Neves, Pedro and Calé, Rui and Costa, Mário Rui and Parada, Carlos and Parreira, Bruno and Alcaraz-Calero, Jose and Wang, Qi and Nightingale, James and Chirivella-Perez, Enrique and Jiang, Wei and Schotten, Hans Dieter and Koutsopoulos, Konstantinos and Gavras, Anastasius and Barros, Maria João (2016) The SELFNET approach for autonomic management in an NFV/SDN networking paradigm. International Journal of Distributed Sensor Networks, 12 (2). ISSN 1550-1329 , http://dx.doi.org/10.1155/2016/2897479

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2016, Article ID 2897479, 17 pages
http://dx.doi.org/10.1155/2016/2897479

Research Article

The SELFNET Approach for Autonomic Management in an NFV/SDN Networking Paradigm

Pedro Neves,[1] Rui Calé,[1] Mário Rui Costa,[1] Carlos Parada,[1] Bruno Parreira,[1] Jose Alcaraz-Calero,[2] Qi Wang,[2] James Nightingale,[2] Enrique Chirivella-Perez,[2] Wei Jiang,[3] Hans Dieter Schotten,[3] Konstantinos Koutsopoulos,[4] Anastasius Gavras,[5] and Maria João Barros[5]

[1] Portugal Telecom Inovação (PTIN), Rua Eng. José Ferreira Pinto Basto, 3810-106 Aveiro, Portugal
[2] University of West Scotland (UWS), Almada Street, Hamilton, South Lanarkshire ML3 0JB, UK
[3] German Research Centre for Artificial Intelligence (DFKI GmbH), 67655 Kaiserslautern, Germany
[4] Creative Systems Engineering (CSE), Agiou Meletiou 45, 11257 Athens, Greece
[5] Eurescom, Wieblinger Weg 19, 69123 Heidelberg, Germany

Correspondence should be addressed to Pedro Neves; pedro-m-neves@telecom.pt

Received 4 December 2015; Accepted 24 December 2015

Academic Editor: Luis Javier Garcia Villalba

To meet the challenging key performance indicators of the fifth generation (5G) system, the network infrastructure becomes more heterogeneous and complex. This will bring a high pressure on the reduction of OPEX and the improvement of the user experience. Hence, shifting today's manual and semi-automatic network management into an autonomic and intelligent framework will play a vital role in the upcoming 5G system. Based on the cutting-edge technologies, such as Software-Defined Networking and Network Function Virtualization, a novel management framework upon the software-defined and Virtualized Network is proposed by EU H2020 SELFNET project. In the paper, the reference architecture of SELFNET, which is divided into Infrastructure Layer, Virtualized Network Layer, SON Control Layer, SON Autonomic Layer, NFV Orchestration and Management Layer, and Access Layer, will be presented.

1. Introduction

As of today, the proliferation of smart phones and tablet computers impose an exponential growth of mobile traffic, which is estimated to increase 1000 times in comparison with that of 2010 [1, 2]. Besides, the emergence of new devices and services, such as wearable electronics, self-driving automobiles, virtual and augmented reality, 3D video, and Machine-to-Machine communications, will bring more challenging and far stricter key performance indicators (KPIs) on huge capacity, massive-connection provision, ultra-low latency, and ultra-reliability. Definitely, such KPIs are impossible to be satisfied by the current 4G cellular systems, which are designed for mobile voice and ordinary Internet access. This motivates the research and development activities of the fifth generation (5G) communication system. To meet these requirements, the following evolutions are envisioned: (1) the system architecture transforming from homogeneity to heterogeneity, which consists of Marco-Cell, Small Cell, Moving Cell, Relaying, and Device-to-Device links; (2) new paradigms of spectrum utilization, such as dynamic spectrum access and cognitive radio, Licensed-Share Access, License-Assisted Access, and higher frequency at millimeter wave spectrum bands, which should be applied, rather than the legacy statically spectral allocation; (3) cutting-edge transmission technologies, such as massive and distributed MIMO, Coordinated Multiple Point (CoMP), and Filter-Bank Multicarrier (FBMC), which are potentially to be used; (4) not only the new Radio Access Technology which will be defined but also the legacy 3G/4G, Wi-Fi, and even Fixed-Mobile Convergence (FMC) which should be covered by the 5G system, which is the distinct feature.

In a nutshell, the 5G system will become extreme complex in order to meet the user requirements and the challenging KPIs. However, this complexity will suffer from difficult, costly, and time-consuming network management under today's manual or semi-automatic approaches. Currently, in financial terms, the cost of mobile operators on operational expenditure (OPEX) is three times that of capital expenditure (CAPEX) [3]. In the 5G system, it can be foreseen that the reduction of OPEX will become more challenging, but at the same time more critical. On the other hand, the manual and semi-automatic management approaches are hard to make full use of the resources within 5G infrastructures to provide the required quality-of-experience (QoE) in terms of service availability, continuity, security, and reliability.

In this context, the EU H2020 SELFNET project [4] will design and implement a highly autonomic management framework to provide network intelligence and self-organizing capability for 5G mobile network infrastructures. By reactive and (more importantly) proactive detecting and mitigating a range of common network problems, currently manually addressed by network administrators, SELFNET can significantly reduce OPEX and substantially improve the user experience.

By exploring the integration of cutting-edge technologies, such as Software-Defined Networking (SDN), Network Function Virtualization (NFV), Self-Organizing Network (SON), Cloud Computing, and Artificial Intelligence, SELFNET can provide a scalable, extensible, and smart network management system. The framework will assist network operators to simplify the key management tasks and save the man power. For example, SDN/NFV sensors that can monitor the network and SDN/NFV actuators that can perform corrective and preventive actions to mitigate existing or potential network problems can be automatically deployed. SELFNET aims to address three major network management concerns. That is, providing self-protection capabilities against distributed cyber-attacks, self-healing capabilities against network failures, and self-optimization features to dynamically improve the performance of the network and the QoE of the end users. The facilities provided by SELFNET will provide the foundations for delivering some of the 5G requirements defined by the 5G-PPP initiative [5].

In this paper, the system architecture proposed by SELFNET project and its associated issues will be presented. Section 2 gives the background of SELFNET related to other existing work and initiatives, in particular from the perspective of 5G PPP. Section 3 provides the standard foundations of SELFNET's architecture. In Section 4, the reference architecture is depicted in detail, which is divided into 6 parts: Infrastructure Layer, Virtualized Network Layer, SON Control Layer, SON Autonomic Layer, NFV Orchestration and Management Layer, and Access Layer. Finally, Section 5 concludes this paper.

2. Related Work

The SELFNET project will build upon cutting-edge technologies in software networking, especially Software-Defined Networks (SDN) and Network Function Virtualization (NFV) research within the EU and globally, and further develop existing and emerging SDN/NFV facilities and services for 5G mobile networks. Research in recent years in the area of SDN has resulted in a paradigm shift in the way network operations are planned and deployed. SDN separates the data and control planes in network devices and establishes a centralized control of the network. SDN is complemented by NFV, which allows the deployment of Virtualized Network Functions (e.g., load balancers and firewalls) as virtual instances over commodity hardware, thereby substantially reducing capital and operational costs.

One of the main strands of recent SDN research has been to extend its application to wireless and mobile environments. Proposals in this area have encompassed Software-Defined Wireless Networks [6, 7], Wireless Mesh Software-Defined Networks [8], and mobile technologies, particularly Long Term Evolution (LTE) [9, 10]. Naudts et al. [11] highlighted the technoeconomic case for the transition to an SDN based mobile network infrastructure supported by claimed capital expenditure savings, compared to traditional infrastructures, ranging from almost 14% with SDN alone and up to 58% when a virtualized shared SDN is considered. Some examples of proposals for SDN integration with mobile networks include the deployment of radio access base stations [12] and LTE network components [13] into cloud environments.

Separation of Control and Data Planes in SDN has enabled the development of NFV [14], which supports the deployment of Virtualized Network Functions (VNFs). These network functions are completely designed in software and operate on generic hardware, reducing reliance on expensive proprietary hardware. This generic hardware can be abstracted to provide virtual computation, storage, or other virtual network resources while providing higher levels of redundancy, elasticity, and compatibility than traditional network architectures. Experimentation in this area has been aided by the integration of SDN and NFV within NetOS [15] which underpins a vision based on software, where any virtual functions can be added, managed, moved, and updated efficiently. NetOS contains the basic SDN and NFV building blocks needed to facilitate higher level coordinated Management and Orchestration functions envisaged in the SELFNET proposal.

The complementary nature of SDN and NFV technologies has been leveraged in recent proposals such as the virtual machine migration scheme from Cannistra et al. [16] that uses an SDN/NFV arrangement to maintain service availability during migration. In the EmPOWER [17] project a 30-node testbed with integral monitoring tools has been built to facilitate SDN/NFV experimentation in the area of energy consumption of Wi-Fi networks.

There are a number of existing EU and international projects utilizing SDN and/or NFV technologies. Among the existing projects, some have focused on deploying large-scale Pan-European and even transcontinental SDN/NFV testbeds; high-profile examples include FED4FIRE [18], OFELIA [19], BonFIRE [20], FELIX-EU [21], and SMART-FIRE [22]. In terms of current state of the art, all of these testbeds may be viewed as facilitators of innovative experimentation. SELFNET may, where possible, make use of these existing EU testbed infrastructures to facilitate experimentation, evaluation, and validation of both individual SELFNET components and the entire SELFNET framework.

Of the other large-scale projects in this area, some have investigated use cases in the SDN domain (e.g., CITYFLOW [23], ALIEN [24], NetIDE [25], and NetVolution [26]) and others have proposed NFV solutions to specific use cases (e.g., PRISTINE [27], CloudNFV [28], and MCN [29]). Only a few projects have addressed the issue of SDN/NFV coordination and deployment (e.g., T-NOVA [30] and UNIFY [31]). Nevertheless, none of the existing projects has achieved a fully integrated and coordinated SDN-NFV solution in particular for managing 5G mobile networks as proposed in SELFNET.

Apart from SDN and NFV, SELFNET further explores Self-Organizing Networks (SON) for automatic 5G network management tasks. SON solutions in LTE (or LTE-Advanced) networks are typically classified into three categories including self-configuration, self-optimization, and self-healing [32]. Self-configuration refers to the dynamic plug-and-play configuration capability of newly deployed LTE eNBs, whilst self-optimization enables already deployed eNBs to automatically adapt to radio conditions and network loads. The self-healing function attempts to recover from temporary bottlenecks or failures in the network, for example, eNB outage.

Few EU projects have explicitly considered any aspects of SON; CROWD [33] is a noticeable example in this category, focusing on fine-tuning capacity, optimizing Medium Access Control (MAC) for LTE and Wi-Fi, and managing energy efficiency and connectivity in dense networks. It is noted that existing work on SON has been concentrating on the low network protocol stack layers (primarily the physical and MAC layers). In contrast, SELFNET targets high-level SON management functions in terms of self-healing, self-protection, and self-optimization, mainly based on the network and upper layer, and thus is largely independent of specific physical or MAC layers (although cross-layer design involving the low layers has also been taken into account). This approach ensures that SELFNET will be future-proof as 5G air interface and MAC schemes have not been defined yet. Moreover, SELFNET SON will operate in the Software-Defined and Virtualized Network domain and thus can be compatible with non-3GPP systems. In addition, SELFNET SON is expected to be complementary to low-level SON functions, for example, those in 3GPP networks.

SELFNET aims to generate a significant impact on the development of 5G, mainly in societal, operational, and innovation levels.

At societal level, SELFNET will contribute by enabling ubiquitous, robust, and continuous service access for subscribers underpinned by a reliable self-managing network. SELFNET will perform QoE oriented self-optimization of the network traffic, especially video traffic. SELFNET will also be able to provide early reactive (or in certain cases proactive) responses against cyber-attacks and therefore effectively reduce the number of attacks reaching the target destination, in concordance with the security requirements of 5G. In addition, autonomic management when combined with SDN and NFV in SELFNET will help to reduce the number of physical devices and the usage of existing devices, therefore reducing the energy consumption.

At operational level, the scalability and extensibility in SELFNET will also help to decrease the capital and operational costs directly related to deployment and management of new network functions. Consequently, SELFNET contributes to reduce the TCO of the network infrastructure, according to the management and operation requirements of 5G. Additionally, through automation, SELFNET will reduce the lifecycle of creating and deploying new service, a KPI proposed by 5G PPP.

At innovation capacity and knowledge integration level, SELFNET will open up a wide range of opportunities in low density areas, facilitating the prompt and cost-effective creation and deployment of Virtualized Network Functions without any significant investment. SELFNET will also provide a new innovative business ecosystem based on open APIs and software, which will allow operators to lease resources on demand.

3. SELFNET Architectural Standard Foundations

The specification of the SELFNET architecture takes into account all the requirements expressed by the SELFNET use cases and is aligned with the most relevant standards which are the foundations of the project: ETSI NFV [34], Open Networking Foundation (ONF) [35], and TMForum [36].

NFV [34] is the ETSI group devoted to standardize the virtualization of Networks Functions (NFs). It intends to define the complete architecture required to accommodate the challenges of the new virtualization paradigm. The architecture covers runtime and management aspects, capable of managing the entire lifecycle of a Virtual Network Function (VNF). Furthermore, it also comprises the management of Network Services (NS), which are built by orchestrating multiple VNFs, according to a Forwarding Graph (FG), using a catalog-driven approach.

ONF [35] is self-described as “an organization devoted to promote the utilization of software-defined technologies to program the network.” Following a Software-Defined Networking (SDN) approach, the network is separated into three different parts: the user-data plane, the controller plane, and the control plane. The user-data plane is responsible for forwarding the user traffic, while the controller plane is composed of SDN controllers which provide high-level APIs to the control plane above. The control plane is responsible for programming the network, easing the creation of new applications and speeding up the rollout of new services.

The TeleManagement Forum (TMForum) is a telecom industry association devoted to provide guidelines to help operators to create and deliver profitable services. One of the biggest TMForum achievements is the definition of a complete telecom business process (eTOM) and application (TAM) maps, including all activities related to an operator, from the services design to the runtime operation, considering assurance, charging, and billing of the customer, among others. In order to accommodate the SDN/NFV impacts,

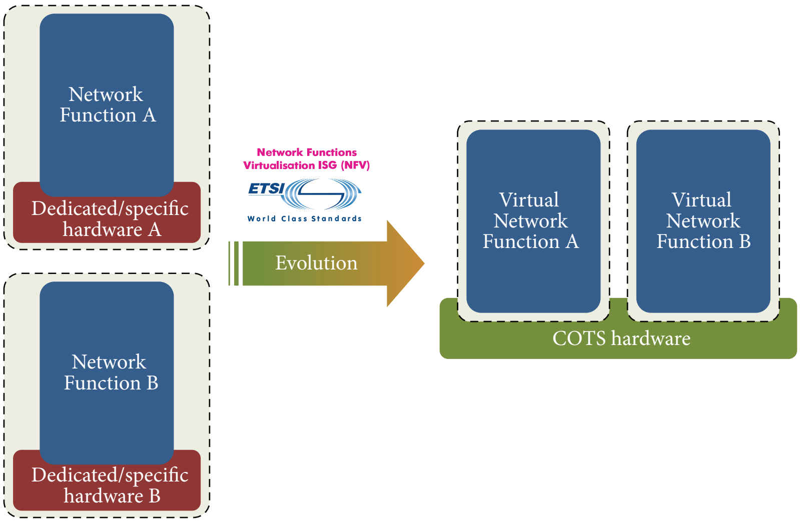
Network Function A
Dedicated/specific hardware A
Network Function B
Dedicated/specific hardware B
Network Functions Virtualisation ISG (NFV)
ETSI
World Class Standards
Evolution
Virtual Network Function A
Virtual Network Function B
COTS hardware

Figure 1: Basic NFV concept.

the TMForum has created the Zero-Touch Orchestration, Operations and Management (ZOOM) program [36], which intends to build more dynamic support systems, fostering service and business agility.

3.1. ETSI NFV. The creation of ETSI NFV [34] intended to bring to the telecom operations domain some IT (Information Technology) tools, in order to take advantage of cloud principles, like on-demand, agility, scalability, or pay-as-you-go (PAYG), among others. The hardware and software decoupling and the consequent utilization of Common Off-The-Shelf (COTS) hardware can also be applied on network functions, leading to a cost reduction and vendor independency.

The first basic step to take towards NFV is the "cloudification" of network functions (NFs). For this, the network function has to be implemented apart from a dedicated/specialized hardware and to be able to run on top of COTS hardware. Typical examples of VNFs are common routers or firewalls but can also be mobile or fixed components, like P-GWs, eNBs, or OLTs. Figure 1 depicts this simple concept.

For the sake of simplicity, it is assumed that the resulting VNF has the very same set of features as the equivalent Physical Network Functions (PNFs). However, the "cloudification" may also lead vendors to reshape their offers, in order to accommodate them to the cloud environment. This way, multiple PNFs can be collapsed into a single VNF; one PNF can be separated into multiple VNFs, or any other N : M mapping. These changes can also be decided in order to optimize resources or to improve the management of the function.

The "cloudification" of network functions can be further enhanced by using the Management and Orchestration environment. In such case, the platform manages the entire lifecycle of VNFs, not only performing the deployment and disposal but also managing the runtime operation, by migrating or scaling in/out VNFs, according to the function load, and making a more efficient use of resources. Such platform is also able to orchestrate combinations of VNFs according to a given Forwarding Graph (FG), in order to create more complex Network Services (NS).

Figure 2 depicts a simplified version of the full ETSI NFV architecture. In the left side, the execution and control (runtime) operations can be seen, while the right side shows Management and Orchestration. The bottom left shows the virtual infrastructure, which comprises hardware resources (COTS), the virtualization layer (hypervisors like KVM or VMware), and the virtual resources (e.g., VMs and VLANs). VNFs run on top of one or multiple VMs and using virtual network resources. On the top left, the Management Support Services (OSSs/BSSs) interact with the Management and Orchestration (right side) and with the VNFs. In the right, on the bottom, the Virtual Infrastructure Management (VIM) (e.g., OpenStack) interacts with the NFVI (hypervisor) to manage resources. On the top right, the Orchestrator and Management (MANO) manages the complete lifecycle of VNFs and orchestrate NSs.

3.2. ONF SDN. SDN intends to make the network simpler, more flexible, and more programmable. For that purpose, the SDN architecture splits the network functions into three parts: the user-data plane, the controller plane, and the control plane. The user-data plane (or Data Plane) is composed of dumb switching Network Elements (NEs), responsible for forwarding the user traffic according to basic commands received from the north (controller) interface. The controller plane is composed of SDN controllers, which provide basic commands to the south (user-data) and high-level APIs to the north (Control or Application Plane). Controller APIs are abstractions used to program the network, speeding up the creation of new services.

Figure 3 depicts the basic SDN concept. In the figure, it is assumed that the starting point is not a traditional PNF relying on a dedicated hardware but an already Virtualized NF (VNF), as described in the section above.

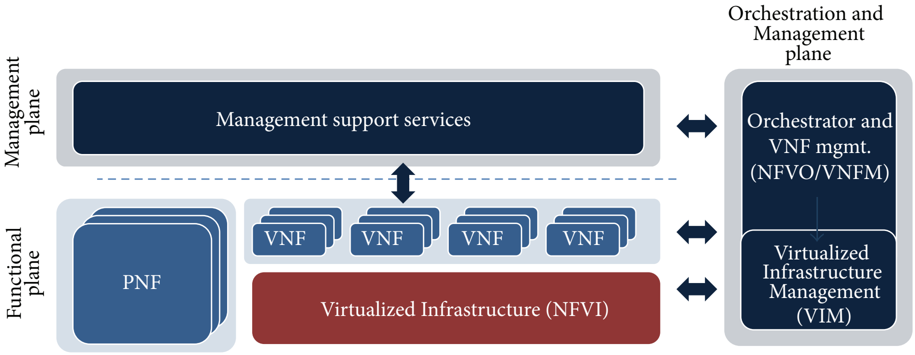
Orchestration and Management plane
Management plane
Management support services
Orchestrator and VNF mgmt. (NFVO/VNFM)
Functional plane
PNF
VNF
VNF
VNF
VNF
Virtualized Infrastructure (NFVI)
Virtualized Infrastructure Management (VIM)

Figure 2: Simplified NFV architecture.

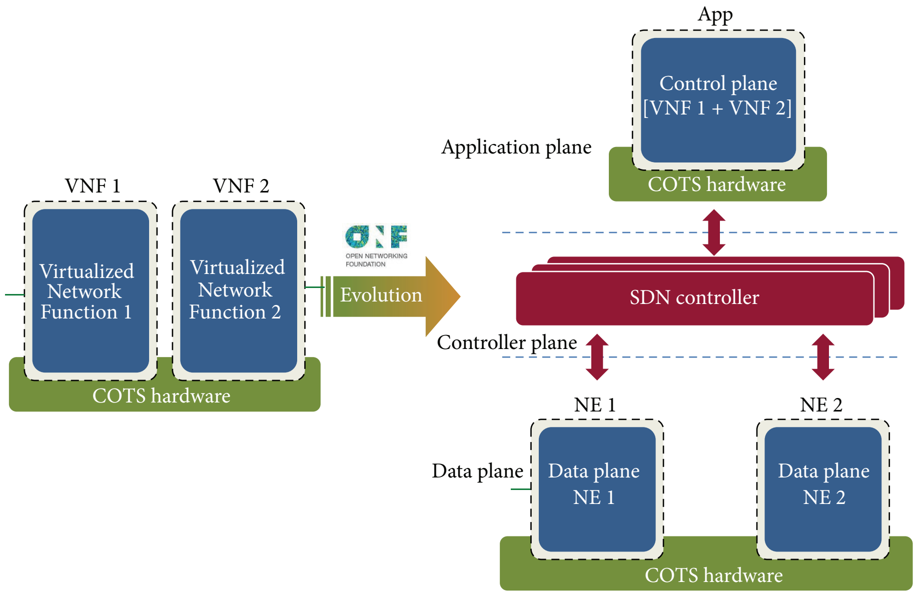

Figure 3: Basic Software-Defined Networking (SDN) concept.

Overall, the NE's forwarding process is fully commanded by the applications, which use high-level APIs provided by the SDN controllers. The SDN controllers interact with the NEs through low-level southbound APIs to enforce basic forwarding rules, using Command Line Interface (CLI) scripting, or protocols like Netconf [37] or the most recent OpenFlow [38]. SDN controllers provide a northbound interface, abstracting the programmer from the network details and making simpler the network service creation. This is the key advantage of the SDN model.

Figure 3 shows the evolution from the traditional to an SDN based approach. The transformation has not to be done using a 1 : 1 mapping. This means that a VNF may not move directly to the SDN paradigm by splitting itself into three parts. In fact, a single VNF can result into multiple NEs and even multiple applications (an N : N mapping).

An example of "SDNification" is illustrated in Figure 4.

The original scenario is a set of already virtualized routers. Each router has IPv4 and IPv6 forwarding feature as well as many others, like traditional routing protocols, for example, Open Shortest Path First (OSPF) and Border Gateway Protocol (BGP). On top of that, operators can configure the routers and build services like IPv4 connectivity, IPv6 connectivity, or enterprise VPNs, among others. Moving to the SDN paradigm, control and forwarding planes are decoupled. On the bottom, NEs are deployed with forwarding-only capabilities, applied according to the policies provided by the control plane. Control logic is implemented on the top, for example, OSPF, BGP, and all the service logic that permits the provisioning of a new access, a new VPN site, or VPN. In this particular case, there is an N : N mapping between the control plane (SDN Apps) and the user-data plane (NEs); that is, multiple applications implement different services on top of a set of NEs. NEs can be virtual software switches (e.g., OpenVSwitch [39]) or powerful hardware-specific switches.

3.3. Combining SDN and NFV. NFV and SDN have been created by different standardization bodies. However, they are complementary technologies. That is why many times they are referred to as NFV/SDN. Although they have value

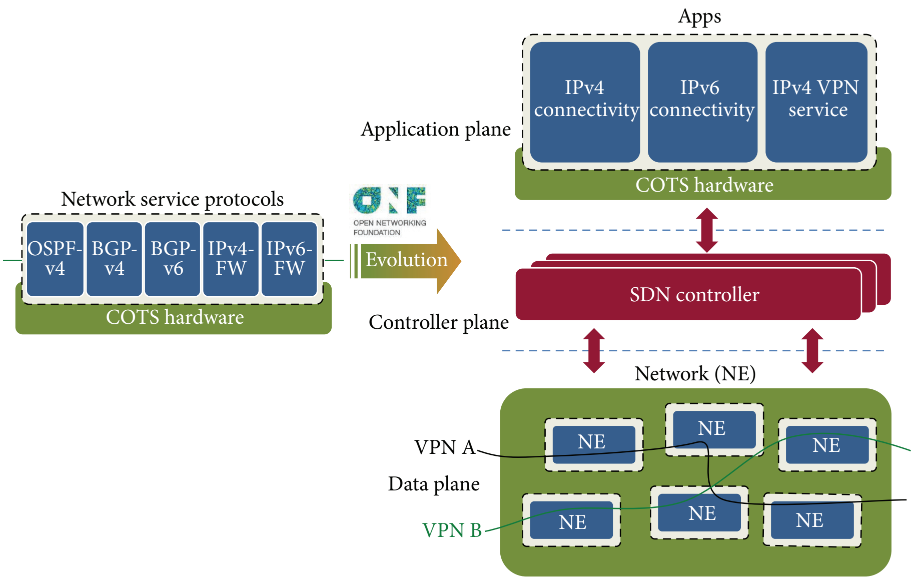

Figure 4: Software-Defined Network (SDN), for example, Transport Network.

separated, combined they have extra value. The NFV technology brings the possibility of creating new network functions on-demand, placing them on the most suitable locations and using the most appropriate amount of resources. However, this requires the SDN to be able to adjust the network accordingly, enabling network (re)configuration (programmability) and (re)sequencing (chaining).

As NFV and SDN come from different Standard Developing Organizations (SDOs), at the time of writing this paper, there is no combined architecture. For this reason, this section aims to seamlessly integrate NFV and SDN for the SELFNET project. We will take the ETSI NFV architecture as a baseline and will introduce the SDN paradigm.

Firstly, the ETSI NFV architecture, depicted in Figure 2, shows the VNFs (in the left part), which represent the Virtualized Network Functions. Next, according to the SDN model, depicted in Figure 3, the monolithic VNF is separated into three parts. Finally, combining both concepts, it results in the architecture depicted in Figure 5.

The combined architecture is compliant with the one defined by the ETSI NFV, while a few changes are introduced to accommodate the SDN concept. The naming of the architectural components is another issue that needs to be decided, since similar boxes have different names, depending on whether one looks at them from the NFV or the SDN perspective.

In order to consider "legacy" components (non-NFV and non-SDN), we kept the physical NFs in the leftmost side of the dashed red square of the architecture, meaning that we may have *Physical Network Functions* (PNFs), which do not apply the NFV and SDN models. The same way, one may have Virtualized NFs (VNFs) with no SDN capabilities. For those, the *Virtual Network Functions* (VNFs) naming is kept, as shown in the rightmost side of the dashed red square. In the middle, all components are SDN-aware, meaning that they are split into three layers. In the bottom layer (user-data plane) there are the Network Elements (NEs), which can be *Physical Network Elements* (PNEs) or *Virtual Network Elements* (VNEs). In this case, the names are chosen from the SDN world, since they describe the roles they are performing more clearly. On the controller layer, it is assumed that SELFNET may have multiple controllers at different levels, naming all of them as *SDN controllers* (SDN Ctrl). Finally, for the control layer, we used the naming *SDN application* (SDN App). This case does not specify if it is virtual or physical as it can be both, although it is believed that this layer will be mostly populated by virtual applications, considering that hardware specific is declining.

4. SELFNET Reference Architecture

Herein, the SELFNET reference architecture is defined with all architectural design considerations to meet 5G requirements and visions taken into account.

4.1. Architecture Overview. Self-organizing capabilities over 5G networks are provided by means of an architecture based on five differentiated layers as detailed in Figure 6.

The following logical high-level scopes are defined for each architectural layer:

(i) *Infrastructure Layer.* It provides the resources required for the instantiation of virtual functions (Compute, Network, and Storage) and supports the mechanisms for that instantiation. It represents the NFVI as defined by the ETSI NFV terminology.

(ii) *Virtualized Network Layer.* It represents the instantiation of the virtual networking infrastructures created by the users of the infrastructure as part of their normal operational plan and those created by the SELFNET framework as part of the SON capabilities.

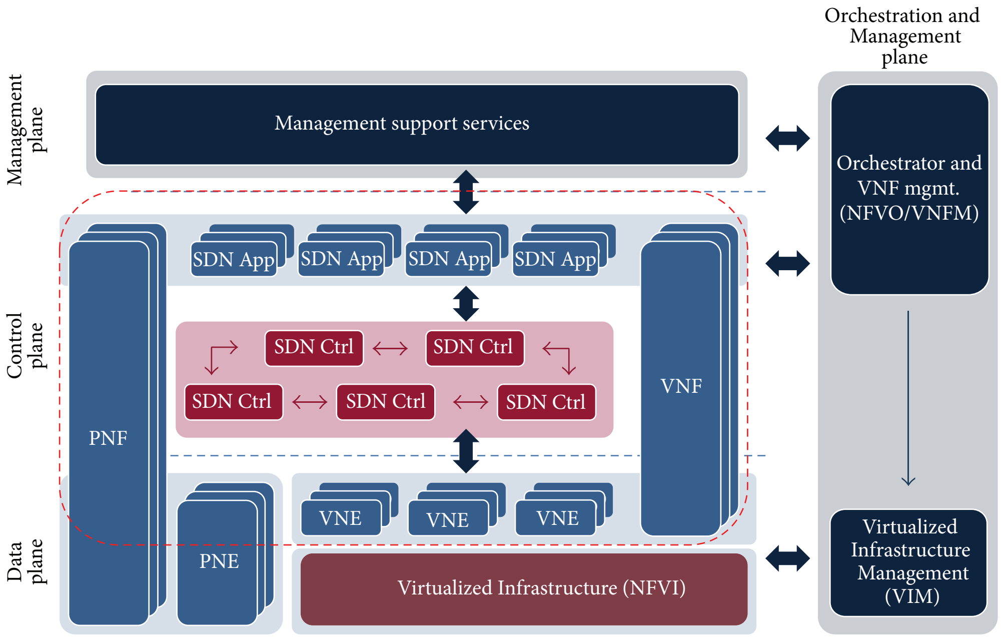
Orchestration and Management plane
Management plane
Management support services
Orchestrator and VNF mgmt. (NFVO/VNFM)
Control plane
SDN App
SDN App
SDN App
SDN App
SDN Ctrl
SDN Ctrl
SDN Ctrl
SDN Ctrl
SDN Ctrl
PNF
VNF
VNE
VNE
VNE
Data plane
PNE
Virtualized Infrastructure (NFVI)
Virtualized Infrastructure Management (VIM)

Figure 5: NFV and SDN combined architecture.

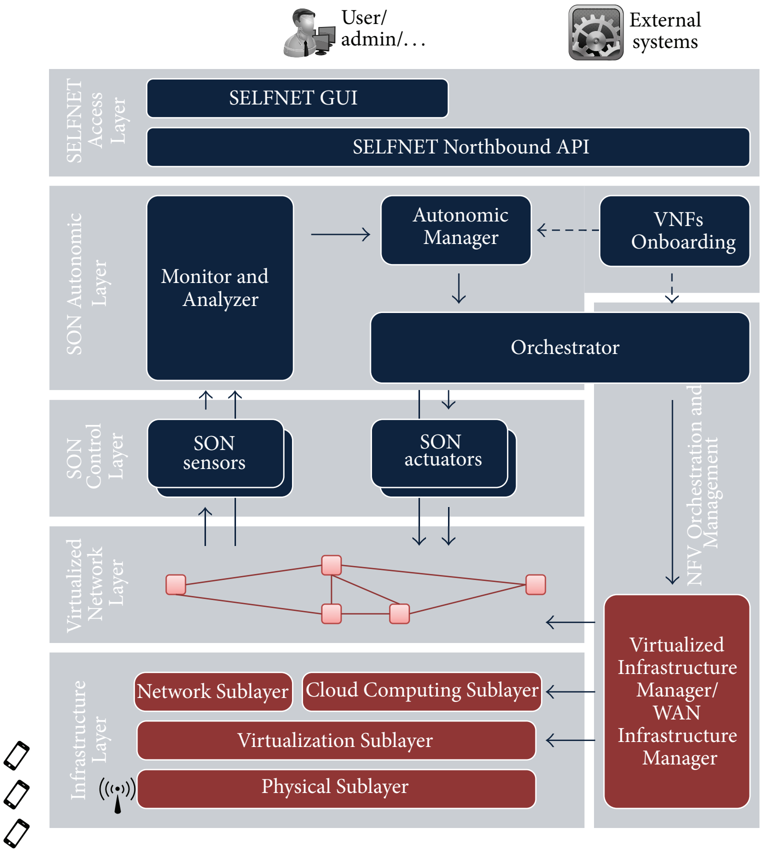

Figure 6: SELFNET Architecture Overview.

The layer is composed of a number of NFs interconnected in a designed topology in order to provide the functionalities required by the user. In the context of SELFNET, all NFs will be virtual and they will be chained across the virtual network topology.

(iii) *SON Control Layer.* It contains the applications that will enable the collection of data from sensors deployed through the entire system (SON Sensors) and the applications that will be responsible for enforcing actions into the network (SON Actuators) as part of the enabling mechanisms to provide network intelligence in 5G networks. *SON Control Layer* and *Virtualized Network Layer* have associated Control and Data Planes of the network. These planes are not represented in Figure 6 but are shown in Figure 5.

(iv) *SON Autonomic Layer.* It provides the mechanisms to provide the network intelligence. It collects pertinent information about the network behaviour, uses that information to diagnose the network condition, and decides what must be done to accomplish the system goals. It then guarantees the organized enforcement of the actions that are determined.

(v) *NFV Orchestration and Management Layer.* It corresponds to ETSI MANO layer and to its basic functions: VNFs orchestration and VIM and VNF management. It includes the management of the data centers virtual infrastructure, as well as the interdata center network, also known as WAN (Wide Area Network).

(vi) *SON Access Layer.* It encompasses the interface functions that are exposed by the framework. Despite the fact that internal components may have specific interfaces for the particular scope of their functions, these components contribute to a general SON API that exposes all aspects of the autonomic framework, which are "used" by external actors, like Business Support Systems or Operational Support Systems.

4.1.1. Infrastructure Layer. The infrastructure Layer envisioned in SELFNET has been intentionally aligned with the information available so far in terms of the architectural designs, principles, and components of 5G networks. SELFNET architecture has been designed to be as flexible as possible in order to deal with later modifications over such architectural decisions as part of the natural evolution of 5G networks.

The SELFNET infrastructure has been divided into different sublayers to split the functionality of the infrastructure in different architectural components. Figure 7 provides an overview of the Infrastructure Layer.

Physical Sublayer. SELFNET infrastructure is based on the principles of Mobile Edge Computing (MEC). A central data center is deployed providing high computational and communicational capabilities, whereas a large number of smaller data centers, located in the network edges and connected to the central data center, provide restricted and specific computational and communicational capabilities. It has been represented in Figure 7 with the central part and the edge parts of the figure, respectively. A logical location has been defined as a way to logically control where different services are allocated within the architecture. However, from the deployment point of view, such logical location can be allocated either in different physical locations geographically separated or within the same physical location. This concept would provide SELFNET with the flexibility to cope with the requirements posed by a large number of infrastructure deployments covering traditional 4G deployments and emerging cloud-RAN 5G deployments.

Communications between different elements of the infrastructure has to be considered from two different angles: physical and virtual connectivity. The physical connectivity to establish communications between different edge locations and the central data center will be assumed by SELFNET using either existing/traditional communication channels or assuming a natural evolution of those communication channels to face the KPIs imposed by 5G networks. SELFNET is not going to explore any innovation in the data plane of these communications channels. An architectural decision that could take significant impact in terms of performance is to consider the usage of a high-end hardware switches with SDN support in the connectivity between edges and data center to enable efficient data processing rates. This is an optional aspect to be analysed in SELFNET.

Virtualization Sublayer. On top of the Physical Sublayer, SELFNET framework will use a Virtualization Sublayer to provide the capability of providing virtual infrastructures on top of physical ones in order to enhance the management, isolation, and consolidation of computational resources. This sublayer is envisioned as an interoperable and interchangeable layer where different virtualization technologies (hypervisors) can be plugged and accessed by means of an abstraction access technology. This sublayer will provide a significant number of added values within the SELFNET since it will enhance the reliability, security, and continuity of services. The virtual connectivity between different VMs will be performed using software-based switching solutions that enable the connectivity of different virtual machines.

Cloud Computing Sublayer. On the control plane, the Control Computing Sublayer will be able to provide multitenancy capabilities over the current virtualized infrastructure enabled by the lower layers. This multitenancy will enable different tenants to use resources in an isolated and controlled way within the same physical infrastructure. In essence, this sublayer will be in charge of managing the physical infrastructure and providing functionalities to create, delete, and administer virtual infrastructures in a multitenant domain. The optimization and control in the usage of physical resources within a truly multitenancy virtual infrastructure will significantly reduce capital costs and will be an enabling technology to open new exploitation scenarios where multiple network operators are sharing computational and communicational resources and where

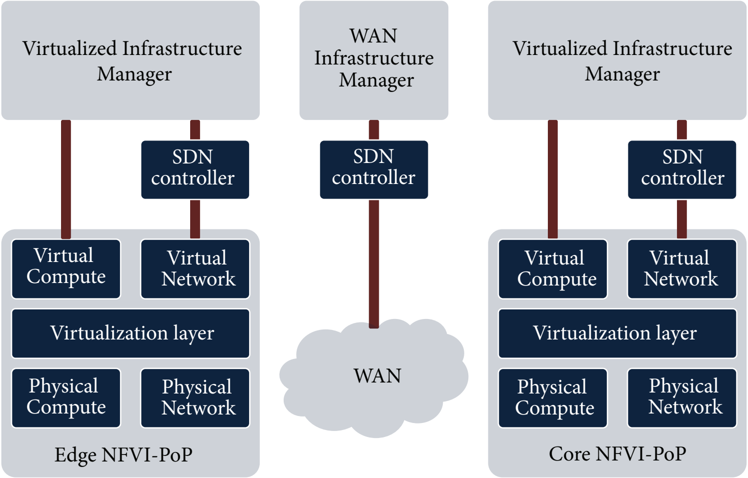

Figure 7: SELFNET Infrastructure Layer.

the same network operator is providing services to multiple customers simultaneously.

SDN Controller Sublayer. The usage of SDN-enabled switches is an architectural decision of the SELFNET infrastructure due to the fact that it will enable SELFNET to deal with the management of Network Services at two different ends: control and data plane. The combination of software-based and hardware-based SDN-enabled switches is being considered within the SELFNET architecture to provide more alternatives in terms of architectural locations and flexibility to deploy SDN applications to better fit the purpose.

4.1.2. Virtualized Network Layer. This layer represents the instantiation of a number of NFs and VMs interconnected in a designed topology in order to provide the functionalities required by the user. The Virtualized Network Layer is logically divided between Control and Data Planes and will be in charge of providing Virtualized Network Functions. This point is where Figure 5 can be used to refer to VNE and SDN App to denote data and control plane, respectively. Also, each application can be allocated in both data center and edge locations. In the data plane, VNE will provide services over the network traffic circulated through the virtual machines where such VNE is running. In the control plane, the usage of SDN controllers in order to plug different SDN applications with such a controller has been envisioned. The number of SDN controllers will depend on the performance required by each controller, the number of flows being processed, scalability requirements, and so forth. Therefore, conceptually SELFNET architecture has envisioned the differentiation of at least two logical places where SDN applications can be logically allocated: an edge SDN controller and a data center SDN controller, in order to provide a mechanism to control the allocation of the SDN applications within the 5G networks. It is worth mentioning that multitenancy support over the SDN applications running within the controller is an innovation that SELFNET will analyse in order to explore different ways to deal with the management aspect of SDN applications.

4.1.3. SON Control Layer. This layer contains the applications that will enable the collection of data from sensors deployed through the entire system (SON Sensors) and the applications that will be responsible for enforcing actions into the network (SON Actuators) as part of the enabling mechanisms to provide network intelligence in 5G networks. Figure 8 depicts an overview of all the sublayers of the SON Control Layer. The SON Control Layer must deal with complex network architectures in the Data Function Sublayer and at the same time ease the management and deployment of novel services. To support these requirements two mechanisms play a pivotal role: the use of homogeneous Actuation APIs and the abstraction of network infrastructures to simplify Management and Orchestration. A policy-based interface can combine these two mechanisms integrating monitoring and control functionalities in a common language. The role of the SON Control Layer is to translate autonomic network-wide policies into specific network elements' configurations.

SON Data Function Sublayer. The SON Data Function Sublayer provides the VNEs executed in the data plane of all the components used by SELFNET framework. Although SDN concepts are of significant importance in the SELFNET approach, the fact that specific implementations of VNF may also integrate control logic (i.e., legacy routing equipment or standalone network functions) leads eventually to an additional classification of the VNF according to this aspect: VNF with no control plane named simply as VNE in SELFNET terminology according to Figure 5 and VNF with control plane. Depending on the API accessed, it can be considered an SDN-compliant VNE and proprietary VNE. The coexistence of all these options is expected to be the norm for the following years. This is why the reader can see different alternatives in Figure 8.

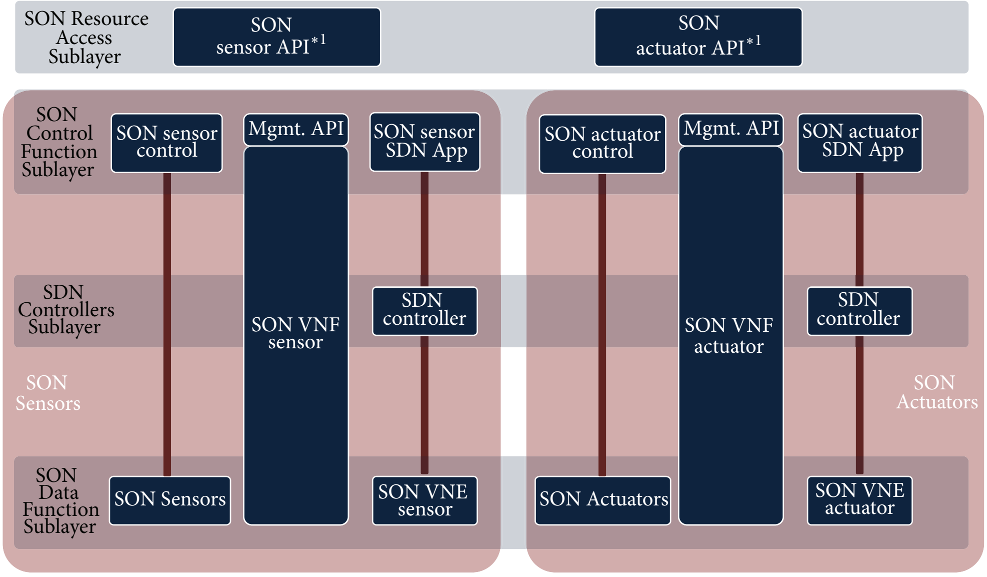

Figure 8: SELFNET SON Control Layer. *1 A logical perspective; from a physical perspective, we can have one or more SON Sensors and Actuator APIs either available on top of each component of the SON Control Function Layer or a single instance in charge of interacting with such components.

SDN Controllers Sublayer. This sublayer has already been explained in the infrastructure sublayer. In summary, it will enable the communication between the Control and Data Planes of the SON Sensors and Actuators in the same way that it enables the communications between any VNE and SDN Apps.

SON Control Function Sublayer. The Control Functions Sublayer consists of all the functionalities that are required to cope with the control of the instances available in the data plane. These functionalities will control VNEs using SDN Apps, proprietary control protocols, or management APIs to interact with the monolithic VNFs. Control Functions will be exposed through the SON Resource Access Sublayer in order to be able to orchestrate and control the VNF components.

SON Control Functions are required to support the following functionalities. Firstly, they provide the representation of the data plane resources in the context of the Orchestration and Management procedures; typically they provide the endpoints to be invoked for the integration and delivery of higher level sensing/actuation concepts. Secondly, they provide the mapping of the sensing and actuation commands onto the proper control commands according to the actual implementation of VNE.

Control Functions can be designed in different ways: virtual machines hosting a single application, physical computing resources hosting a number of applications, virtual computing resources hosting a number of applications, and so forth. SELFNET will provide flexibility for most of them and will try to provide support for the most appropriate design to fit its purpose efficiently. Accordingly, instantiation of a control function may involve deployment of a virtual machine, or provisioning of apps in application containers.

SON Resource Access Sublayer. The SON Resource Access Sublayer provides a homogeneous Sensing and Actuation API. It will make easier the integration of new SON Sensors and SON Actuators. In addition, it will enable the integration between the SON Autonomic Layer and this layer in order to enforce the autonomic decisions.

4.1.4. SON Autonomic Layer.

The SON Autonomic Layer is the topmost layer of the SELFNET architecture. This layer provides the mechanisms to provide network intelligence. The layer collects pertinent information about the network behaviour, uses that information to evaluate the network condition, diagnoses any pending/existing network issues, and decides what must be done to accomplish the system goals. It then guarantees the organized enforcement of the actions. In essence, the SON Autonomic Layer is split into four main sublayers: Monitor and Analyzer, Autonomic Management, Orchestrator, and VNFs Onboarding. These sublayers and their modules will be described in the upcoming subsections.

Monitor and Analyzer Sublayer. The main purpose of this sublayer is to provide a general overview of the framework that will be designed for monitoring and analysing the network behaviour or incidents. This sublayer is divided into three data processing levels, as shown in Figure 9. Each of them is briefly described below.

The *Monitor and Discovery stage* collects and stores all data from SDN sensors and NFV sensors. In addition, it must detect and notice changes in devices and NFV sensors involved in SELFNET. The required information is provided by SON Control Layer and Virtualized Network Layer. At this point, information provided by SDN/NFV sensors is differentiated from SDN/NFV devices. This differentiation is to speed up the processing tasks and facilitate the later

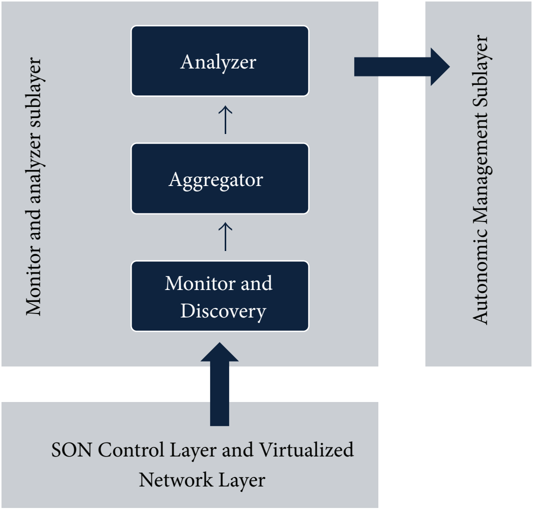
Monitor and analyzer sublayer
Analyzer
Aggregator
Monitor and Discovery
Autonomic Management Sublayer
SON Control Layer and Virtualized Network Layer

Figure 9: Monitor and Analyzer Sublayer.

correlation/classification. Gathered information is treated by a Data Management System (DMS). The DMS manages a database containing all the collected information. Among DMS responsibilities are storing/querying, access/privacy control, or refresh DB content. It also involves removing obsolete entries, which do not correspond to the current state of the network. DMS also deals with orchestration of information distributed in different DBs or SELFNET regions, or with applying the required control access functions and privacy policies.

The *Aggregation Module* performs aggregation and correlation of the low-level metrics provided by the Monitor and Discovery Module. This may involve performing different actions such as data normalization, verification, or correlation. In order to facilitate the later processing stages, redundant information will also be removed. At the end of this stage, aggregated, higher level metrics must be available at later stages of processing.

The goal of the *Analyzer Module* is the analysis of information provided by the Aggregation Module, thereby deriving information required for facilitating decision making. For this purpose, a comprehensive analysis of the information received is performed. Activities in this stage are divided into five main submodules: Identification, Assessment, Review, Prediction, and Situation Awareness as follows.

(i) *Identification*. It is the recognition of anomalous or suspicious SELFNET behaviours. Identified situations could match with different natures, such as deployment of new NFV or devices, congestion because of legitimate reasons, or suspicious threats labelled by self-protection sensors. Consequently, recognition of these events is required for allowing triggering self-optimization, self-healing, or self-protection actions.

(ii) *Assessment*. Once a new/suspicious situation is identified, the Assessment Submodule evaluates its relevance; a value which summarizes its impact is calculated based on predefined HoNs. It is important to highlight that the assessment will differ depending on the use cases features. For example, if an attempt of intrusion is detected, the impact of the situation will be calculated by taking into account self-protection measures, such as kinds of intruders, methods, pre/post conditions, or preview of the threat evolution. The assessment is in direct relationship with the latter decision making stages and facilitates adopting SELFNET response in proportion with relevance.

(iii) *Prediction*. A forecast of the coming situations is provided by the Prediction Submodule. It considers the global state of the system, and particular situations triggered by the Identification Submodule. This allows making proactive decisions and a better understanding of the Situation Awareness on SELFNET.

(iv) *Review*. Whilst the Assessment Submodule deals with previously identified situations, this Review Submodule analyses the impact of decisions made. This allows an evaluation of whether to adopt decisions that will involve different course of action or to continue with the current strategies based on historical decisions for a similar scenario. Through such a review, it is also possible to improve the assessment of identified situations.

(v) *Situation Awareness*. It refers to the global situation of SELFNET but also displays particular events discovered on the previous processing stages. In order to enhance the behaviour of the coming processing stages, this Situation Awareness Submodule summarizes the information gathered and calculated by them.

Autonomic Management Sublayer. The Autonomic Management Sublayer can be regarded as the "brain" of the SELFNET framework, which takes advantage of mechanisms in the field of Self-Organizing Networks, artificial intelligence, data mining, and pattern recognition to enable autonomic and automated network management. As illustrated in Figure 10, this sublayer will provide the capabilities of self-healing, self-protection, and self-optimization by means of proactively and reactively dealing with existing and/or potential network problems. This sublayer also defines the Tactical Autonomic Language (TAL) that specifies the resolution strategies to guide the automatic actions provided. The Autonomic Management Sublayer consists of the following modules.

(i) *Tactical Autonomic Language* (TAL) includes the autonomic language and its associated library, which define the autonomic strategies inside the SELFNET framework.

(ii) *Diagnostic Module* takes advantage of artificial intelligence, data mining, and stochastic algorithms to provide intelligence in diagnosis of the network problems and provide the information of the best strategy to be taken. This will be achieved by using tactical strategies to determine the resolution actions to be taken in the

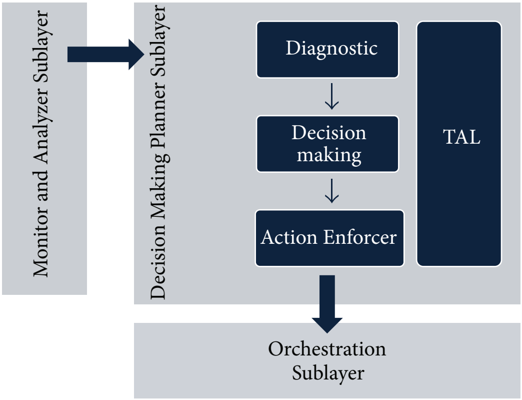

Figure 10: Autonomic Management Sublayer.

network for the cases where no completely defined, well-known strategies about how to react exist.

(iii) *Decision Making Planner Module* is in charge of deciding a set of corrective and preventive actions to deal with the identified and potential network problems, in both reactive and proactive manners, based on the incoming diagnostic information. For such purposes, this framework will interrogate the HoN Query Services in order to know the state of the network. This module will make use of two different approaches for providing a response. The first approach will be the direct application of the resolution strategies defined in the TAL. The second approach will be the integration of artificial intelligence algorithms guided by such tactical strategies in order to determine resolution actions to be taken in the network, even for the cases where there are not completely defined tactical strategies about how to react.

(iv) *Action Enforcer Module* is responsible for providing a consistent and coherent scheduled set of actions to be enforced in the network infrastructure. For this purpose, this module will receive and recognize the messages provided by the Decision Making Planner and will validate, organize, and refine such messages by means of applying conflict detection and resolutions techniques, language refining techniques, and so forth, in order to provide an implementable plan ready to be enforced.

VNFs Onboarding Sublayer. The VNFs Onboarding Sublayer is composed of those VNF management functions, mechanisms, and tools responsible for the encapsulation of the SELFNET actuators and sensors into common and homogeneous VNFs. The main target of the VNFs Onboarding is therefore to provide a unified abstraction of the SELFNET VNFs while exposing common lifecycle management primitives to be used for automated deployment, configuration, reconfiguration, and termination of any VNF.

This means that the VNFs Onboarding Sublayer should provide mechanisms and tools to containerize the developed SELFNET actuators and sensors into encapsulated VNFs that can be managed, configured, and controlled through common APIs. On top of this unified VNF abstraction that aims to standardize the way SELFNET actuators and sensors are stored, advertised, and used by the SELFNET platform, the VNF Onboarding also provides automated mechanisms and procedures for the deployment of the encapsulated VNFs into the virtualized infrastructure.

With respect to the standardized ETSI approach for VNF management and coordination, the VNF Onboarding sublayer advances the traditional ETSI NFV architecture and in particular the Management and Orchestration (MANO) part with the aim of enriching its well-defined APIs for a common management, configuration, and control of heterogeneous VNFs. In SELFNET, the goal is to containerize both VNF and EMS components of the ETSI NFV architecture to provide a common and uniform approach to lifecycle management across all the heterogeneous SELFNET sensors and actuators VNFs that will be developed. In addition, the enhancement of pure Management and Orchestration functions performed by MANO components (NFVO, VNFM, and VIM) will provide automated deployment and configuration functions for the encapsulated VNFs.

Orchestration Sublayer. The Orchestration Sublayer is in charge of the real deployment of the NFV and SDN Apps (mainly actuators and/or sensors depending on different use cases), following the specified instructions (action plans) from the Decision Making Planner Sublayer. As shown in Figure 11, this sublayer is composed internally by three modules: the Orchestrator whose role is to receive and process action requests and implement the actions; the Application Manager who enables the Orchestrator with management capabilities of the VNE, SDN, and VNF Apps; and the Resources Manager who allows the management and configuration of infrastructural resources to support the actions of the Orchestrator. Further description of these elements is available below.

Orchestrator Module. Orchestrator Module consists of the following.

(i) *Apps Deployment*. It processes the received action plans from the Decision Enforcer module, resolves the dependency/order/priority among different actions, and executes the actions by calling and deploying the selected Apps to the right place at the right time.

(ii) *Resource Brokering*. It provides a resource brokering service to optimize system-wide resource usage especially virtual/cloud resource brokering. This also ensures that the Apps are allocated with sufficient resources to execute their actions and fulfill their tasks.

The translation of the requirements in resources can be obtained through resource mapping algorithms. The result of the mapping algorithms is a workflow that after completion will fulfill the intent. Moreover, to execute the workflow, the Orchestrator will process the dependencies between the steps

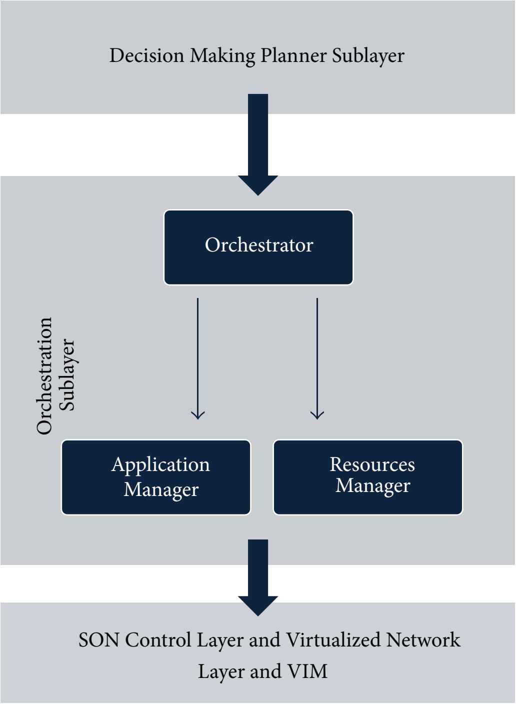
Decision Making Planner Sublayer
Orchestrator
Orchestration Sublayer
Application Manager
Resources Manager
SON Control Layer and Virtualized Network Layer and VIM

Figure 11: Orchestration Sublayer.

in the workflow so that an ordered list of steps is obtained. A single step can be an action on an application or infrastructure resource, which are available through the Application Manager and the Resources Manager, respectively.

Application Manager Module. The main role of the Application Manager is to manage the lifecycle of the Apps, thereby enabling the Orchestrator to install, remove, update, configure, start/restart, stop, or resume a selected App based on the action plan being executed. A uniform application layer is provided to the Orchestrator for executing the actions via deploying Apps. Moreover, the Application Manager maintains the current status of the Apps, which can be queried by the Orchestrator.

This module deals with provisioning both of data layer artefacts that are not subject to virtual resource management but rely on deployment of application components in application container that may be in a VM provisioned in earlier steps and control layer artefacts that are linked with associated data layer resources via the appropriate control APIs and that are responsible for abstracting the NFV Apps under the Sensor/Actuator Abstraction API. To fulfil this set of events, the Application Manager will use processes common in IT environments to configure applications, such as configuration agents, specific protocols, automation tools, or scripts.

Resources Manager Module. This component provides a reference point for the Orchestrator to interact with the cloud controller and network controller. Moreover, the Orchestrator can use this component to request the provisioning of virtual resources and subsequent management. The Resource Manager provides an abstract, uniform representation of available resources and their dependencies to facilitate the successful call, deployment, and operation of the Apps by the Orchestrator. Also as a part of its role, the Resources Manager is to provide an overview of the resources, which will equip the Orchestrator with service/application and resource mapping capabilities. The Resources Manager will have the means to discover and assess the availability and current load of resources. In addition, although this component has limited autonomy, it should still be able to perform automated management and optimization procedures, for example, releasing virtual resources that are idle for some time.

4.1.5. NFV Orchestration and Management Layer. This layer contains the management capabilities to control both compute infrastructures and network infrastructures available in the architecture. Figure 12 shows graphically the different components available in this layer.

VIM Cloud Management Sublayer. The VIM is a multitenant service platform used in data centers to provide Cloud Computing services (e.g., Openstack). Through this platform tenants, previously registered in the platform, can request infrastructure resources such as computing, network, storage, or other supporting resources (e.g., load balancers and DNS). In NFV infrastructures, VIMs are used to manage data centers that can be in a central location or at the edge of the network, also known as PoPs. In SELFNET the VIM will provide the necessary resources for the provisioning of SDN Apps, VNFs, or other components deployed on top of COTS. At the north of VIM the Orchestration Sublayer, more specifically the Resources Manager, will request resources (e.g., VMs and internal virtual networks) that will be provisioned using hypervisors and SDN controllers.

WIM NFV Management Sublayer. The Wide Area Network Infrastructure Manager (WIM) is the Wide Area Network counterpart of the VIM. Likewise, it is a multitenant service platform used by network operators to control the networking services. In legacy environments, WIMs typically use technologies such as MPLS which were not designed to support NFV services. The movement to SDN in the WAN plays an important role to fulfill these novel services with heavier dynamic requirements. In SELFNET the WIM uses SDN controllers to provide dedicated and isolated virtual networks. These virtual networks, which enable the Virtualized Network Layer are requested at the north by the Resources Manager, part of the Orchestration Sublayer. This layer provides the enabling technologies to perform the deployment of new SDN Apps within the SELFNET framework by interfacing with the SDN Controller Sublayer. The SDN Apps running in this layer will be provided by a number of other layers of the SELFNET architecture. For example, in the NFV Orchestration and Management Layer will provide the SDN Apps for the management functionality to be able to control the topology of the Virtual Network Layer. Also, the SON Control Layer will contribute to this layer providing the SDN Apps for the monitoring of network metrics and for the

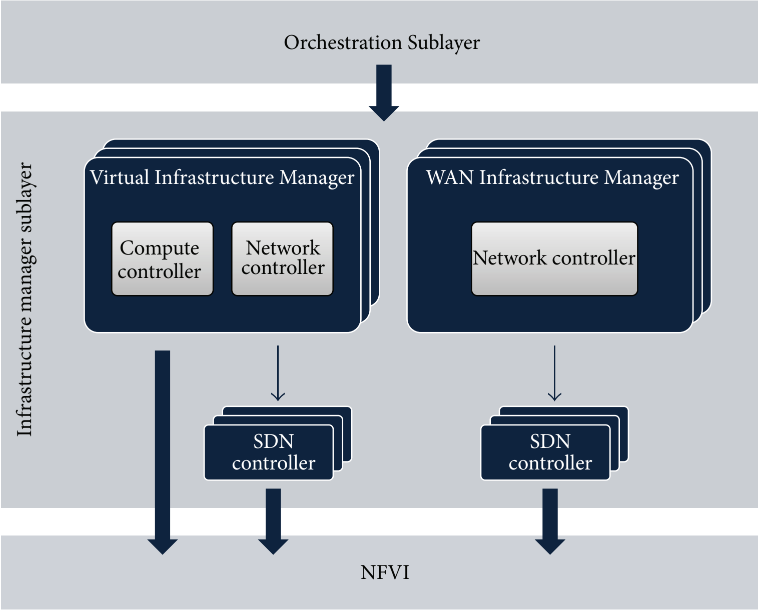

Figure 12: NFV Management and Orchestration Sublayer.

enforcing of actions within the network. This layer will also provide support to perform the deployment of non-SDN-compliant control applications if it is appropriate to optimize the control plane of the architecture.

4.1.6. Access Layer. The Access Layer is located at the top of SELFNET architecture and is the interaction point between the service administrator and service platform, providing an abstraction to manage the service's lifecycle. As shown in Figure 13, this layer is composed of a northbound API, divided into a set of APIs with very specific goals which are based on REST. A Web Graphical Interface (GI) is also provided.

SELFNET Northbound API Sublayer. The northbound APIs' main goal is to provide a network abstraction, that is, a uniform way of communication between the control layer and the user interface while producing an abstraction for GI interactions. The GI is heavily dependent on the REST API, and these components will interact with each other in order to favour a simple way of work hiding the complexity from the administrator. These APIs provide interfaces to control and manage the SELFNET's services lifecycle, divided into authentication, authorization and a Policy Decision Point (PDP), infrastructure overview, monitoring, service composition, autonomic decision, conflict resolution, reporting, UC specific API, VNF catalog, and alarms and events. Although at this stage this API has not been designed yet and it will be done as next step along the roadmap of SELFNET project, the envisioned functionalities are the following.

(i) *AuthN/AuthZ/PDP*. This controls not only the access to the platform but also the actions that each user can perform, that is, managing user groups and permissions. Within this API a PDP exists that, via a service adapter, interacts with other layers, like OSS/BSS, and controls higher permissions out of SELFNET's scope. This not only is the entry point for SELFNET's GI but also allows for the appearance of other custom applications to be developed in the future or the integration of already existing applications.

(ii) *Infrastructure Overview*. It offers functions to control and manage SELFNET service provider physical resources consumed by a specific service. Using this, the service administrator can evaluate the service performance.

(iii) *Monitoring*. It allows the network administrator to understand if all SLAs or KPIs are being fulfilled as requested. Through this, the administrator will be capable of recovering HoN, QoE, and QoS metrics.

(iv) *Service Composition*. It offers an overview of the service, that is, all the VNFs that compose the service, but also an interface to control and manage these VNFs. Changing service's SLAs or KPIs and manual operations like scaling, removing, reconfiguring, or migrating the service will be available on this module.

(v) *Autonomic Decision*. Based on the defined SLAs and KPIs, SELFNET will perform autonomic decisions, like scale service's physical resources. The Autonomic API will not only list these decisions but also allow a feedback system that will improve the data mining system responsible for improving these decisions. In addition, it will provide capabilities to assist in the configuration of the autonomic policies that govern SELFNET self/organized capabilities.

(vi) *Conflict Resolution*. Whenever an autonomic decision creates a serious conflict between rules and it cannot be resolved by SELFNET SON automatically, some

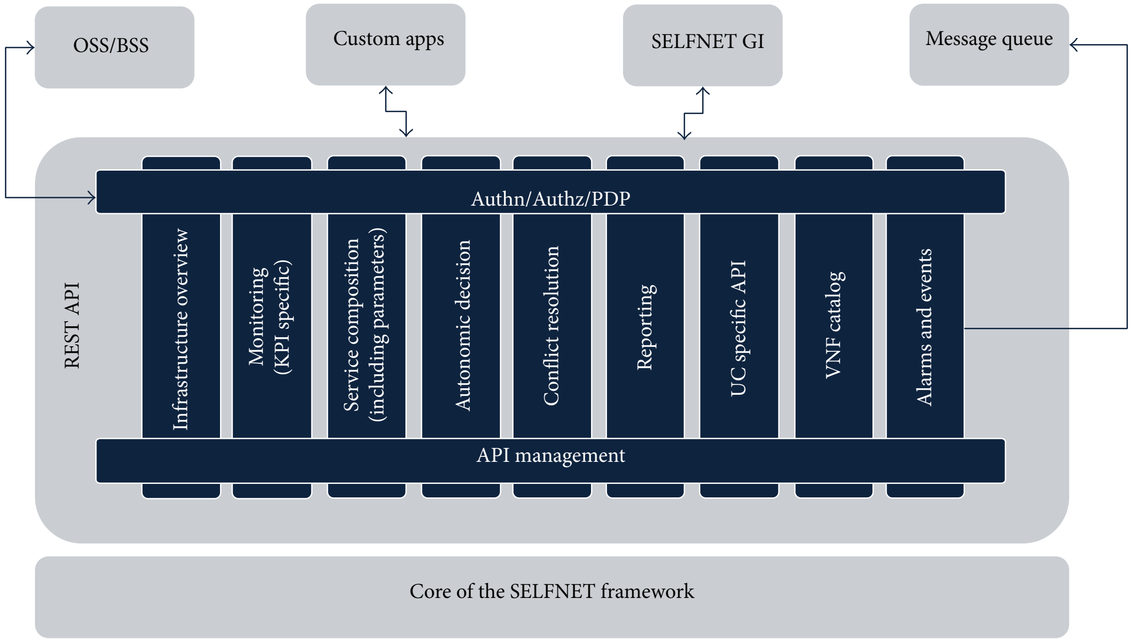

Figure 13: SELFNET Access Layer.

level of human intervention will be required, and thus the Conflict Resolution API will be called to action providing methods to solve the conflicts and overrule actions. In fact, the network administrator has the privilege to stop, verify, or manually enforce any of the actions.

(vii) *Reporting*. All service logs will be available through the report interface, differentiated by its own category and level, so the logs can easily be traced by service, date, and level.

(viii) *UC Specific API*. Any specific API, related to a use case, can be included in the SELFNET Northbound. This adaptation falls under the UC specific API.

(ix) *VNF Catalog*. In order to have an overview of the available VNFs there is an interface, VNF Catalogue, responsible for listing and describing each VNF available on the SELFNET's catalogue. The services can query this interface in order to select which VNF to use in different situations.

(x) *Alarms and Events*. In order to report every alarm and system event, for example, whenever a VNF fails or a VNF instance is scaled, there is an alarm and report interface. These operations will be sent to an external message queue that will be used to report these cases to systems that need this information. This module allows a control over who receives a specific alarm and where an alarm must be sent.

SELFNET Graphical User Interface Sublayer. The GI will be the main interaction point between SELFNET's operators and the SELFNET framework. In order to retrieve and send information to/from the SELFNET's framework, the REST API client allows seamlessly accessing SELFNET Northbound APIs' features. All information displayed by the GI as well as all the features provided by it will use SELFNET Northbound APIs' provided features. GI will authenticate its usage by requesting users to insert a valid set of credentials and not only check if a given user can access the SELFNET's GI but also identify the respective role of the authenticated user. This role will be used by GI to filter which features will be unlocked and displayed to the user.

SELFNET's GI will allow viewing and managing SELFNET both on a high level and on a low level. On a higher level, users allowed will be able to see the current status of SELFNET monitoring system's KPIs as well as listing system errors, messages, and warnings. Users will be able to audit SELFNET's autonomous decisions allowing users to understand if autonomously taken actions are valid. Users will also be able to take actions on these autonomous behaviours, correcting problems that may arise. As mentioned before, SELFNET will provide a low-level view on SELFNET's deployed sensors and devices listing them and allowing selecting each one of them in order to view its information or take an action. For each device, its performance, logging information, and errors will be available allowing operators to check the status of each one of them. Furthermore, the connections with other devices and sensors will also be listed allowing operators to understand its interactions and communications flow. By using this GI, operators will be able to take actions on these sensors' and devices' behaviours by reconfiguring their properties and interconnections.

5. Conclusions

This paper presented the SELFNET reference architecture for Self-Organized Network Management in Virtualized and

Software-Defined Networks, contextualizing it with related work and elaborating on the used technologies.

SELFNET is a scalable, extensible, and smart network management architecture which explored the integration of cutting-edge technologies widely recognized as key enabling technologies for 5G systems, such as Software-Defined Networking (SDN), Network Function Virtualization (NFV), Self-Organizing Network (SON), Cloud Computing, and Artificial Intelligence to implement an autonomic network management system addressing a number of essential management tasks, such as automated network monitoring, autonomic network maintenance, automated deployment of network tools, and automated network service provisioning.

The SELFNET design focused on three major network management concerns: self-protection, self-healing, and self-optimization, to propose its six layers' architecture topology: the Infrastructure Layer, the Virtualized Network Layer, the SON Control Layer, the SON Autonomic Layer, the NFV Orchestration and Management Layer, and the Access Layer. These layers have been proposed in line with the NFV and SDN concepts, namely, on the ETSI architecture that is determined by ETSI.

SELFNET introduces intelligence, self-organizing, and autonomic capacities to 5G networks, providing an open environment to foster innovation and decrease the CAPEX and OPEX of new applications.

More information about SELFNET project can be found at the website [4] and more details on the work present on this paper will be publically available on SELFNET Deliverable 2.1—Use Cases Definition and Requirements of the System and its Components [40].

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work was funded by the European Commission Horizon 2020 5G-PPP Programme under Grant Agreement number H2020-ICT-2014-2/671672—SELFNET (Framework for Self-Organized Network Management in Virtualized and Software-Defined Networks). The authors thank all the SELFNET partners for their support in this work.

References

[1] White paper—Enhance Mobile Networks to Deliver 1000 Times More Capacity by 2020, NOKIA, 2014.

[2] "Mobile traffic forecasts 2010-2020—a report by the UMTS Forum," January 2011.

[3] "Top Ten Pain Points of Operating Networks," Aviat Networks, 2011, http://www.portals.aviatnetworks.com/exLink.asp?8497200ON93Q69I35489972.

[4] EU SELFNET Project, "Framework for Self-Organized Network Management in Virtualized and Software Defined Networks," Project reference: ICT-2014-2/671672. Funded under H2020, http://www.selfnet-5g.eu/.

[5] 5G Infrastructure Public Private Partnership—5G PPP, https://5g-ppp.eu/.

[6] S. Costanzo, L. Galluccio, G. Morabito, and S. Palazzo, "Software defined wireless networks: unbridling SDNs," in *Proceedings of the European Workshop on Software Defined Networking (EWSDN '12)*, pp. 1–6, Darmstadt, Germany, October 2012.

[7] C. Chaudet and Y. Haddad, "Wireless software defined networks: challenges and opportunities," in *Proceedings of the IEEE International Conference on Microwaves, Communications, Antennas and Electronics Systems (COMCAS '13)*, pp. 1–5, IEEE, Tel Aviv, Israel, October 2013.

[8] A. Detti, C. Pisa, S. Salsano, and N. Blefari-Melazzi, "Wireless Mesh Software Defined Networks (wmSDN)," in *Proceedings of the IEEE 9th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob '13)*, pp. 89–95, IEEE, Lyon, France, October 2013.

[9] A. Basta, W. Kellerer, M. Hoffmann, K. Hoffmann, and E.-D. Schmidt, "A virtual SDN-enabled LTE EPC architecture: a case study for S-/P-gateways functions," in *Proceedings of the IEEE SDN for Future Networks and Services (SDN4FNS '13)*, pp. 1–7, Trento, Italy, November 2013.

[10] K. Pentikousis, Y. Wang, and W. Hu, "Mobileflow: toward software-defined mobile networks," *IEEE Communications Magazine*, vol. 51, no. 7, pp. 44–53, 2013.

[11] B. Naudts, M. Kind, F. Westphal, S. Verbrugge, D. Colle, and M. Pickavet, "Techno-economic analysis of software defined networking as architecture for the virtualization of a mobile network," in *Proceedings of the 1st European Workshop on Software Defined Networking (EWSDN '12)*, pp. 67–72, IEEE, Darmstadt, Germany, October 2012.

[12] B. Haberland, F. Derakhshan, H. Grob-Lipski et al., "Radio base stations in the cloud," *Bell Labs Technical Journal*, vol. 18, no. 1, pp. 129–152, 2013.

[13] J. Costa-Requena, "SDN integration in LTE mobile backhaul networks," in *Proceedings of the 28th International Conference on Information Networking (ICOIN '14)*, pp. 264–269, IEEE, Phuket, Thailand, February 2014.

[14] ETSI Industry Specification Group (ISG), "Network function virtualization (NFV) white paper," in *Proceedings of the SDN and OpenFlow World Congress*, Frankfurt, Germany, October 2013.

[15] V. Lopez, O. Gonzalez de Dios, B. Fuentes et al., "Towards a network operating system," in *Proceedings of the Optical Fiber Communications Conference and Exhibition (OFC '14)*, vol. 31, no. 6, pp. 1–3, San Francisco, Calif, USA, March 2014.

[16] R. Cannistra, B. Carle, R. Johnson et al., "Enabling Autonomic provisioning in SDN cloud networks with NFV service chaining," in *Proceedings of the Optical Fiber Communication Conference (OFC '14)*, pp. 1–3, Optical Society of America, San Francisco, Calif, USA, March 2014.

[17] R. Riggio, T. Rasheed, and F. Granelli, "EmPOWER: a testbed for network function virtualization research and experimentation," in *Proceedings of the Workshop on Software Defined Networks for Future Networks and Services (SDN4FNS '13)*, pp. 1–5, IEEE, Trento, Italy, November 2013.

[18] EU FED4FIRE, Federation for FIRE. Project reference: 318389. Funded under: FP7-ICT, http://www.fed4fire.eu/.

[19] EU OFELIA Project, "Open Flow in Europe: Linking Infrastructure and Applications," Project reference: 258365. Funded under: FP7-ICT, http://www.fp7-ofelia.eu/.

[20] EU BONFIRE Project, "Building service testbeds on FIRE," Project reference: 257386. Funded under: FP7-ICT, http://www.bonfire-project.eu/.

[21] EU FELIX-EU, FEderated Test-beds for Large-scale Infrastructure eXperiments Project reference: 608638. Funded under: FP7-ICT, http://www.ict-felix.eu/.

[22] EU SmartFIRE Project, Enabling SDN ExperiMentAtion in WiReless Testbeds exploiting Future Internet Infrastructure in South KoRea and Europe, Project reference: 611165. Funded under: FP7-ICT, http://eukorea-fire.eu/pilots/.

[23] EU CITYFLOW, "OpenFlow City Experiment—Linking Infrastructure and Applications," Project reference: 317576. Funded under: FP7-ICT, http://www.onesource.pt/cityflow/site/.

[24] EU ALIEN Project, Abstraction Layer for Implementation of Extensions in Programmable Networks Project reference: 317880. Funded under: FP7-ICT, 2014, http://www.fp7-alien.eu/.

[25] EU NetIDE Project, An Integrated Development Environment for Portable Network Applications. Project reference: 619543. Funded under: FP7-ICT, http://www.netide.eu/.

[26] EU NETVOLUTION Project, "Evolving Internet Routing: A Paradigm Shift to Foster Innovation," Project reference: 338402. Funded under: FP7-IDEAS-ERC, https://www.forth.gr/index_main.php?l=e&c=20&i=464.

[27] EU PRISTINE Project, "Programmability in RINA for European supremacy of virTualised NEtworks," Project reference: 619305. Funded under: FP7-ICT, http://ict-pristine.eu/.

[28] CloudNFV Project, http://www.cloudnfv.com/page3.html.

[29] MCN Project, http://www.mobile-cloud-networking.eu/site/.

[30] EU T-NOVA Project, "Network Functions as-a-Service over Virtualised Infrastructures," Project reference: 619520. Funded under: FP7-ICT, http://www.t-nova.eu/.

[31] EU UNIFY Project, "Unifying Cloud and Carrier Networks," Project reference: 619609. Funded under: FP7-ICT, http://www.fp7-unify.eu/.

[32] 3GPP SON, http://www.3gpp.org/technologies/keywords-acronyms/105-son.

[33] EU CROWD Project, "Connectivity Management for eneRgy Optimised Wireless Dense networks," Project reference: 318115. Funded under: FP7-ICT, http://www.ict-crowd.eu/.

[34] "ETSI NFV," 2015, http://www.etsi.org/technologies-clusters/technologies/nfv.

[35] ONF, 2015, https://www.opennetworking.org/about/onf-overview.

[36] TMForum ZOOM, https://www.tmforum.org/zoom.

[37] NETCONF Protocol, http://www.rfc-editor.org/rfc/rfc4741.txt.

[38] OpenFlow, https://www.opennetworking.org/sdn-resources/openflow/.

[39] "Open vSwitch," http://openvswitch.org/.

[40] L. J. García Villalba, Á. L. Valdivieso Caraguay, and L. I. Barona López, Eds., *SELFNET Deliverable 2.1—Use Cases Definition and Requirements of the System and Its Components*, Universidad Complutense de Madrid, 2015, https://selfnet-5g.eu/dissemination/.