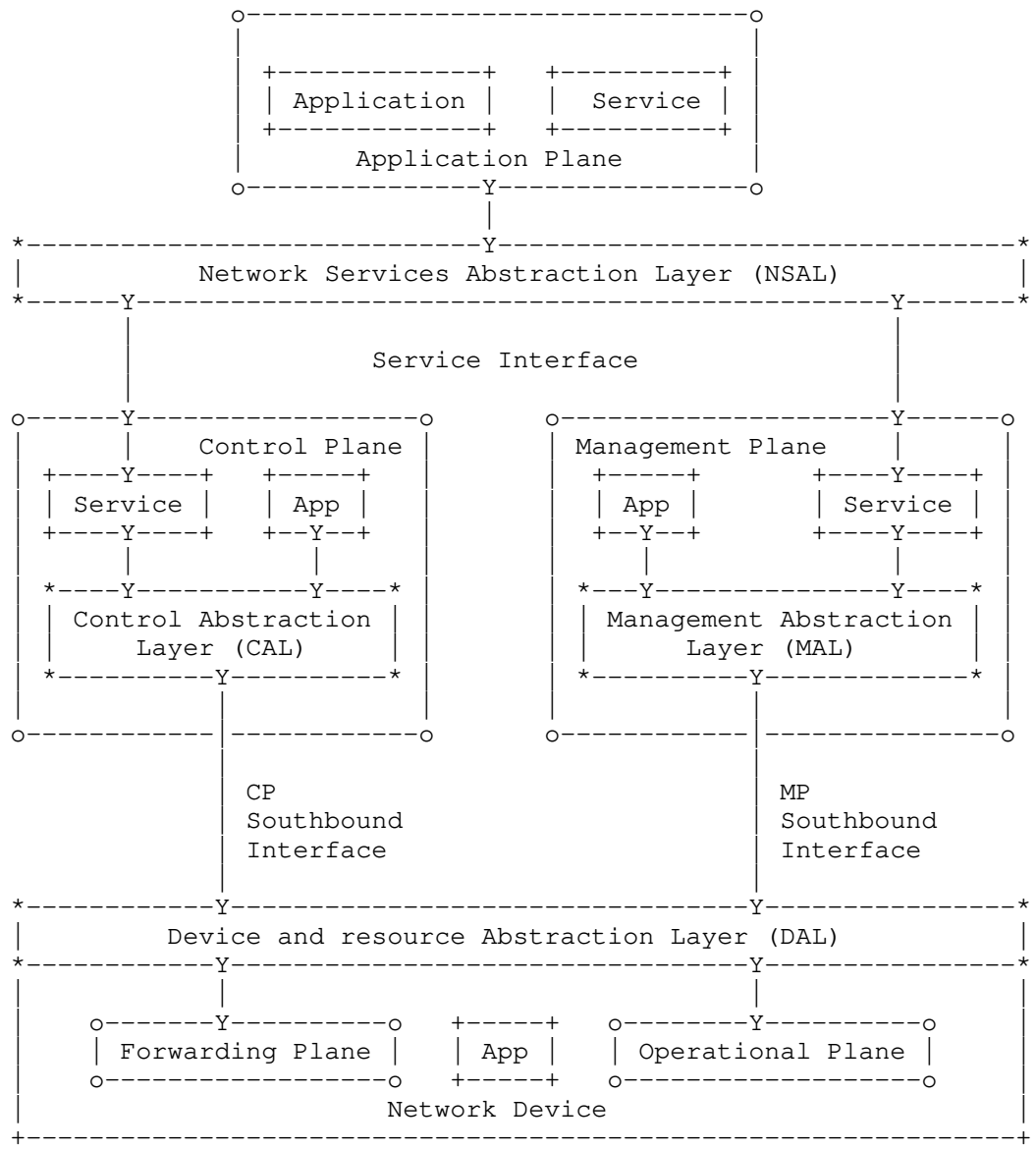


Figure 4: SDN-Layer Architecture

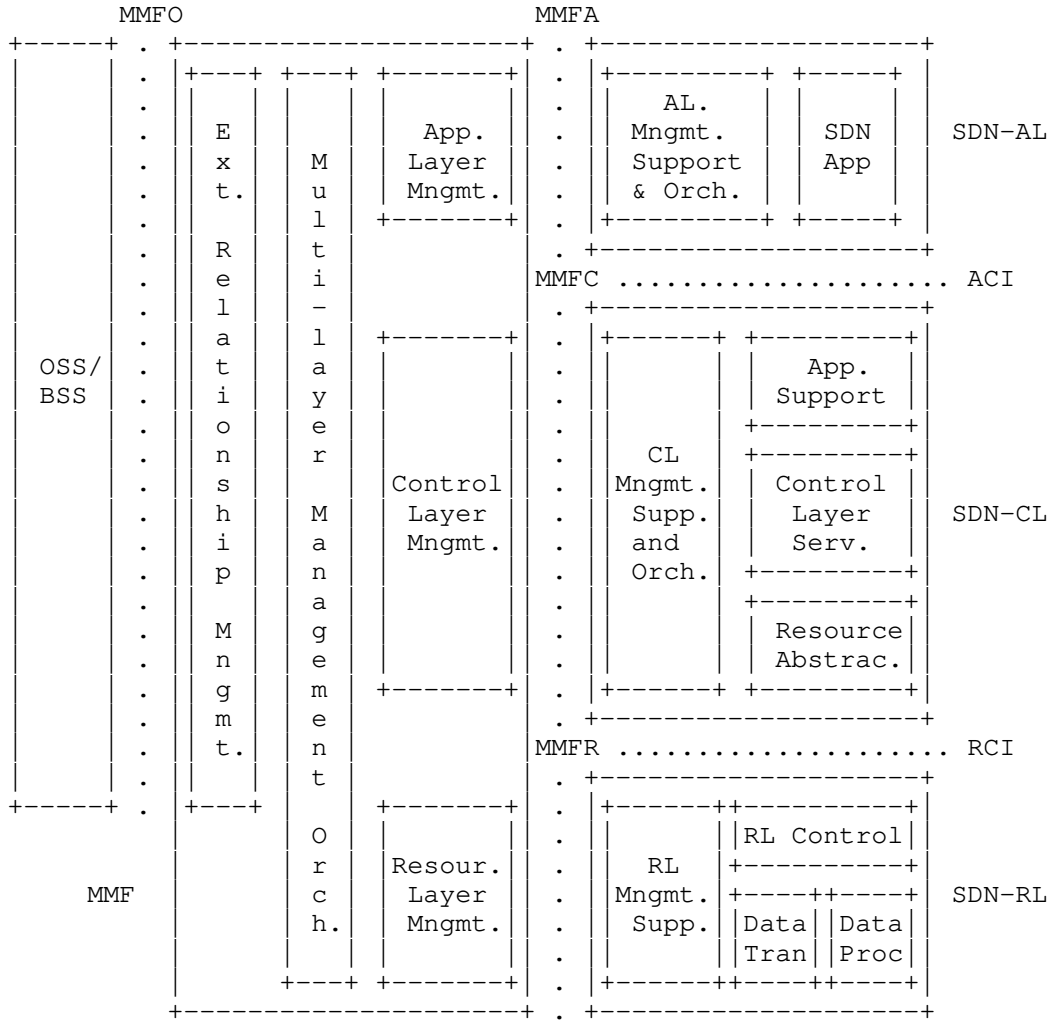
While SDN is often directly associated to OpenFlow, this is just one (relevant) example of a southbound protocol between the central controller and the network entities. Other relevant examples of protocols in the SDN family are NETCONF [RFC6241], RESTCONF [RFC8040], and ForCES [RFC5810].

3.3. ITU-T Functional Architecture of SDN

The ITU-T (the Telecommunication standardization sector of the International Telecommunication Union) has also looked into SDN architectures, defining a slightly modified one from what other SDOs have done. In ITU-T recommendation Y.3302 [itu-t-y.3302], the ITU-T provides a functional architecture of SDN with descriptions of functional components and reference points. The described functional architecture is intended to be used as an enabler for further studies on other aspects such as protocols and security as well as being used to customize SDN in support of appropriate use cases (e.g., cloud computing, mobile networks). This recommendation is based on ITU-T Y.3300 [itu-t-y.3300] and ITU-T Y.3301 [itu-t-y.3301]. While the first describes the framework of SDN (including definitions, objectives, high-level capabilities, requirements, and the high-level architecture of SDN), the second describes more-detailed requirements.

Figure 5 shows the SDN functional architecture defined by the ITU-T. It is a layered architecture composed of the SDN application layer (SDN-AL), the SDN control layer (SDN-CL), and the SDN resource layer (SDN-RL). It also has multi-layer management functions (MMF), which provide the ability to manage the functionalities of SDN layers, i.e., SDN-AL, SDN-CL, and SDN-RL. MMF interacts with these layers using Multi-layer Management Functions Application (MMFA), Multi-layer Management Functions Control (MMFC), and Multi-layer Management Functions Resource (MMFR) reference points.

The SDN-AL enables a service-aware behavior of the underlying network in a programmatic manner. The SDN-CL provides programmable means to control the behavior of SDN-RL resources (such as data transport and processing) following requests received from the SDN-AL according to MMF policies. The SDN-RL is where the physical or virtual network elements perform transport and/or processing of data packets according to SDN-CL decisions.



Legend:
- ACI: Application Control Interface
- MMFA: Multi-layer Management Functions Application
- MMFC: Multi-layer Management Functions Control
- MMFO: Multi-layer Management Functions OSS/BSS
- MMFR: Multi-layer Management Functions Resource
- RCI: Resource Control Interfaces
- RL: Resource Layer

Figure 5: ITU-T SDN Functional Architecture

3.4. Multi-Access Edge Computing

Multi-access Edge Computing (MEC) -- formerly known as Mobile Edge Computing -- capabilities deployed in the edge of the mobile network can facilitate the efficient and dynamic provision of services to mobile users. The ETSI ISG MEC working group, operative from end of 2014, intends to specify an open environment for integrating MEC capabilities with service providers' networks, also including applications from third parties. These distributed computing capabilities provide IT infrastructure as in a cloud environment for the deployment of functions in mobile access networks. It can be seen then as a complement to both NFV and SDN.

3.5. IEEE 802.1CF (OmniRAN)

The IEEE 802.1CF Recommended Practice [omniran] specifies an access network that connects terminals to their access routers utilizing technologies based on the family of IEEE 802 Standards (e.g., 802.3 Ethernet, 802.11 Wi-Fi, etc.). The specification defines an access network reference model, including entities and reference points along with behavioral and functional descriptions of communications among those entities.

The goal of this project is to help unify the support of different interfaces, enabling shared-network control and use of SDN principles, thereby lowering the barriers to new network technologies, to new network operators, and to new service providers.

3.6. Distributed Management Task Force (DMTF)

The DMTF <https://www.dmtf.org/> is an industry standards organization working to simplify the manageability of network-accessible technologies through open and collaborative efforts by some technology companies. The DMTF is involved in the creation and adoption of interoperable management standards, supporting implementations that enable the management of diverse traditional and emerging technologies including cloud, virtualization, network, and infrastructure.

There are several DMTF initiatives that are relevant to the network virtualization area, such as the Open Virtualization Format (OVF) for VNF packaging; the Cloud Infrastructure Management Interface (CIMI) for cloud infrastructure management; the Network Management (NETMAN), for VNF management; and the Virtualization Management (VMAN), for virtualization infrastructure management.

3.7. Open-Source Initiatives

The open-source community is especially active in the area of network virtualization and orchestration. We next summarize some of the active efforts:

- OpenStack. OpenStack is a free and open-source cloud-computing software platform. OpenStack software controls large pools of compute, storage, and networking resources throughout a data center, managed through a dashboard or via the OpenStack API.

- Kubernetes. Kubernetes is an open-source system for automating deployment, scaling and management of containerized applications. Kubernetes can schedule and run application containers on clusters of physical or virtual machines. Kubernetes allows (i) Scale on the fly, (ii) Limit hardware usage to required resources only, (iii) Load-balancing Monitoring, and (iv) Efficient life-cycle management.

- OpenDayLight. OpenDayLight (ODL) is a highly available, modular, extensible and scalable multiprotocol controller infrastructure built for SDN deployments on modern heterogeneous multi-vendor networks. It provides a model-driven service abstraction platform that allows users to write apps that easily work across a wide variety of hardware and southbound protocols.

- ONOS. The Open Network Operating System (ONOS) project is an open-source community hosted by The Linux Foundation. The goal of the project is to create an SDN operating system for communications service providers that is designed for scalability, high performance, and high availability.

- OpenContrail. OpenContrail is a licensed Apache 2.0 project that is built using standards-based protocols and that provides all the necessary components for network virtualization: an SDN controller, a virtual router, an analytics engine, and published northbound APIs. It has an extensive Representational State Transfer (REST) API to configure and gather operational and analytics data from the system.

- OPNFV. The Open Platform for NFV (OPNFV) is a carrier-grade, integrated, open-source platform to accelerate the introduction of new NFV products and services. By integrating components from upstream projects, the OPNFV community aims at conducting performance and use case-based testing to ensure the platform's suitability for NFV use cases. The scope of OPNFV's initial release is focused on building NFV Infrastructure (NFVI) and Virtualized Infrastructure Manager (VIM) by integrating components

from upstream projects such as OpenDayLight, OpenStack, Ceph Storage, Kernel-based Virtual Machine (KVM), Open vSwitch, and Linux. These components, along with APIs to other NFV elements, form the basic infrastructure required for Virtualized Network Functions (VNFs) and Management and Orchestration (MANO) components. OPNFV's goal is to (i) increase performance and power efficiency, (ii) improve reliability, availability, and serviceability, and (iii) deliver comprehensive platform instrumentation.

- OSM. Open Source Mano (OSM) is an ETSI-hosted project to develop an Open Source NFV Management and Orchestration (MANO) software stack aligned with ETSI NFV. OSM is based on components from previous projects, such Telefonica's OpenMANO or Canonical's Juju, among others.

- OpenBaton. OpenBaton is a Network Function Virtualization Orchestrator (NFVO) that is ETSI NFV compliant. OpenBaton was part of the OpenSDNCore project started with the objective of providing a compliant implementation of the ETSI NFV specification.

- ONAP. Open Network Automation Platform (ONAP) is an open-source software platform that delivers capabilities for the design, creation, orchestration, monitoring, and life-cycle management of (i) Virtual Network Functions (VNFs), (ii) The carrier-scale Software-Defined Networks (SDNs) that contain them, and (iii) higher-level services that combine the above. ONAP (derived from the AT&T's ECOMP) provides for automatic, policy-driven interaction of these functions and services in a dynamic, real-time cloud environment.

- SONA. The Simplified Overlay Network Architecture (SONA) is an extension to ONOS to have an almost full SDN network control in OpenStack for virtual tenant network provisioning. Basically, SONA is an SDN-based network virtualization solution for cloud DC.

Among the main areas that are being developed by the aforementioned open-source activities that relate to network virtualization research, we can highlight policy-based resource management, analytics for visibility and orchestration, and service verification with regard to security and resiliency.

4. Network Virtualization Challenges

4.1. Overview

Network virtualization is changing the way the telecommunications sector will deploy, extend, and operate their networks. These new technologies aim at reducing the overall costs by moving communication services from specific hardware in the operators' cores to server farms scattered in data centers (i.e., compute and storage virtualization). In addition, the networks interconnecting the functions that compose a network service are fundamentally affected in the way they route, process, and control traffic (i.e., network virtualization).

4.2. Guaranteeing Quality of Service

Achieving a given QoS in an NFV environment with virtualized and distributed computing, storage, and networking functions is more challenging than providing the equivalent in discrete non-virtualized components. For example, ensuring a guaranteed and stable forwarding data rate has proven not to be straightforward when the forwarding function is virtualized and runs on top of COTS server hardware [openmano_dataplane] [NFV-COTS] [etsi_nfv_whitepaper_3]. Again, the comparison point is against a router or forwarder built on optimized hardware. We next identify some of the challenges that this poses.

4.2.1. Virtualization Technologies

The issue of guaranteeing a network QoS is less of an issue for "traditional" cloud computing because the workloads that are treated there are servers or clients in the networking sense and hardly ever process packets. Cloud computing provides hosting for applications on shared servers in a highly separated way. Its main advantage is that the infrastructure costs are shared among tenants and that the cloud infrastructure provides levels of reliability that can not be achieved on individual premises in a cost-efficient way [intel_10_differences_nfv_cloud]. NFV has very strict requirements posed in terms of performance, stability, and consistency. Although there are some tools and mechanisms to improve this, such as Enhanced Performance Awareness (EPA), Single Root I/O Virtualization (SR-IOV), Non-Uniform Memory Access (NUMA), Data Plane Development Kit (DPDK), etc., these are still unsolved challenges. One open research issue is finding out technologies that are different from Virtual Machines (VMs) and more suitable for dealing with network functionalities.

Lately, a number of lightweight virtualization technologies including containers, unikernels (specialized VMs) and minimalistic distributions of general-purpose OSes have appeared as virtualization

approaches that can be used when constructing an NFV platform. [LIGHT-NFV] describes the challenges in building such a platform and discusses to what extent these technologies, as well as traditional VMs, are able to address them.

4.2.2. Metrics for NFV Characterization

Another relevant aspect is the need for tools for diagnostics and measurements suited for NFV. There is a pressing need to define metrics and associated protocols to measure the performance of NFV. Specifically, since NFV is based on the concept of taking centralized functions and evolving them to highly distributed software (SW) functions, there is a commensurate need to fully understand and measure the baseline performance of such systems.

The IP Performance Metrics (IPPM) WG defines metrics that can be used to measure the quality and performance of Internet services and applications running over transport-layer protocols (e.g., TCP and UDP) over IP. It also develops and maintains protocols for the measurement of these metrics. While the IPPM WG is a long-running WG that started in 1997, at the time of writing, it does not have a charter item or active Internet-Drafts related to the topic of network virtualization. In addition to using IPPM to evaluate QoS, there is a need for specific metrics for assessing the performance of network-virtualization techniques.

The Benchmarking Methodology Working Group (BMWG) is also performing work related to NFV metrics. For example, [RFC8172] investigates additional methodological considerations necessary when benchmarking VNFs that are instantiated and hosted in general-purpose hardware, using bare-metal hypervisors or other isolation environments (such as Linux containers). An essential consideration is benchmarking physical and VNFs in the same way when possible, thereby allowing direct comparison.

There is a clear motivation for the work on performance metrics for NFV [etsi_gs_nfv_per_001], as stated in [RFC8172] (and replicated here):

> I'm designing and building my NFV Infrastructure platform. The first steps were easy because I had a small number of categories of VNFs to support and the VNF vendor gave HW recommendations that I followed. Now I need to deploy more VNFs from new vendors, and there are different hardware recommendations. How well will the new VNFs perform on my existing hardware? Which among several new VNFs in a given category are most efficient in terms of capacity they deliver? And, when I operate multiple categories of VNFs (and PNFs) *concurrently* on a hardware platform such that they

share resources, what are the new performance limits, and what are the software design choices I can make to optimize my chosen hardware platform? Conversely, what hardware platform upgrades should I pursue to increase the capacity of these concurrently operating VNFs?

Lately, there are also some efforts looking into VNF benchmarking. The selection of an NFV Infrastructure Point of Presence to host a VNF or allocation of resources (e.g., virtual CPUs, memory) needs to be done over virtualized (abstracted and simplified) resource views [vnf_benchmarking] [VNF-VBAAS].

4.2.3. Predictive Analysis

On top of diagnostic tools that enable an assessment of the QoS, predictive analyses are required to react before anomalies occur. Due to the SW characteristics of VNFs, a reliable diagnosis framework could potentially enable the prevention of issues by a proper diagnosis and then a reaction in terms of acting on the potentially impacted service (e.g., migration to a different compute node, scaling in/out, up/down, etc.).

4.2.4. Portability

Portability in NFV refers to the ability to run a given VNF on multiple NFVIs, that is, guaranteeing that the VNF would be able to perform its functions with a high and predictable performance given that a set of requirements on the NFVI resources is met. Therefore, portability is a key feature that, if fully enabled, would contribute to making the NFV environment achieve a better reliability than a traditional system. Implementing functionality in SW over "commodity" infrastructure should make it much easier to port/move functions from one place to another. However, this is not yet as ideal as it sounds, and there are aspects that are not fully tackled. The existence of different hypervisors, specific hardware dependencies (e.g., EPA related), or state-synchronization aspects are just some examples of troublemakers for portability purposes.

The ETSI NFV ISG is doing work in relation to portability. [etsi_gs_nfv_per_001] provides a list of minimal features that the VM Descriptor and Compute Host Descriptor should contain for the appropriate deployment of VM images over an NFVI (i.e., a "telco data center"), in order to guarantee high and predictable performance of data-plane workloads while assuring their portability. In addition, [etsi_gs_nfv_per_001] provides a set of recommendations on the minimum requirements that hardware (HW) and hypervisor should have for a "telco data center" suitable for different workloads (data plane, control plane, etc.) present in VNFs. The purpose of

[etsi_gs_nfv_per_001] is to provide the list of VM requirements that should be included in the VM Descriptor template, and the list of HW capabilities that should be included in the Compute Host Descriptor (CHD) to assure predictable high performance. ETSI NFV assumes that the MANO functions will make the mix & match. Therefore, there are still several research challenges to be addressed here.

4.3. Performance Improvement

4.3.1. Energy Efficiency

Virtualization is typically seen as a direct enabler of energy savings. Some of the enablers for this that are often mentioned [nfv_sota_research_challenges] are (i) the multiplexing gains achieved by centralizing functions in data centers reduce the overall energy consumed and (ii) the flexibility brought by network programmability enables to switch off infrastructure as needed in a much easier way. However, there is still a lot of room for improvement in terms of virtualization techniques to reduce the power consumption, such as enhanced-hypervisor technologies.

Some additional examples of research topics that could enable energy savings are [nfv_sota_research_challenges]:

- Energy-aware scaling (e.g., reductions in CPU speeds and partially turning off some hardware components to meet a given energy consumption target.
- Energy-aware function placement.
- Scheduling and chaining algorithms, for example, adapting the network topology and operating parameters to minimize the operation cost (e.g., tracking energy costs to identify the cheapest prices).

Note that it is also important to analyze the trade-off between energy efficiency and network performance.

4.3.2. Improved Link Usage

The use of NFV and SDN technologies can help improve link usage. SDN has already shown that it can greatly increase average link utilization (e.g., Google example [google_sdn_wan]). NFV adds more complexity (e.g., due to service-function chaining / VNF forwarding graphs), which needs to be considered. Aspects like the ones described in [NFVRG-TOPO] (on NFV data center topology design) have to be looked at carefully as well.

4.4. Multiple Domains

Market fragmentation has resulted in a multitude of network operators each focused on different countries and regions. This makes it difficult to create infrastructure services spanning multiple countries, such as virtual connectivity or compute resources, as no single operator has a footprint everywhere. Cross-domain orchestration of services over multiple administrations or over multi-domain single administrations will allow end-to-end network and service elements to mix in multi-vendor, heterogeneous technology, and resource environments [multi-domain_5GEx].

For the specific use case of 'Network as a Service', it becomes even more important to ensure that Cross Domain Orchestration also takes care of hierarchy of networks and their association, with respect to provisioning tunnels and overlays.

Multi-domain orchestration is currently an active research topic, which is being tackled, among others, by ETSI NFV ISG and the 5GEx project <https://www.5gex.eu/> [MULTI-NMRG] [multi-domain_5GEx].

Another side of the multi-domain problem is the integration/ harmonization of different management domains. A key example comes from Multi-access Edge Computing, which, according to ETSI, comes with its own MANO system and would require integration if interconnected to a generic NFV system.

4.5. 5G and Network Slicing

From the beginning of all 5G discussions in the research and industry fora, it has been agreed that 5G will have to address many more use cases than the preceding wireless generations, which first focused on voice services and then on voice and high-speed packet data services. In this case, 5G should be able to handle not only the same (or enhanced) voice and packet data services, but also emerging services like tactile Internet and the Internet of Things (IoT). These use cases take the requirements to opposite extremes, as some of them require ultra-low latency and higher-speed, whereas some others require ultra-low power consumption and high-delay tolerance.

Because of these very extreme 5G use cases, it is envisioned that selective combinations of radio access networks and core network components will have to be combined into a given network slice to address the specific requirements of each use case.

For example, within the major IoT category, which is perhaps the most disrupting one, some autonomous IoT devices will have very low throughput, will have much longer sleep cycles (and therefore high

latency), and a battery life time exceeding by a factor of thousands that of smartphones or some other devices that will have almost continuous control and data communications. Hence, it is envisioned that a customized network slice will have to be stitched together from virtual resources or sub-slices to meet these requirements.

The actual definition of a "network slice" from an IP infrastructure viewpoint is currently undergoing intense debate; see [COMS-PS], [NETSLICES], [SLICE-3GPP], and [ngmn_5G_whitepaper]. Network slicing is a key for introducing new actors in existing markets at a low cost -- by letting new players rent "blocks" of capacity, if the new business model enables performance that meets the application needs (e.g., broadcasting updates to many sensors with satellite broadcasting capabilities). However, more work needs to be done to define the basic architectural approach of how network slices will be defined and formed. For example, is it mostly a matter of defining the appropriate network models (e.g., YANG) to stitch the network slice from existing components? Or do end-to-end timing, synchronization, and other low-level requirements mean that more fundamental research has to be done?

4.5.1. Virtual Network Operators

The widespread use/discussion/practice of system and network virtualization technologies has led to new business opportunities, enlarging the offer of IT resources with virtual network and computing resources, among others. As a consequence, the network ecosystem now differentiates between the owner of physical resources, the Infrastructure Provider (InP), and the intermediary that conforms and delivers network services to the final customers, the Virtual Network Operator (VNO).

VNOs aim to exploit the virtualized infrastructures to deliver new-and-improved services to their customers. However, current network virtualization techniques offer poor support for VNOs to control their resources. It has been considered that the InP is responsible for the reliability of the virtual resources, but there are several situations in which a VNO requires a finer control on its resources. For instance, dynamic events, such as the identification of new requirements or the detection of incidents within the virtual system, might urge a VNO to quickly reform its virtual infrastructure and resource allocation. However, the interfaces offered by current virtualization platforms do not offer the necessary functions for VNOs to perform the elastic adaptations they need to conduct in dynamic environments.

Beyond their heterogeneity, which can be resolved by software adapters, current virtualization platforms do not have common methods and functions, so it is difficult for the virtual network controllers used by the VNOs to actually manage and control virtual resources instantiated on different platforms, not even considering different InPs. Therefore, it is necessary to reach a common definition of the functions that should be offered by underlying platforms to give such overlay controllers the possibility to allocate and deallocate resources dynamically and get monitoring data about them.

Such common methods should be offered by all underlying controllers, regardless of being network-oriented (e.g., ODL, ONOS, and Ryu) or computing-oriented (e.g., OpenStack, OpenNebula, and Eucalyptus). Furthermore, it is important for those platforms to offer some "PUSH" function to report resource state, avoiding the need for the VNO's controller to "POLL" for such data. A starting point to get proper notifications within current REST APIs could be to consider the protocol proposed by the WEBPUSH WG [RFC8030].

Finally, in order to establish a proper order and allow the coexistence and collaboration of different systems, a common ontology regarding network and system virtualization should be defined and agreed upon, so different and heterogeneous systems can understand each other without requiring reliance on specific adaptation mechanisms that might break with any update on any side of the relation.

4.5.2. Extending Virtual Networks and Systems to the Internet of Things

The Internet of Things (IoT) refers to the vision of connecting a multitude of automated devices (e.g., lights, environmental sensors, traffic lights, parking meters, health and security systems, etc.) to the Internet for purposes of reporting and remote command and control of the device. This vision is being realized by a multi-pronged approach of standardization in various forums and complementary open-source activities. For example, in the IETF, support of IoT web services has been defined by an HTTP-like protocol adapted for IoT called "CoAP" [RFC7252]; and, lately, a group has been studying the need to develop a new network layer to support IP applications over Low-Power Wide Area Networks (LPWAN).

Elsewhere, for 5G cellular evolution, there is much discussion on the need for supporting virtual network slices for the expected massive numbers of IoT devices. A separate virtual network slice is considered necessary for different 5G IoT use cases because devices will have very different characteristics than typical cellular

devices like smartphones [ngmn_5G_whitepaper], and the number of IoT devices is expected to be at least one or two orders of magnitude higher than other 5G devices (see Section 4.5).

The specific nature of the IoT ecosystem, particularly reflected in the Machine-to-Machine (M2M) communications, leads to the creation of new and highly distributed systems which demand location-based network and computing services. A specific example can be represented by a set of "things" that suddenly require the setup of a firewall to allow external entities to access their data while outsourcing some computation requirements to more powerful systems relying on cloud-based services. This representative use case exposes important requirements for both NFV and the underlying cloud infrastructures.

In order to provide the aforementioned location-based functions integrated with highly distributed systems, the so-called fog infrastructures should be able to instantiate VNFs, placing them in the required place, e.g., close to their consumers. This requirement implies that the interfaces offered by virtualization platforms must support the specification of location-based resources, which is a key function in those scenarios. Moreover, those platforms must also be able to interpret and understand the references used by IoT systems to their location (e.g., "My-AP" or "5BLDG+2F") and also the specification of identifiers linked to other resources, such as the case of requiring the infrastructure to establish a link between a specific Access Point (AP) and a specific virtual computing node. In summary, the research gap is exact localization of VNFs at far network edge infrastructure, which is highly distributed and dynamic.

4.6. Service Composition

Current network services deployed by operators often involve the composition of several individual functions (such as packet filtering, deep-packet inspection, load-balancing). These services are typically implemented by the ordered combination of a number of service functions that are deployed at different points within a network, not necessarily on the direct data path. This requires traffic to be steered through the required service functions, wherever they are deployed [RFC7498].

For a given service, the abstracted view of the required service functions and the order in which they are to be applied is called "Service Function Chaining" (SFC) [sfc_challenges], which is called "Network Function Forwarding Graph" (NF-FG) in ETSI. SFC is instantiated through the 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.).

Service composition is a powerful means that can provide significant benefits when applied in a softwarized network environment. However, there are many research challenges in this area; for example, the ones related to composition mechanisms and algorithms to enable load-balancing and improve reliability. The service composition should also act as an enabler to gather information across all hierarchies (underlays and overlays) of network deployments that may span across multiple operators for faster serviceability, thus facilitating accomplishing aforementioned goals of "load-balancing and improving reliability".

As described in [dynamic_chaining], different algorithms can be used to enable dynamic service composition that optimizes a QoS-based utility function (e.g., minimizing the latency per-application traffic flows) for a given composition plan. Such algorithms can consider the computation capabilities and load status of resources executing the VNF instances, either deduced through estimations from historical usage data or collected through real-time monitoring (i.e., context-aware selection). For this reason, selections should include references to dynamic information on the status of the service instance and its constituent elements, i.e., monitoring information related to individual VNF instances and links connecting them as well as derived monitoring information at the chain level (e.g., end-to-end delay). At runtime, if one or more VNF instances are no longer available or QoS degrades below a given threshold, the service selection task can be rerun to perform service substitution.

There are different research directions that relate to the previous point. For example, the use of Integer Linear Programming (ILP) techniques can be explored to optimize the management of diverse traffic flows. Deep-machine learning can also be applied to optimize service chains using information parameters, such as some of the ones mentioned above. Newer scheduling paradigms, like co-flows, can also be used.

The SFC working group is working on an architecture for SFC [RFC7665] that includes the necessary protocols or protocol extensions to convey the SFC and SFP information to nodes that are involved in the implementation of service functions and SFCs as well as mechanisms for steering traffic through service functions.

In terms of actual work items, the SFC WG has not yet considered working on the management and configuration of SFC components related to the support of SFC. This part is of special interest for

operators and would be required in order to actually put SFC mechanisms into operation. Similarly, redundancy and reliability mechanisms for SFC are currently not dealt with by any WG in the IETF. While this was the main goal of the VNFpool BoF efforts, it still remains unaddressed.

4.7. Device Virtualization for End Users

So far, most of the network softwarization efforts have focused on virtualizing functions of network elements. While virtualization of network elements started with the core, mobile-network architectures are now heavily switching to also virtualize Radio Access Network (RAN) functions. The next natural step is to get virtualization down at the level of the end-user device (e.g., virtualizing a smartphone) [virtualization_mobile_device]. The cloning of a device in the cloud (central or local) bears attractive benefits to both the device and network operations alike (e.g., power saving at the device by offloading computational-heaving functions to the cloud, optimized networking -- both device-to-device and device-to-infrastructure) for service delivery through tighter integration of the device (via its clone in the networking infrastructure). This is, for example, being explored by the European H2020 ICIRRUS project <https://www.icirrus-5gnet.eu>.

4.8. Security and Privacy

Similar to any other situations where resources are shared, security and privacy are two important aspects that need to be taken into account.

In the case of security, there are situations where multiple service providers will need to coexist in a virtual or hybrid physical/virtual environment. This requires attestation procedures amongst different virtual/physical functions and resources as well as ongoing external monitoring. Similarly, different network slices operating on the same infrastructure can present security problems, for instance, if one slice running critical applications (e.g., support for a safety system) is affected by another slice running a less critical application. In general, the minimum common denominator for security measures on a shared system should be equal to or higher than the one required by the most-critical application. Multiple and continuous threat model analysis as well as a DevOps model are required to maintain a certain level of security in an NFV system. Simplistically, DevOps is a process that combines multiple functions into single cohesive teams in order to quickly produce quality software. Typically, it relies on also applying the Agile development process, which focuses on (among many things) dividing large features into multiple, smaller deliveries. One part of this

is to immediately test the new smaller features in order to get immediate feedback on errors so that if present, they can be immediately fixed and redeployed.

On the other hand, privacy refers to concerns about the control of personal data and the decision of what to reveal to whom. In this case, the storage, transmission, collection, and potential correlation of information in the NFV system, for purposes not originally intended or not known by the user, should be avoided. This is particularly challenging, as future intentions and threats cannot be easily predicted and still can be applied on data collected in the past. Therefore, well-known techniques, such as data minimization using privacy features as default and allowing users to opt in/out, should be used to prevent potential privacy issues.

Compared to traditional networks, NFV will result in networks that are much more dynamic (in function distribution and topology) and elastic (in size and boundaries). Thus, NFV will require network operators to evolve their operational and administrative security solutions to work in this new environment. For example, in NFV, the network orchestrator will become a key node to provide security policy orchestration across the different physical and virtual components of the virtualized network. For highly confidential data, for example, the network orchestrator should take into account if certain physical HW of the network is considered to be more secure (e.g., because it is located in secure premises) than other HW.

Traditional telecom networks typically run under a single administrative domain controlled by (exactly) one operator. With NFV, it is expected that in many cases, the telecom operator will now become a tenant (running the VNFs), and the infrastructure (NFVI) may be run by a different operator and/or cloud service provider (see also Section 4.4). Thus, there will be multiple administrative domains involved, making security policy coordination more complex. For example, who will be in charge of provisioning and maintaining security credentials such as public and private keys? Also, should private keys be allowed to be replicated across the NFV for redundancy reasons? Alternatively, it can be investigated how to develop a mechanism that avoids such a security policy coordination, thus making the system more robust.

On a positive note, NFV may better defend against denial-of-service (DoS) attacks because of the distributed nature of the network (i.e., no single point of failure) and the ability to steer (undesirable) traffic quickly [etsi_gs_nfv_sec_001]. Also, NFVs that have physical HW that is distributed across multiple data centers will also provide

better fault isolation environments. Particularly, this holds true if each data center is protected separately via firewalls, Demilitarized Zones (DMZs), and other network-protection techniques.

SDN can also be used to help improve security by facilitating the operation of existing protocols, such as Authentication, Authorization and Accounting (AAA). The management of AAA infrastructures, namely the management of AAA routing and the establishment of security associations between AAA entities, can be performed using SDN, as analyzed in [SDN-AAA].

4.9. Separation of Control Concerns

NFV environments offer two possible levels of SDN control. One level is the need for controlling the NFVI to provide connectivity end-to-end among VNFs or among VNFs and Physical Network Functions (PNFs). A second level is the control and configuration of the VNFs themselves (in other words, the configuration of the network service implemented by those VNFs), taking advantage of the programmability brought by SDN. Both control concerns are separated in nature. However, interaction between both could be expected in order to optimize, scale, or influence each other.

Clear mechanisms for such interactions are needed in order to avoid malfunctioning or interference concerns. These ideas are considered in [etsi_gs_nfv_eve005] and [LAYERED-SDN].

4.10. Network Function Placement

Network function placement is a problem in any kind of network telecommunications infrastructure. Moreover, the increased degree of freedom added by network virtualization makes this problem even more important, and also harder to tackle. Deciding where to place VNFs is a resource-allocation problem that needs to (or may) take into consideration quite a few aspects: resiliency, (anti-)affinity, security, privacy, energy efficiency, etc.

When several functions are chained (typical scenario), placement algorithms become more complex and important (as described in Section 4.6). While there has been research on the topic ([nfv_piecing], [dynamic_placement], and [vnf-p]), this still remains an open challenge that requires more attention. The use of multi-domains adds another component of complexity to this problem that has to be considered.

4.11. Testing

The impacts of network virtualization on testing can be divided into three groups:

1. Changes in methodology
2. New functionality
3. Opportunities

4.11.1. Changes in Methodology

The largest impact of NFV is the ability to isolate the System Under Test (SUT). When testing PNFs, isolating the SUT means that all the other devices that the SUT communicates with are replaced with simulations (or controlled executions) in order to place the SUT under test by itself. The SUT may be comprised of one or more devices. The simulations use the appropriate traffic type and protocols in order to execute test cases.

As shown in Figure 2, NFV provides a common architecture for all functions to use. A VNF is executed using resources offered by the NFVI, which have been allocated using the MANO function. It is not possible to test a VNF by itself, without the entire supporting environment present. This fundamentally changes how to consider the SUT. In the case of a VNF (or multiple VNFs), the SUT is part of a larger architecture that is necessary in order to run the SUTs.

Therefore, isolation of the SUT becomes controlling the environment in a disciplined manner. The components of the environment necessary to run the SUTs that are not part of the SUT itself become the test environment. In the case of VNFs that are part of the SUT, the NFVI and MANO become the test environment. The configurations and policies that guide the test environment should remain constant during the execution of the tests, and also from test to test. Configurations such as CPU pinning, NUMA configuration, the SW versions and configurations of the hypervisor, vSwitch and NICs should remain constant. The only variables in the testing should be those controlling the SUT itself. If any configuration in the test environment is changed from test to test, the results become very difficult, if not impossible, to compare since the test environment behavior may change the results as a consequence of the configuration change.

Testing the NFVI itself also presents new considerations. With a PNF, the dedicated hardware supporting it is optimized for the particular workload of the function. Routing hardware is specially

built to support packet forwarding functions, while the hardware to support a purely control-plane application (say, a DNS server, or a Diameter function) will not have this specialized capability. In NFV, the NFVI is required to support all types of potentially different workload types.

Therefore, testing the NFVI requires careful consideration about what types of metrics are sought. This, in turn, depends on the workload type the expected VNF will be. Examples of different workload types are data forwarding, control plane, encryption, and authentication. All these types of expected workloads will determine the types of metrics that should be sought. For example, if the workload is control plane, then a metric such as jitter is not useful, but dropped packets are critical. In a multi-tenant environment, the NFVI could support various types of workloads. In this case, testing with a variety of traffic types while measuring the corresponding metrics simultaneously becomes necessary.

Test beds for any type of testing for an NFV-based system will be largely similar to previously used test architectures. The methods are impacted by virtualization, as described above, but the design of test beds are similar as in the past. There are two main new considerations:

- Since networking is based on software, which has lead to greater automation in deployment, the test system should also be deployable with the rest of the system in order to fully automate the system. This is especially relevant in a DevOps environment supported by a Continuous Integration and Continuous Deployment (CI/CD) tool chain (see Section 4.11.3 below).

- In any performance test bed, the test system should not share the same resources as the SUT. While multi-tenancy is a reality in virtualization, having the test system share resources with the SUT will impact the measured results in a performance test bed. The test system should be deployed on a separate platform in order not to impact the resources available to the SUT.

4.11.2. New Functionality

NFV presents a collection of new functionality in order to support the goal of software networking. Each component on the architecture shown in Figure 2 has an associated set of functionality that allows VNFs to run the following: onboarding, life-cycle management for VNFs and Network Services (NS), resource allocation, hypervisor functions, etc.

One of the new capabilities enabled by NFV is VNF Forwarding Graphs (VNFFG). This refers to the graph that represents a network service by chaining together VNFs into a forwarding path. In practice, the forwarding path can be implemented in a variety of ways using different networking capabilities: vSwitch, SDN, and SDN with a northbound application. Additionally, the VNFFG might use tunneling protocols like Virtual eXtensible Local Area Network (VXLAN). The dynamic allocation and implementation of these networking paths will have different performance characteristics depending on the methods used. The path implementation mechanism becomes a variable in the network testing of the NSs. The methodology used to test the various mechanisms should largely remain the same; as usual, the test environment should remain constant for each of the tests, focusing on varying the path establishment method.

"Scaling" refers to the change in allocation of resources to a VNF or NS. It happens dynamically at run-time, based on defined policies and triggers. The triggers can be network, compute, or storage based. Scaling can allocate more resources in times of need, or reduce the amount of resources allocated when the demand is reduced. The SUT in this case becomes much larger than the VNF itself: MANO controls how scaling is done based on policies, and then allocates the resources accordingly in the NFVI. Essentially, the testing of scaling includes the entire NFV architecture components into the SUT.

4.11.3. Opportunities

Softwarization of networking functionality leads to softwarization of the test as well. As PNFs are being transformed into VNFs, so are the test tools. This leads to the fact that test tools are also being controlled and executed in the same environment as the VNFs. This presents an opportunity to include VNF-based test tools along with the deployment of the VNFs supporting the services of the service provider into the host data centers. Therefore, tests can be automatically executed upon deployment in the target environment, for each deployment, and each service. With PNFs, this was very difficult to achieve.

This new concept helps to enable modern concepts like DevOps and Continuous Integration and Continuous Deployment in the NFV environment. The CI/CD pipeline supports this concept. It consists of a series of tools, among which immediate testing is an integral part, to deliver software from source to deployment. The ability to deploy the test tools themselves into the production environment stretches the CI/CD pipeline all the way to production deployment, allowing a range of tests to be executed. The tests can be simple,

with a goal of verifying the correct deployment and networking establishment, but can also be more complex, like testing VNF functionality.

5. Technology Gaps and Potential IETF Efforts

Table 1 correlates the open network virtualization research areas identified in this document to potential IETF and IRTF groups that could address some aspects of them. An example of a specific gap that the group could potentially address is identified as a parenthetical beside the group name.

Open Research Area	Potential IETF/IRTF Group
1) Guaranteeing QoS	IPPM WG (Measurements of NFVI)
2) Performance improvement	SFC WG, NFVRG (energy-driven orchestration)
3) Multiple Domains	NFVRG (multi-domain orchestration)
4) Network Slicing	NVO3 WG, NETSLICES bar BoF (multi-tenancy support)
5) Service Composition	SFC WG (SFC Mgmt and Config)
6) End-user device virtualization	N/A
7) Security	N/A
8) Separation of control concerns	NFVRG (separation between transport control and services)
9) Testing	NFVRG (testing of scaling)
10) Function placement	NFVRG, SFC WG (VNF placement algorithms and protocols)

Table 1: Mapping of Open Research Areas to Potential IETF Groups

6. NFVRG Focus Areas

Table 2 correlates the currently identified NFVRG topics of interest / focus areas to the open network virtualization research areas enumerated in this document. This can help the NFVRG in identifying and prioritizing research topics. The current list of NFVRG focus points is the following:

- Re-architecting functions, including aspects such as new architectural and design patterns (e.g., containerization, statelessness, serverless, control/data plane separation), SDN integration, and proposals on programmability.
- New management frameworks, considering aspects related to new OAM mechanisms (e.g., configuration control, hybrid descriptors) and lightweight MANO proposals.
- Techniques to guarantee low latency, resource isolation, and other data-plane features, including hardware acceleration, functional offloading to data-plane elements (including NICs), and related approaches.
- Measurement and benchmarking, addressing both internal measurements and external applications.

NFVRG Focus Point	Open Research Area
1) Re-architecting functions	- Performance improvem. - Network Slicing - Guaranteeing QoS - Security - End-user device virt. - Separation of control
2) New management frameworks	- Multiple Domains - Service Composition - End-user device virt.
3) Low latency, resource isolation, etc.	- Performance improvem. - Separation of control
4) Measurement and benchmarking	- Guaranteeing QoS - Testing

Table 2: Mapping of NFVRG Focus Points to Open Research Areas

7. IANA Considerations

This document has no IANA actions.

8. Security Considerations

This is an Informational RFC that details research challenges; it does not introduce any security threat. Research challenges and gaps related to security and privacy have been included in Section 4.8.

9. Informative References

[COMS-PS] Geng, L., Slawomir, S., Qiang, L., Matsushima, S., Galis, A., and L. Contreras, "Problem Statement of Common Operation and Management of Network Slicing", Work in Progress, draft-geng-coms-problem-statement-04, March 2018.

[dynamic_chaining] Martini, B. and F. Paganelli, "A Service-Oriented Approach for Dynamic Chaining of Virtual Network Functions over Multi-Provider Software-Defined Networks", Future Internet Vol. 8, No. 2, DOI 10.3390/fi8020024, June 2016.

[dynamic_placement] Clayman, S., Maini, E., Galis, A., Manzalini, A., and N. Mazzocca, "The dynamic placement of virtual network functions", 2014 IEEE Network Operations and Management Symposium (NOMS) pp. 1-9, DOI 10.1109/NOMS.2014.6838412, May 2014.

[etsi_gs_nfv_003] ETSI NFV ISG, "Network Functions Virtualisation (NFV); Terminology for Main Concepts in NFV", ETSI GS NFV 003 V1.2.1 NFV 003, December 2014, <http://www.etsi.org/deliver/etsi_gs/NFV/001_099/003/01.02.01_60/gs_NFV003v010201p.pdf>.

[etsi_gs_nfv_eve005] ETSI NFV ISG, "Network Functions Virtualisation (NFV); Ecosystem; Report on SDN Usage in NFV Architectural Framework", ETSI GS NFV-EVE 005 V1.1.1 NFV-EVE 005, December 2015, <http://www.etsi.org/deliver/etsi_gs/NFV-EVE/001_099/005/01.01.01_60/gs_NFV-EVE005v010101p.pdf>.

[etsi_gs_nfv_per_001]
ETSI NFV ISG, "Network Functions Virtualisation (NFV); NFV Performance & Portability Best Practises", ETSI GS NFV-PER 001 V1.1.2 NFV-PER 001, December 2014, <https://www.etsi.org/deliver/etsi_gs/nfv-per/001_099/001/01.01.02_60/gs_nfv-per001v010102p.pdf>.

[etsi_gs_nfv_sec_001]
ETSI NFV ISG, "Network Functions Virtualisation (NFV); NFV Security; Problem Statement", ETSI GS NFV-SEC 001 V1.1.1 NFV-SEC 001, October 2014, <http://www.etsi.org/deliver/etsi_gs/NFV-SEC/001_099/001/01.01.01_60/gs_NFV-SEC001v010101p.pdf>.

[etsi_nfv_whitepaper_3]
ETSI, "Network Functions Virtualisation (NFV) - White Paper #3: Network Operator Perspectives on Industry Progress", Issue 1, SDN & OpenFlow World Congress Dusseldorf, Germany, October 2014, <http://portal.etsi.org/NFV/NFV_White_Paper3.pdf>.

[google_sdn_wan]
Jain, S., et al., "B4: experience with a globally-deployed Software Defined WAN", SIGCOMM '13: Proceedings of the ACM SIGCOMM 2013 conference on SIGCOMM, pp. 3-14, Hong Kong China, DOI 10.1145/2486001.2486019, August 2013.

[intel_10_differences_nfv_cloud]
Torre, P., "Discover the Top 10 Differences Between NFV and Cloud Environments", November 2015, <https://software.intel.com/en-us/videos/discover-the-top-10-differences-between-nfv-and-cloud-environments>.

[itu-t-y.3300]
ITU-T, "Y.3300: Framework of software-defined networking", ITU-T Recommendation Y.3300, June 2014, <http://www.itu.int/rec/T-REC-Y.3300-201406-I/en>.

[itu-t-y.3301]
ITU-T, "Y.3301: Functional requirements of software-defined networking", ITU-T Recommendation Y.3301, September 2016, <http://www.itu.int/rec/T-REC-Y.3301-201609-I/en>.

[itu-t-y.3302]
ITU-T, "Y.3302: Functional architecture of software-defined networking", ITU-T Recommendation Y.3302, January 2017, <http://www.itu.int/rec/T-REC-Y.3302-201701-I/en>.

[LAYERED-SDN]
Contreras, L., Bernardos, C., Lopez, D., Boucadair, M., and P. Iovanna, "Cooperating Layered Architecture for Software Defined Networking (CLAS)", Work in Progress, draft-contreras-layered-sdn-03, November 2018.

[LIGHT-NFV]
Sriram, N., Krishnan, R., Ghanwani, A., Krishnaswamy, D., Willis, P., Chaudhary, A., and F. Huici, "An Analysis of Lightweight Virtualization Technologies for NFV", Work in Progress, draft-natarajan-nfvrg-containers-for-nfv-03, July 2016.

[multi-domain_5GEx]
Bernardos, C., Gero, B., Di Girolamo, M., Kern, A., Martini, B., and I. Vaishnavi, "5GEx: Realizing a Europe-wide Multi-domain framework for software-defined infrastructures", Transactions on Emerging Telecommunications Technologies Vol. 27, No. 9, pp. 1271-1280, DOI 10.1002/ett.3085, July 2016.

[MULTI-NMRG]
Bernardos, C., Contreras, L., Vaishnavi, I., Szabo, R., Li, X., Paolucci, F., Sgambelluri, A., Martini, B., Valcarenghi, L., Landi, G., Andrushko, D., and A. Mourad, "Multi-domain Network Virtualization", Work in Progress, draft-bernardos-nmrg-multidomain-00, March 2019.

[NETSLICES]
Galis, A., Dong, J., Makhijani, K., Bryant, S., Boucadair, M., and P. Martinez-Julia, "Network Slicing - Introductory Document and Revised Problem Statement", Work in Progress, draft-gdmb-netslices-intro-and-ps-02, February 2017.

[NFV-COTS] Mo, L. and B. Khasnabish, "NFV Reliability using COTS Hardware", Work in Progress, draft-mlk-nfvrg-nfv-reliability-using-cots-01, October 2015.

[nfv_piecing]
Luizelli, M., Bays, L., Buriol, L., Barcellos, M., and L. Gaspary, "Piecing together the NFV provisioning puzzle: Efficient placement and chaining of virtual network functions", 2015 IFIP/IEEE International Symposium on Integrated Network Management (IM) pp. 98-106, DOI 10.1109/INM.2015.7140281, May 2015.

[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, pp. 236-262, DOI 10.1109/COMST.2015.2477041, September 2015.

[NFVRG-TOPO]
Bagnulo, M. and D. Dolson, "NFVI PoP Network Topology: Problem Statement", Work in Progress, draft-bagnulo-nfvrg-topology-01, March 2016.

[ngmn_5G_whitepaper]
NGMN Alliance, "NGMN 5G White Paper", Version 1.0, February 2015, <https://www.ngmn.org/fileadmin/ngmn/content/images/news/ngmn_news/NGMN_5G_White_Paper_V1_0.pdf>.

[omniran] IEEE, "Recommended Practice for Network Reference Model and Functional Description of IEEE 802 Access Network", P802.1CF IEEE Draft, December 2017.

[onf_tr_521]
Open Networking Foundation, "SDN Architecture", ONF TR-521 TR-521, Issue 1.1, February 2016, <https://www.opennetworking.org/images/stories/downloads/sdn-resources/technical-reports/TR-521_SDN_Architecture_issue_1.1.pdf>.

[OpenFlow] Open Networking Foundation, "OpenFlow Switch Specification", ONF TS-025, Version 1.5.1 (Protocol version 0x06), March 2015.

[openmano_dataplane]
Lopez, D., "OpenMANO: The Dataplane Ready Open Source NFV MANO Stack", March 2015, <https://www.ietf.org/proceedings/92/slides/slides-92-nfvrg-7.pdf>.

[RFC5810] Doria, A., Ed., Hadi Salim, J., Ed., Haas, R., Ed., Khosravi, H., Ed., Wang, W., Ed., Dong, L., Gopal, R., and J. Halpern, "Forwarding and Control Element Separation (ForCES) Protocol Specification", RFC 5810, DOI 10.17487/RFC5810, March 2010, <https://www.rfc-editor.org/info/rfc5810>.

[RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <https://www.rfc-editor.org/info/rfc6241>.

[RFC7252] Shelby, Z., Hartke, K., and C. Bormann, "The Constrained Application Protocol (CoAP)", RFC 7252, DOI 10.17487/RFC7252, June 2014, <https://www.rfc-editor.org/info/rfc7252>.

[RFC7426] Haleplidis, E., Ed., Pentikousis, K., Ed., Denazis, S., Hadi Salim, J., Meyer, D., and O. Koufopavlou, "Software-Defined Networking (SDN): Layers and Architecture Terminology", RFC 7426, DOI 10.17487/RFC7426, January 2015, <https://www.rfc-editor.org/info/rfc7426>.

[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>.

[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>.

[RFC8030] Thomson, M., Damaggio, E., and B. Raymor, Ed., "Generic Event Delivery Using HTTP Push", RFC 8030, DOI 10.17487/RFC8030, December 2016, <https://www.rfc-editor.org/info/rfc8030>.

[RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017, <https://www.rfc-editor.org/info/rfc8040>.

[RFC8172] Morton, A., "Considerations for Benchmarking Virtual Network Functions and Their Infrastructure", RFC 8172, DOI 10.17487/RFC8172, July 2017, <https://www.rfc-editor.org/info/rfc8172>.

[SDN-AAA] Lopez, R. and G. Lopez-Millan, "Software-Defined Networking (SDN)-based AAA Infrastructures Management", Work in Progress, draft-marin-sdnrg-sdn-aaa-mng-00, November 2015.

[sfc_challenges] Medhat, A., Taleb, T., Elmangoush, A., Carella, G., Covaci, S., and T. Magedanz, "Service Function Chaining in Next Generation Networks: State of the Art and Research Challenges", IEEE Communications Magazine vol. 55, no. 2, pp. 216-223, DOI 10.1109/MCOM.2016.1600219RP, February 2017.

[SLICE-3GPP] Foy, X. and A. Rahman, "Network Slicing - 3GPP Use Case", Work in Prgoress, draft-defoy-netslices-3gpp-network-slicing-02, October 2017.

[virtualization_mobile_device] Sproule, W. and A. Fernando, "Virtualization of Mobile Device User Experience", US Patent 9.542.062 B2, filed October 2013 and issued December 2014, Current Assignee: Microsoft Technology Licensing LLC.

[vnf-p] Moens, H. and , "VNF-P: A model for efficient placement of virtualized network functions", 10th International Conference on Network and Service Management (CNSM) and Workshop pp. 418-423, DOI 10.1109/CNSM.2014.7014205, November 2014.

[VNF-VBAAS] Rosa, R., Rothenberg, C., and R. Szabo, "VNF Benchmark-as-a-Service", Work in Progress, draft-rorosz-nfvrg-vbaas-00, October 2015.

[vnf_benchmarking] Rosa, R., Rothenberg, C., and R. Szabo, "A VNF Testing Framework Design, Implementation and Partial Results", NFVRG IETF 97, November 2016, <https://www.ietf.org/proceedings/97/slides/slides-97-nfvrg-06-vnf-benchmarking-00.pdf>.

Acknowledgments

The authors want to thank Dirk von Hugo, Rafa Marin, Diego Lopez, Ramki Krishnan, Kostas Pentikousis, Rana Pratap Sircar, Alfred Morton, Nicolas Kuhn, Saumya Dikshit, Fabio Giust, Evangelos Haleplidis, Angeles Vazquez-Castro, Barbara Martini, Jose Saldana, and Gino Carrozzo for their very useful reviews and comments to the document. Special thanks to Pedro Martinez-Julia, who provided text for the network slicing section.

The authors want to also thank Dave Oran and Michael Welzl for their very detailed IRSG reviews.

The work of Carlos J. Bernardos and Luis M. Contreras is partially supported by the H2020 5GEx (Grant Agreement no. 671636) and 5G-TRANSFORMER (Grant Agreement no. 761536) projects.

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/

Akbar Rahman
InterDigital Communications, LLC
1000 Sherbrooke Street West, 10th floor
Montreal, Quebec H3A 3G4
Canada

Email: Akbar.Rahman@InterDigital.com
URI: http://www.InterDigital.com/

Juan Carlos Zuniga
SIGFOX
425 rue Jean Rostand
Labege 31670
France

Email: j.c.zuniga@ieee.org
URI: http://www.sigfox.com/

Luis M. Contreras
Telefonica I+D
Ronda de la Comunicacion, S/N
Madrid 28050
Spain

Email: luismiguel.contrerasmurillo@telefonica.com

Pedro Aranda
Universidad Carlos III de Madrid
Av. Universidad, 30
Leganes, Madrid 28911
Spain

Email: pedroandres.aranda@uc3m.es

Pierre Lynch
Keysight Technologies
800 Perimeter Park Dr, Suite A
Morrisville, NC 27560
United States of America

Email: pierre.lynch@keysight.com
URI: http://www.keysight.com