Integrated NFV/SDN Architectures: A Systematic Literature Review

MICHEL S. BONFIM, KELVIN L. DIAS, and STENIO F. L. FERNANDES,
Universidade Federal de Pernambuco

Network Functions Virtualization (NFV) and Software-Defined Networking (SDN) are new paradigms in the move towards open software and network hardware. While NFV aims to virtualize network functions and deploy them into general purpose hardware, SDN makes networks programmable by separating the control and data planes. NFV and SDN are complementary technologies capable of providing one network solution. SDN can provide connectivity between Virtual Network Functions (VNFs) in a flexible and automated way, whereas NFV can use SDN as part of a service function chain. There are many studies designing NFV/SDN architectures in different environments. Researchers have been trying to address reliability, performance, and scalability problems using different architectural designs. This Systematic Literature Review (SLR) focuses on integrated NFV/SDN architectures, with the following goals: (i) to investigate and provide an in-depth review of the state of the art of NFV/SDN architectures, (ii) to synthesize their architectural designs, and (iii) to identify areas for further improvements. Broadly, this SLR will encourage researchers to advance the current stage of development (i.e., the state of the practice) of integrated NFV/SDN architectures and shed some light on future research efforts and the challenges faced.

CCS Concepts: • **General and reference** → **Surveys and overviews**; • **Networks** → **Network design principles**; **Network services**;

Additional Key Words and Phrases: Software-defined networking, network function virtualization, network virtualization, cloud computing, mobile networks, resource provisioning, autonomic management, service-level agreement, quality of service, resource scheduling, resource management, scalability, elasticity, reliability, security

ACM Reference format:
Michel S. Bonfim, Kelvin L. Dias, and Stenio F. L. Fernandes. 2019. Integrated NFV/SDN Architectures: A Systematic Literature Review. *ACM Comput. Surv.* 51, 6, Article 114 (February 2019), 39 pages.
https://doi.org/10.1145/3172866

1 INTRODUCTION

Proprietary network hardware equipment is everywhere in businesses, homes, and data center networks. Each vendor explores and exploits the maximum capability of its platforms as a way to meet the performance, reliability, and availability requirements demanded by the various types of users. However, such an approach has resulted in incompatibility between different

This work is supported by grants from UFPE and CNPq (304422/2013-4, 305223/2016-0).
Authors' addresses: M. S. Bonfim, K. L. Dias, and S. F. L. Fernandes (corresponding author), Centro de Informática (CIn), Universidade Federal de Pernambuco (UFPE), P.O. Box 7851, Cidade Universitária, 50740-560, Recife PE, Brazil; emails: {msb6, kld, stenio}@cin.ufpe.br.
Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
© 2019 Association for Computing Machinery.
0360-0300/2019/02-ART114 $15.00
https://doi.org/10.1145/3172866

manufacturer's technologies. Restricted licensing agreements and proprietary source code have further contributed to this limitation.

Because of such incompatibility of platforms, as well as the need for network engineers to add new features into their networks (e.g., firewalls, load balancing), they often need to purchase new equipment from different vendors. Each item of equipment is responsible for a share of the traffic processing that requires specific management strategies. Difficulties in the management and configuration of such heterogeneous environments are the norm. The requirement to allow such flexibility in configuration and the deployment of new network functions must be met in new business and engineering models. The complexities of current networking environments result in high operational (OPEX) and capital (CAPEX) expenditure costs [1].

Network Functions Virtualization (NFV) aims at solving these problems by transferring networking functions from vendor-specific and proprietary hardware appliances to software hosted on Common-Off-The-Shelf (COTS) systems (a.k.a. commodity platforms), i.e., with standard processing, memory, and storage components. These systems usually provide the network services in virtual machines (e.g., Virtual Network Functions, VNFs), each one performing different operations (e.g., firewall, packet inspection, routing, etc.) [2]. NFV has the potential to allow cost reduction and the increase in speed of network expansion. Also, NFV has the potential to increase network flexibility for fast service delivery, an option difficult to achieve with traditional methods [3].

Software-Defined Networking (SDN) is a new network paradigm. Its main feature is the separation of the network control plane from the data plane compared to current networks where the IP layer integrates both planes vertically into the network devices [4]. In the SDN control plane, represented by a software called SDN Controller, which is responsible for decisions on how to handle the underlying network traffic concerning network policies and rules. The SDN Controller can run on COTS systems, separated from the forwarding devices. The data plane, deployed as network devices, is responsible for forwarding data according to a set of rules. The SDN controller allows the creation and management of such rules through an Application Programming Interface (API) in the Northbound interface. It does have direct control over the data plane elements through protocols in the Southbound interface. Such a separation provides some definite advantages, such as simplification and flexibility in network policy enforcement, facilitating network configuration, development, and fostering innovation [5]. It also brings research and development challenges that have attracted researchers from both industry and academia.

According to Mijumbi et al. (2016) [6], "NFV and SDN have a lot in common, since they both advocate for a passage toward open software and network hardware." Even with different purposes, NFV and SDN do indeed represent complementary paradigms and technologies capable of providing one consolidated solution. To this end, SDN can provide connectivity between VNFs in a flexible and automated way, thus simplifying network management. However, NFV can make use of SDN as part of a Service Function Chaining (SFC). In this case, both SDN Controllers and Management Applications can run as VNFs in a scalable environment and hence benefit from essential features, such as availability, reliability, and elasticity.

Some studies are tackling the integration of NFV and SDN in different environments (e.g., Cloud Computing, Wide Area Network, Customer Premise Equipment, 5G, etc.). Industrial and academic research studies address several challenges, such as reliability, overall performance, and scalability. Those studies use distinct architectural design rationale and functional and non-functional requirements. Although NFV/SDN architectures have clear potential benefits, they are still at an early stage of development. There are several open research questions to be answered and development issues to be addressed [5–7].

It is worth emphasizing that there have been some initial efforts to review the body of knowledge on NFV and SDN, but most efforts treat them in isolation. Mijumbi et al. (2016) [6] and Gil and

Botero (2016) [8] surveyed the state of the art in NFV, whereas Kreutz et al. (2015) [5] presented a survey exclusively on SDN. Furthermore, both Li and Chen (2015) [9] and López et al. (2015) [10] propose studies to integrate both technologies. However, there is still a need for a detailed vision of the different integrated NFV/SDN architectures (e.g., target environment, problems to solve, and architectural designs) as well as trends for research and development. We argue that the research community would benefit from an in-depth view of the NFV/SDN architectural designs so that researchers can have a clear picture of the past relevant studies as well as the current challenges. Therefore, our study fills an important gap by providing an in-depth view of the SDN/NFV architectures, as well as highlighting the challenges to further advances in this topic.

In this work, we cover the state of the art of integrated NFV/SDN architectures. We aim at (i) investigating the characteristics (target environment and problems to solve) of integrated NFV/SDN solutions and current practices, (ii) comparing their architectural designs (i.e., NFV framework design and tools, SDN APIs, and placement of SDN elements), and (iii) identifying the challenges and the possibilities for improving them. To this end, we conducted a Systematic Literature Review (SLR) to provide an overview of this research area, based on a well-known methodological framework introduced by Kitchenham et al. (2009) [11].

SLR is an evidence-based approach used to identify, evaluate, and interpret all available evidence about a focused topic, in a repeatable and impartial manner [12]. For this, the SLR framework follows a predefined protocol with a set of steps to perform sources and studies selection and data extraction. In the end, results are synthesized from this well-defined approach by comparing the individual studies and providing consistent evidence of the research questions being posed.

Our original contributions are threefold. First, this SLR provides an in-depth understanding of both state of the art and state of the practice of NFV/SDN architectural solutions, highlighting their main characteristics (e.g., potential deployment scenarios and problems raised) and their underlying architectural designs. Second, the SLR study identifies trends for future research and development as well as open research issues and challenges. Last, but not least, our SLR provides the necessary details for replicating it or broadening its scope in the future.

The remainder of the article is organized as follows. Section 2 describes NFV and SDN technologies. Section 3 introduces the details of the adopted SLR. It explicitly defines the steps of the protocol and the strategies to retrieve the evidence, to allow this SLR to be reproduced and criticized by other professionals. Section 4 describes the search process that resulted from the SLR execution. Section 5 describes the target environments and problems addressed by the studies. Section 6 describes a taxonomy to organize the various decision-making levels for the design of NFV/SDN architectures. Such a taxonomy was derived by analyzing implementations found in the researched literature and reference architectures proposed by vendors and standardization bodies. Sections 7 and 8 provide the technical details of this taxonomy. Section 9 presents a brief description of the results obtained. Section 10 lists the challenges involved in developing NFV/SDN solutions. Finally, Section 11 concludes the article.

2 BACKGROUND

2.1 Network Functions Virtualization

NFV is transforming the computer and communication networks industry. NFV allows customers to transfer the networking functions from vendor-specific and proprietary hardware appliances to software hosted on COTS platforms [1].

NFV provides the network services in virtual machines (VMs) working in Cloud infrastructures, where each VM performs different network operations (e.g., firewall, intrusion detection, Deep

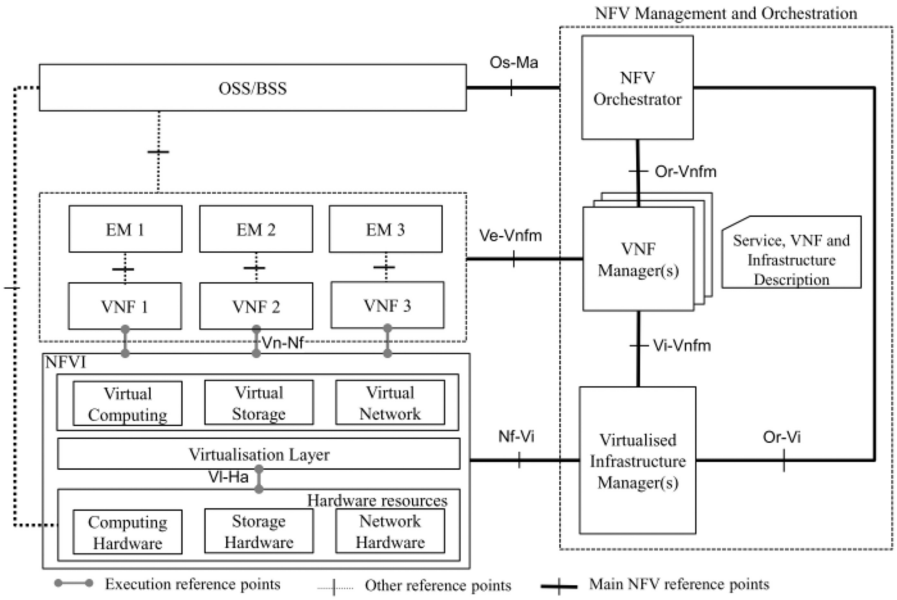

Fig. 1. NFV architecture [13].

Packet Inspection, load balancing, etc.) [2]. Some benefits of deploying network services as virtual functions are [1]:

- Flexibility in the allocation of network functions in general-purpose hardware;
- Rapid implementation and deployment of new network services;
- Support of multiple versions of service and multi-tenancy scenarios;
- Reduction in CAPEX costs by managing energy usage efficiently;
- Automation of the operational processes, thus improving efficiency and reducing OPEX costs.

From 2012, the European Telecommunications Standards Institute (ETSI) has led the standardization process for NFV technology through the NFV Industry Specification Group (NFV ISG). The NFV ISG has already published tens of specifications documents, such as requirements, use cases, terminologies, proofs of concept, and the like [13]. These specifications allow researchers and engineers to have a clear picture of the elements of a particular NFV infrastructure.

Figure 1 illustrates the high-level architecture for NFV, which comprises three main functional blocks, as detailed below.

Virtual Network Functions (VNFs): VNF is the virtualization of a certain network function, which should operate independently of the others. It may run on one or more virtual machines. A particular VNF can also be divided into several sub-functions called VNF Components (VNFCs). Elemental Management Systems (EMSs) can be used for VNF monitoring;

NFV Infrastructure (NFVI): NFVI comprises all hardware and software required to deploy, operate, and monitor VNFs. To this end, NFVI has a virtualization layer necessary for abstracting the hardware resources (processing, storage, and network connectivity). It ensures the independence of the VNF software from the physical resources. The virtualization layer is usually composed of the server (e.g., Xen, KVM, VMware, etc.) and the

network (e.g., VXLANs, NVGRE, OpenFlow, etc.) hypervisors. The NFVI Point of Presence (NFVI-PoP) defines a location for Network Function deployments as one or many VNFs.

NFV Management and Orchestration (MANO): MANO comprises three components: (i) The Virtualized Infrastructure Manager (VIM), which manages and controls the interaction of VNFs with physical resources under its control (e.g., allocation, deallocation, and inventory); (ii) the VNF Manager (VNFM), which is responsible for managing the VNF life-cycle (e.g., initialization, suspension, and termination); and (iii) the NFV Orchestrator (NFVO), which is responsible for realizing network services on NFVI. It also performs monitoring operations of the NFVI as a way to collect information for operations and performance management.

Another component to be considered as part of the NFV framework is the Operations Support Systems and Business Support Systems (OSS/BSS). This element comprises the legacy management systems and assists MANO in the execution of network policies, either automatically or manually.

2.2 Software-Defined Networking (SDN)

SDN is a new network paradigm that was designed to overcome the difficulty in developing and testing new solutions and protocols in production environments, where the underlying code running in business switches and routers are proprietary and closed [4].

According to Kreutz et al. (2015), currently, both control and data planes are integrated into most commercial networking devices, which makes IP networks difficult to manage. Due to this, operators need to configure network policies into each device individually, often using low-level commands that are specific to the manufacturer. Further, automatic reconfiguration mechanisms, necessary for network adaptation during failures and load changes, are non-existent in today's networks. Such issues reduce the flexibility for deploying new network services and management strategies, as well as hindering development and innovation.

The main feature of the SDN paradigm is the separation of the control and data planes. It has clear advantages where network programmability is achieved through the centralization of the control plane in conjunction with the availability of open APIs, thus making easier the process of creating and deploying new network applications. SDN provides simplification and flexibility in network policy enforcement, facilitating network configuration and management [5].

The control plane, represented by a software called the SDN Controller, is responsible for decisions on how to handle network traffic, assuming the role of the "brain" of the network. The SDN Controller can run on COTS platforms, separated from the network equipment. The data plane, represented by the network devices, is responsible for forwarding traffic according to a set of rules [5]. Such rules are created at and managed by the SDN Controller. The SDN controller has a global view of the network topology and has direct control over the data plane elements through a southbound protocol, such as OpenFlow [14].

Figure 2 shows the given three layers of an SDN architecture and the APIs responsible for the interaction between them. The SDN Northbound API is responsible for providing support for communication between the application layer and the control plane layer. It also includes support for SDN Applications, such as traffic engineering, routing, firewall, quality of service, and so on. The Southbound API is responsible for the communication between the SDN Controllers and switches.

3 SYSTEMATIC LITERATURE REVIEW PLANNING

This section presents the adopted plan to perform the SLR. This phase aims to define the way the review is executed, including the research questions and the procedure for sources and studies selection.

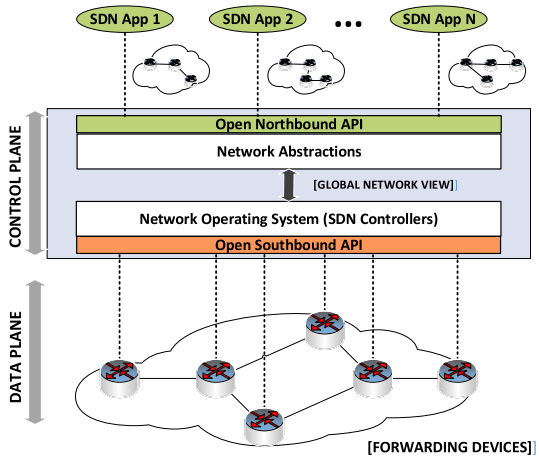

Fig. 2. SDN reference architecture [4, 5, 14].

3.1 Research Questions

To identify the state of the art of integrated NFV/SDN solutions and open issues, we initially focus our study on the following research questions.

- **Q1) In which environments are the integrated NFV/SDN solutions applied?**
  This question aims at mapping the actual environments (e.g., Enterprise Networks, WANs, CPEs, Data Centers, Wireless Networks, etc.) in which the proposed integrated NFV/SDN solutions have been tested and deployed.
- **Q2) What are the problems that such integrated NFV/SDN solutions are trying to solve?**
  This question aims at identifying the issues (e.g., middlebox and network virtualization, vCPE, reliability, scalability, dynamic service function chaining, performance, etc.) that NFV/SDN solutions are trying to tackle. We also aim to classify them according to their respective target environments.
- **Q3) What are the differences among the design architectures of integrated NFV/SDN solutions?**
  This question seeks clarification on how the studies have been proposing design architectures of integrated NFV/SDN solutions. Answering this question requires a classification of the proposed architectures using a subset of their characteristics extracted from the ETSI documentation [15], published in December 2015. These characteristics include NFV framework design and tools, SDN Northbound and Southbound APIs, the placement of the SDN elements in the NFV Framework, the use of multiple SDN controllers, and the like. We aim at identifying the differences among such designs based on the types of target environments and general objectives, including their main advantages and disadvantages.

3.2 Sources Selection

To find the relevant evidence to answer the research questions, a set of sources must be selected to perform the search of primary studies. We now describe the criteria used to select such sources, the search strings, and the sources identification.

For the selection criteria of sources, we considered the availability of articles on the Web and the existence of advanced search mechanisms using keyword and filters based on content type

Table 1. List of Terms and Synonyms

	Group 1	Group 2
Term 1	Software-Defined Networking	Network Function Virtualisation
Term 2	Software Defined Networking	Network Function Virtualization
Term 3	Software Defined Network	NFV
Term 4	Software-Defined Network	
Term 5	SDN	

(conference publications, journals, and magazines, etc.) and year of publication. We considered only studies in the English language. Therefore, we selected the following web search engines: ACM Digital Library, Engineering Village, IEEE Xplore, Science Direct, Scopus, Springer Links, and Web of Science. The literature searches were performed manually using all the selected web search engines.

To compose our search string, we considered the keywords listed in Table 1, where each group represents a keyword with its synonyms. The general form of the search string is shown as follows:

Search String: (([G1,T1] **OR** [G1,T2] **OR** [G1,T3] **OR** [G1,T4] **OR** [G1,T5]) **AND** ([G2,T1] **OR** [G2,T2] **OR** [G2,T3]))

3.3 Procedure for Studies Selection

A priori, all studies in the English language obtained from web search engines were selected as primary studies. These primary studies then went through a studies selection and evaluation process, based on three stages. One researcher (M. Bonfim) was assigned to evaluate the selected studies. An article is included for further processing in the next steps when it is approved in the previous one. Otherwise, the article is discarded.

Below, we describe the three stages of the studies selection and evaluation process:

- **Stage 1:** Eliminate studies selected as primary studies based on exclusion criteria. An article will be only included in the following stages if it proposes an integrated NFV/SDN solution. This stage considers only information provided in abstract and conclusion.
- **Stage 2:** Eliminate studies selected in Stage 1 based on exclusion criteria. An article will be only included for the following stages if it proposes an integrated NFV/SDN solution and describes its architecture design. This stage evaluates all content of the articles.
- **Stage 3:** In this stage, studies selected at Stage 2 pass for a quality screening. An article will be excluded if it does not meet the following quality criteria:
  - **QC1:** Is there a clear statement of the goals (i.e., target environments and problems to solve) of the research?
  - **QC2:** Is the architecture design well detailed? In other words, is it possible identify the used tools, the place of SDN elements and NFV framework design?
  - **QC3:** Are the experiments realized to evaluate the ideas presented in the study?

Each criterion has three possible responses: Yes, Partly, or No. "Yes" responses count for 1 (one) point, "Partly" count for 0.5 points, and "No" count for 0 (zero) points. To be accepted, a article must obtain a score of greater or equal to 2 (two) as described in Equation (1):

$$QC1 + QC2 + QC3 \geq 2.0. \tag{1}$$

Finally, at the end of execution, we included some reports from Proof of Concepts (PoCs) registered in ETSI NFV ISG PoC Projects,[1] regarding NFV/SDN solutions provided by different vendors and carrier networks.

4 SEARCH RESULTS

The initial search was performed in April 2016. Initially, a total of 1,644 articles were identified (Identity Phase) as primary studies. In the Identity Phase, 907 duplicate findings were removed from the result set. It is important to emphasize that we do not delimit a specific range of years for the searching process. Then we started the execution of the three stages of selection, as described in Subsection 3.3.

In Stage 1 (Screening Phase), having reviewed all abstracts and conclusions, we considered only articles that proposed an integrated NFV/SDN solution. In this case, we selected 138 studies and discarded 769. In Stage 2 (Screening Phase), we considered only records included in Stage 1. After evaluating all the content of articles, we considered only those that described the design of the integrated NFV/SDN solution. In this case, we selected 88 studies and discarded 50. In Stage 3 (Eligibility Phase), studies selected in Stage 2 were passed for a quality screening, described in Subsection 3.3. In this stage, we discarded 40 studies that did not meet the quality criteria, with the result that 48 studies were included (Included Phase) for data extraction.

At the end of the execution process, 12 Proof of Concepts (PoCs) reports (registered in ETSI NFV ISG PoC Projects[2]) are included, generating a total of 60 studies for data extraction. The PoCs are part of the Hot Topic 01,[3] "Use of SDN in an NFV architectural framework", which includes SDN/NFV solutions provided by different vendors and carrier networks.

The following step is the data collection for each selected work in the Included Phase. One (1) researcher (M. Bonfim) performed the data extraction, extracting the following properties from each study: (i) author's identification, (ii) article type (conference article, journal article, report, etc.), (iii) publication description (title, ISSN, date, DOI, etc.), (iv) work's title and abstract, (v) environments in which the NFV/SDN solution is applied, (vi) problems that NFV/SDN solution try to solve, (vii) technical aspects related to NFV/SDN solution proposed, and (viii) quality criteria evaluation.

After performing the SLR, during the article's revision, another 14 relevant references were found and/or recommended by experts in the field to complement the SLR findings. The total number of research articles is now 74, including new references from 2016 until 2017. These articles respected the same Exclusion (Stage 2), and Quality Criteria (Stage 3) adopted previously. All results and discussions presented in this article were derived from these 74 studies.

There is a growing interest from both the academia and the industry in integrating NFV and SDN technologies. The number of studies increased from 1 research article in 2013 to 43 in 2015, and this number has been rising since then. The researchers prefer scientific conferences to publish their studies (35 articles), followed by journals (26 articles) and PoC reports (12 articles).

The reader can find more details regarding the data collection (extracting documents) at the GitHub link.[4]

5 APPLICATION FOR NFV/SDN ARCHITECTURES

This section aims at answering the first and second research questions, thus relating the different environments where the NFV/SDN architectures found are applied, in addition to identifying the main problems those architectures are trying to solve.

[1] http://www.etsi.org/technologies-clusters/technologies/nfv/nfv-poc.
[2] http://www.etsi.org/technologies-clusters/technologies/nfv/nfv-poc.
[3] http://nfvwiki.etsi.org/index.php?title=HT01_-_Use_of_SDN_in_an_NFV_architectural_framework.
[4] Link for data collection archives: https://github.com/michelsb/SLRNFVSDNFiles.git.

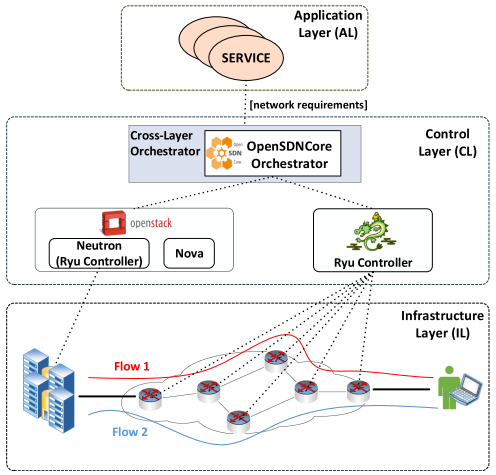

Fig. 3. NFV/SDN architecture for application-specific traffic steering.

5.1 On-demand and Application-specific Traffic Steering

Traffic Steering is the ability to direct users' requests to the appropriate service/content sources. Traffic steering might be based on many factors such as the available networking resources and capabilities on the client and server side, user's permissions and location, and the like. For example, a certain user may request a video streaming service that has stringent application performance requirements. In this case, on-demand and application-specific traffic steering could guarantee efficient network resource usage and better Quality of Experience (QoE) for the user.

In the NFV Framework context, SDN can enhance traffic steering between VNFs, providing dynamic service chaining. With the separation of control and data planes, SDN enables the exchange of information between the application and network layers, allowing users' services to have an overview of the general state of the network and to make intelligent decisions (to meet service requirements) on how to steer traffic through VNFs better.

Carella et al. (2015) [16] proposed an NFV/SDN architecture to provide a cross-layer interface between the application and network layers, to allow the deployment of network services with on-demand and application-specific traffic steering. Figure 3 shows our detailed view of this architecture. It comprises three layers: Application, Control, and Infrastructure. The Application Layer consists of network services. These services must interface with the Control Plane API to communicate their network requirements (e.g., bandwidth, maximum latency). The Control Plane has a global view of computer and network resources and provides the traffic steering capabilities to the Application Layer. Its main component is the Cross-Layer Orchestrator (CLO) that acts as an NFVO and VNFM to manage the lifecycle of services over the cloud (OpenStack-based) and WAN domains (OpenFlow-based). The CLO was implemented over the OpenSDNCore Orchestrator [17], using Java programming language.

5.2 Middleboxes Virtualization

According to Reference [18], middleboxes have been increasingly used in enterprise networks (45% of network devices). They are deployed to increase performance (e.g., traffic shaping, load balancing, and TCP optimization) and to provide security functionalities (e.g., firewalls, Intrusion Detection and Prevention systems (IDPS), and Deep Packet Inspection (DPI)) for both incoming and outcoming traffic.

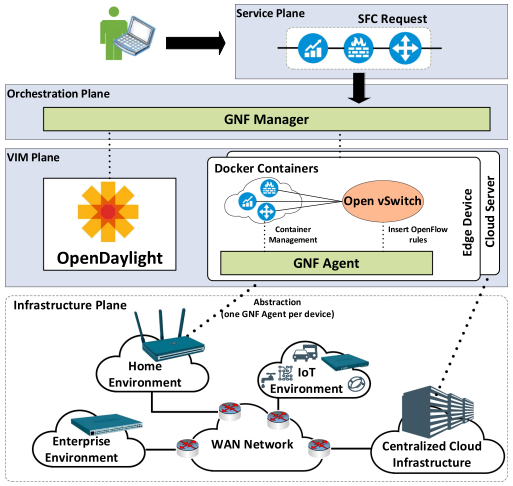

Fig. 4. The GNF platform [29].

However, hardware-based middleboxes have the following disadvantages [19]. First, they incur high operational costs (OPEX) due to the management complexity. These middleboxes come from different manufacturers and must be deployed, configured, and managed individually. Second, hardware-based middleboxes incur capital costs (CAPEX). When new network functions are necessary, enterprises must purchase one or more middleboxes due to the inflexibility in proprietary hardware that creates vendor lock-in and limits innovation.

NFV/SDN architectures can be used to deal with these challenges. The primary objectives are to reduce both CAPEX and OPEX and to provide fast delivery of network function, elasticity, and dynamic service chaining. In these works, NFV manages virtual middleboxes, while SDN provides interconnection between VNFs to delivery Service Function Chaining (network services).

In this context, several NFV/SDN architectures [19–29] have been proposed to deal with Middleboxes Virtualization. Some of them will be presented below.

The works of Cziva et al. (2015) proposed an NFV/SDN framework,[5] so-called Glasgow Network Functions (GNF), to deploy and manage container-based network services in public [26] and private [19, 29] Cloud environments. This framework aims at overcoming the limited network reconfigurability in these scenarios, delivering network programmability and fast deployment of new network services. As shown in Figure 4, GNF is composed of four planes: The infrastructure encompasses all the physical resources of network and computations, where we have only NFV Centralized Cloud Infrastructures in Reference [19], and incorporated with edge devices (e.g., CPEs, virtual routers, and IoT gateways) in Reference [29]. The VIM and Orchestration planes are responsible for Resource Orchestration. For this, the operator must deploy the GNF Agent on all Cloud servers and all edge devices. This feature has two functions: (i) local VNF instantiation by using Docker Engine [30] for fast deployments and low resource utilization and (ii) local traffic steering management by using OpenFlow rules (via OVSDB) and virtual switches. Also, GNF uses the OpenDaylight controller for network connectivity in the NFVI. The GNF Manager is responsible for receiving NFV service requests (Service plane) and performing the necessary operations using the OpenDaylight and GNF Agent instances.

[5] https://netlab.dcs.gla.ac.uk/projects/glasgow-network-functions.

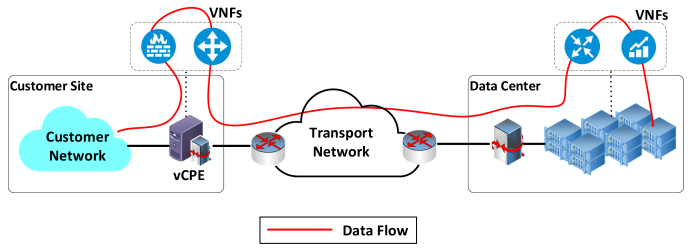

Fig. 5. NFV/SDN architecture for virtualized customers premises equipment (vCPE).

In addition, Sonkoly et al. (2015) [24] extended the UNIFY architecture to create a service function chaining control plane. This solution aims to support SFC in distributed cloud scenarios, where VNFs from the same SFC can run in different NFVI-PoPs. A prototype framework, called Extensible Service Chain Prototyping Environment (ESCAPE), was implemented in Python on top of a POX Controller. This prototype works with two domains in the Infrastructure Layer (IL): Cloud and OpenFlow. The OpenStack Cloud Platform [31] and the OpenDaylight Controller [32] perform the management of Cloud domains while VNFs are deployed as KVM virtual machines (running a Click process). The OpenFlow domains handle transport networks with Linux nodes running Open vSwitches (OpenFlow support). The POX Controller [33] (network management) and the NETCONF/YANG (VNF management) manage these domains while VNFs are deployed as distinct processes (Linux cgroups) and run network functions implemented in Click Modular Router.

Finally, Deng et al. (2015) [25] proposed the VNGuard framework that uses NFV to provide fast and dynamic virtual firewalls in a cloud environment, for the protection of Virtual Networks. Aimed at considering VN's changeable topology (VMs dispersion and migration), this framework uses SDN to provide fast and flexible traffic steering to virtual firewalls. They used OpenStack as a VIM and ClickOS for VNF development. CloudLab[6] is a testbed that provides an Infrastructure as a Service (IaaS) for cloud-based experiments.

5.3 Virtualized Customers Premises Equipment (vCPE)

Customers Premises Equipment (CPE) means any equipment (router, modem, etc.) within the customer domain that receives a communication service. CPEs have been a barrier to the current goals of both telecommunications companies and service providers, due to the high cost of maintenance, management difficulties, and the impossibility of remote upgrades. As an alternative, a solution is the CPE virtualization using an NFV architecture, also known as Virtualized CPE (vCPE) [34].

According to an IHS Markit Survey,[7] published in 2016, 100% of consulted service providers said they intend to deploy NFV at some point. 81% expect to roll out this deployment by 2017. Most service providers (more than 80%) have a preference for deploying vCPE.

vCPE is a service in which some or all of the functions associated with CPE are virtualized [34]. One of the main problems related to CPEs virtualization is how to instantiate network services in distributed infrastructures (using multiple NFVI-PoPs) [35]. In this type of scenario (see Figure 5), also called Distributed NFV [36], the VNFs are placed either in the service provider Cloud platform (Cloud CPE) or the on-premise CPE, depending on where they are most efficient regarding latency, available resources, and so on. SDN has been the technology adopted to implement the communication management of different scenarios (e.g., Cloud, CPE, and WAN) to provide Distributed NFV.

[6] https://www.cloudlab.us/.
[7] NFV Strategies Service Provider Survey - https://technology.ihs.com/572348/nfv-strategies-service-provider-survey-2016.

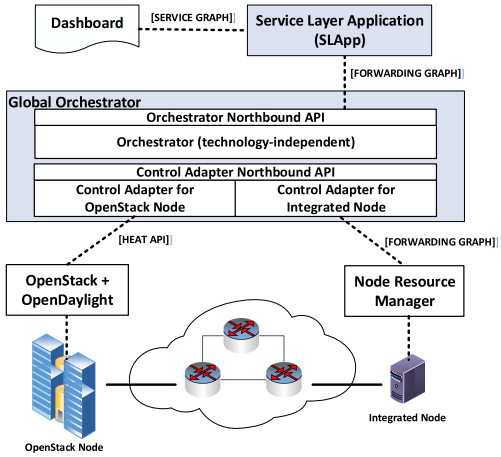

Fig. 6. An NFV/SDN architecture design for vCPE.

Cerrato et al. (2015) [35] proposed a service-oriented NFV/SDN architecture for Telco networks that delivers generic network services selected by telecom operators (DHCP and NAT) or end users (BitTorrent client). The deployment of these network services can occur in a distributed manner either in the telecom data center or the CPE. Figure 6 shows our simplified view of this architecture.

This solution is based on the UNIFY architecture , including its three layers. The Service Layer Application (SLApp) represents the UNIFY SL enabling different players (operators and end users) to select their network services. For this, the SLApp includes an authentication mechanism and provides a high-level data model for defining flexible network services (including traffic steering primitives), called Service Graph (SG).

Also, the Global Orchestrator (GO) represents the UNIFY Orchestration Layer (OL). The GO manipulates the Forwarding Graph (FG) received from SLApp to enable the network service deployment according to the VNF requirements and infrastructure capabilities. To allow distributed NFV, the GO implements multiple Control Adaptors to coordinate different infrastructures and an Orchestrator component responsible for the centralized coordination of multiple Control Adaptors. Then, the GO selects one of the infrastructures to implement all the network service requested. The authors proposed two different infrastructures (UNIFY Infrastructure Layer, IL) to host network services: the integrated node and the OpenStack node.

The integrated node represents the CPE (home gateway). It receives an FG from the GO through the Node Resource Manager (NRM) via REST API. The NRM will instantiate all VNFs using Docker containers, DPDK process or any hypervisor supported by *libvirt*. For traffic steering, the NRM uses an extensible Data-Path daemon (xDPd) to create an OpenFlow switch (and its correspondent Controller) for each FG. Separately, the OpenStack node represents the telco data center and uses the OpenStack Cloud Platform for network service deployment. In this case, the KVM hypervisor creates the VNFs, and the OpenDaylight and Open vSwitch control the traffic steering.

The works of Soares [37, 38] proposed the Cloud4NFV platform, an NFV/SDN framework for Telco network virtualization. This platform considers multiple NFVI-PoPs and WAN domains when deploying new Service Function Chaining. Cloud4NFV considers a topology with multiples customer sites (NFVI-PoPs). All NFVI-PoPs include an OpenStack distribution working as a Cloud VIM and an OpenDaylight controller to provide VNF connectivity. The VNFs are CPE functions, and they are deployed as VMs in the NFVI-PoP closest to the customer. Further, Cloud4NFV

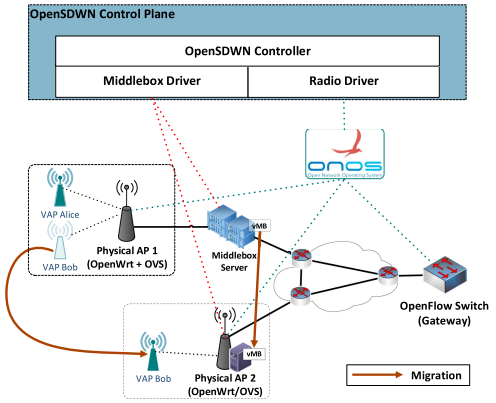

Fig. 7. NFV/SDN architecture for WiFi networks.

includes a WAN VIM to provide a view of a unified WAN domain connecting the NFVI-PoPs and Telco Data Center.

5.4 Wireless Networks

Due to the increasing popularity of wireless networks, new requirements have arisen, such as mobility support, programmability, fast delivery of network services, performance, and security [22]. However, the management and configuration of today's large WiFi networks are complex and inflexible, ignoring the application requirements or user needs. We present the problems addressed by NFV/SDN architectures designed for different wireless networks scenarios.

Regarding a WiFi network, the studies related to WLANs leverage the Virtual Access Point (VAP) abstraction by moving the MAC layer or middleboxes processing to the Cloud. When associated with the wireless network, each client acquires a VAP that will be dedicated, independent of the client migrating from one access point to another (handover), as shown in Figure 7. In this example, the Physical AP 1 allocates two VAPs to two clients, Bob and Alice. If Bob moves toward another AP, then his VAP will also be migrated and deployed to the new AP.

Shulz-Zander et al. (2015) [22, 39] proposed the OpenSDWN, an NFV/SDN approach to implement per client access points and virtual middleboxes. Figure 7 shows our detailed view of this architecture. For the access point case, the authors created an extension to Odin [40], called Light Virtual Access Point (LVAP). An LVAP uses SDN applications to abstract some functionality of the 802.11 Access Point, such as authentication, handoff, and client associations. A physical AP supports multiple LVAPs, one for each client (which receives a unique BSSID). Therefore, a certain LVAP serves as a dedicated link between its client and infrastructure. The authors also implemented virtual middleboxes (e.g., firewall), which can be deployed either on a Middlebox Server or at the access point itself and can integrate them with LVAPs using virtual networks. A service differentiation mechanism (DPI-based) tries to identify and classify flows to redirect traffic to the correct vMiddleboxes (vMB). OpenSDWN Controller performs all the above functionalities, using the Floodlight [22] and ONOS [39] as SDN controllers. Such an abstraction allows the seamless mobility with the migration of both LVAPs and vMBs among APs.

In Reference [41], the authors extended CloudMAC framework [42] to provide QoS for VAPs. In the proposed solution, the VAP is responsible for the MAC layer management frames (e.g., beacons, probes request/response) and runs as a VM in a Cloud environment. The physical AP redirects these frames to the destination VAP, using OpenFlow rules. The QoS mechanism implements VAP traffic prioritization using different queue management strategies (e.g., Stochastic Fair Queueing)

on all Open vSwitches between the APs and VAPs. The OpenDaylight Controller manages both traffic redirection and prioritization. Seamless handovers can be achieved by just changing the SDN forwarding rules.

5.5 Wireless and Mobile Networks

A new generation of mobile network technologies appears every 10 years. The first generation (1G) came in the mid-1980s with analog cellular networks. The second generation (2G) began in the mid-1990s and started the era of digital mobile phones encompassing technologies such as CDMA, TDMA, GSM, GPRS, and EDGE. The third-generation (3G) emerged in the late 1990s introducing the use of packet switching rather than circuit switching for data transmission. 3G technologies such as the Universal Mobile Telecommunications System (UMTS) achieves high connection speeds (up to 42Mbit/s downlink), making it possible to use multimedia applications. The fourth generation (4G) with its Long Term Evolution (LTE) appeared in the mid-2010s with the aim of providing speed improvements up to 10-fold over the existing 3G technologies.

It is worth emphasizing that every new generation tries to address service and network requirements not met by its predecessors. One of the current challenges for mobile networks is how to handle the ever-increasing traffic volume. To address this growth, mobile operators are investing in infrastructure, thus increasing OPEX/CAPEX costs and management complexity. In this context, 5G networks aim at addressing the following demands [43]: improved data rate, decreased latency, and increased capacity for consistent QoS/QoE.

Currently, the standardization of 5G networks is still a work-in-progress [44]. The International Telecommunications Union (ITU[8]) will be the specialized agency responsible for publishing the final standard in mid-2020s, which is also referenced as International Mobile Telecommunications (IMT)-2020. The 3rd Generation Partnership Project (3GPP[9]) is the standard body that unites several mobile industries with the objective of elaborating and submitting a proposed specification to the ITU (mid-2018's) to be part of the IMT-2020 standard.

Both industrial and academic researchers have also put research efforts on the architectural components of 5G networks. For instance, Verizon created a 5G Tech Forum (5GTF) in September 2015 [45]. The 5GTF is a vital initiative where major vendors such as Verizon, Cisco, Ericsson, Nokia, and Apple work together to develop early 5G specifications and then contribute to the 3GPP. 5GTF published its first specification release in July 2016.

Also, the European Union funded 5G Public-Private Partnership (5GPPP). 5GPPP is a joint initiative between the European Commission (EC) and the European ICT industry to develop solutions, architectures, and standards to put Europe in the leadership position for the 5G networks [46]. The 5GPPP has been supporting different projects in its first[10] and second[11] phases such as METIS and SELFNET. Those projects focus on research topics ranging from physical infrastructure to overall architecture, virtualization, network management, and software networks. As a result, 5GPPP has published several specifications, including a view on the 5G architecture [47].

A 5G infrastructure must provide features that support different types of vertical business such as Automotive (e.g., car manufacturers), eHealth (e.g., health industry), Energy (e.g., power companies), Factories (e.g., IoT technology providers), Media & Entertainment (e.g., content providers) [47]. All of those markets encompass different types of use cases (e.g., automated driving, robotics for remote surgery, on-site live event experience, etc.) that have their characteristics (e.g., data

[8]ITU website: http://www.itu.int.
[9]3GPP website: www.3gpp.org/.
[10]5G PPP Phase I Projects - https://5g-ppp.eu/5g-ppp-phase-1-projects/.
[11]5G PPP Phase II Projects - https://5g-ppp.eu/5g-ppp-phase-2-projects/.

traffic patterns, mobility support, etc.) and requirements (e.g., throughput, latency, etc.). An exhaustive list of case studies for 5G can be found in Reference [48].

To support such heterogeneity in the use cases as well as to meet the performance requirements, new 5G technologies will impact the entire mobile network including mobile devices; radio access, transport, and core networks; and the cloud (local, regional, or global). A 5G architecture should enable speed, agility, and cost-efficiency when delivering new services such as those in the context of the Internet of Things (IoT) and Smart Cities. 5G networks should also provide multi-tenancy, multi-service, and multi-domain support. To this end, the infrastructure providers must allocate logical networks (accessible by northbound APIs), so-called network slices (Network Slice layer), or for mobile operators or service providers, who in turn can create their own slices or services (Software Network Service Chain and Service layers). To build logical networks, the infrastructure providers will have to deploy end-to-end resource, infrastructure (Resource Abstraction and Virtualization layer), and service orchestration functions to reserve appropriate computing and network resources from different administrative domains, keeping QoS tailored to user demand [47].

Finally, in the Radio Access Network (RAN), 5G architecture should operate in a broad spectrum range with a diverse variety of characteristics, provide efficient transmission and data processing, support the coexistence of different radio access technologies (5G, LTE, and Wi-Fi) and be energy efficient. In this case, techniques such as Mobile Edge Computing (MEC) can be used as it allows pushing the services to the RAN with the objective of meeting the ultra-low latency and higher-speed requirements [47, 49].

The key enablers to achieve these functions are virtualization, softwarization, and programmability features, which can deliver the suitable level of flexibility in 5G networks. The use of NFV and SDN technologies will play a significant role in 5G networks, since they allow the network programmability and the fast delivery of new services, enabling network slicing and MEC implementation and orchestration [50]. In this context, several NFV/SDN architectures have been designed to overcome most of these challenges. We highlight some prospective solutions in the following sub-sections.

5.5.1 Mobile Network Function Virtualization. These architectures use cloud computing to assist mobile network virtualization. The goal is to provide a virtualization and communication platform for mobile network services as a way to deliver a flexible and scalable environment.

Regarding 3G services, [51] proposed the Software Defined Transitional Networking (SDTN), an NFV/SDN architecture to support legacy service integration in 4G networks. The authors assumed LTE as the underlying network (SDTN data plane). 3G functions are VNF instances that replace the following components: Serving General Packet Radio Service (GPRS) Support Node (SGSN), Gateway GPRS Support Node (GGSN), and Home Location Register (HLR). The Edge Controller acts as an SDN Controller mapping the actions performed by the virtualized 3G functions to the physical 4G network (using the 4G forwarding control plane). The Edge Controller coordinates the redirection of network services traffic to VNFs by programming edge switches (using OpenFlow).

When considering 4G services, most of the studies that focus on the mobile core network include the virtualization of Evolved Packet Core (EPC), such as: Serving Gateway (SGW) [52–60], Packet Data Network Gateway (PGW) [52–61], Mobility Management Entity (MME) [52–59], Home Subscriber Server (HSS) [52, 57, 59], and Policy and Charging Rules Function (PCRF) [57, 59].

In References [52] and [57], the authors proposed a PoC to evaluate the virtualization of EPC components (vSGW, vPGW, and vMME) as VNFs over an NFV/SDN architecture. The testbed comprised eNodeBs (emulator or eNodeB model Flexi Zone from Nokia Networks) interconnected with a Cloud data center through OpenFlow switches with MPLS support (Coriant Oy 8615 Smart Router). All EPC VNFs run in a Cloud environment and were implemented using the following

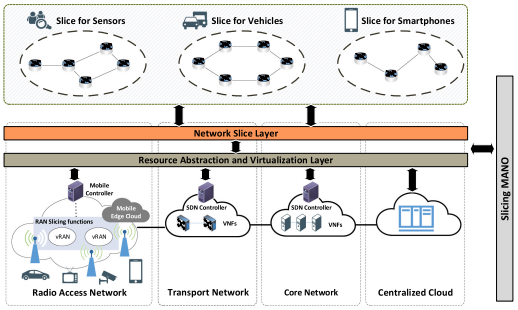

Fig. 8. Conceptual illustration of network slicing.

tools: eMME SW module (Aalto University) for vMME, open source nwEPC[12] for vSGW and vPGw, and an SQL database fo vHSS. A Ryu Controller coordinates the OpenFlow switches providing NFVI connectivity, eNodeB and VNFs interconnection and QoS support using MPLS tagging.

In addition, several studies proposed NFV/SDN architectures for virtualization of SGi-LAN services [20, 62, 63, 64]. Serving Gateway interface (SGi) interconnects mobile packet core and external IP networks. In a 4G network, SGi runs between PGW and a Packet Data Network (PDN) being responsible for ensuring the intercommunication performance and reliability. For this, SGi encompasses some services, such as Deep Packet Inspection (DPI), firewall, NAT, TCP optimization, and several caches. Gronsund et al. (2015) [64] proposed to replace the SGi elements for a physical OpenFlow Switch. The VNFs (TCP and Video Optimizer, firewall, HTTP content filter, etc.) run in a Cloud data center environment with RHEL OpenStack (Red Hat). The OpenStack coordinates the OpenDaylight Helium Controller to create OpenFlow rules in the switch to redirect and load balancing traffic to the VNFs. To keep track of VNF instances (in constant quantity variation for elasticity purposes) the OpenStack uses the LISP mapping service of OpenDaylight.

5.5.2 Network Slicing. Network Slicing refers to the partitioning of a certain physical infrastructure, composed of both network and computational resources, into multiple logical networks, called network slices [65]. Figure 8 shows that each slice is a self-contained network with its own virtual resources created on top of the underlying infrastructure. It can be designed and optimized for a particular mobile operator or service provider.

When compared to traditional physical networks, Network Slicing have the following advantages [65]: (i) customization of logical networks according to service requirements; ii) on-demand provisioning to scale resources up or down as conditions change, and (iii) network resource isolation for improved security and reliability.

Network Slicing aims at providing efficient resource sharing, traffic differentiation per slice, and management and protection tools [47]. NFV and SDN technologies are capable of providing the flexibility required in this context [50].

A use case for the use of NFV/SDN architectures in Network Slicing is for the creation of SDN-enabled Virtual Tenant Networks (VTNs). VTNs are virtual networks deployed to different tenants in an isolated way (independent of underlying physical network resources) to support specific Quality of Service (QoS) and Service Level Agreement (SLA) requirements. There is a trend to use SDN in the creation of virtual networks. By enabling network programmability, SDN renders the abstraction necessary for its use as a network hypervisor. In the case of SDN-Enabled VTNs, one

[12] https://www.openhub.net/p/nwepc.

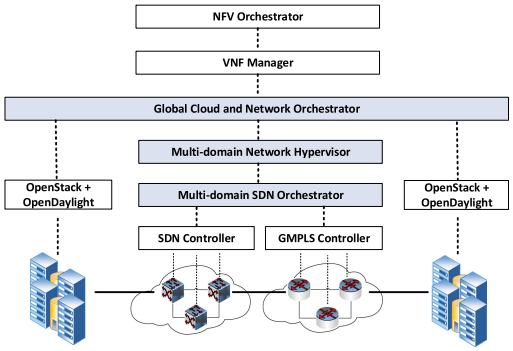

Fig. 9. An NFV/SDN architecture design for SDN-enabled VTNs [66].

or many SDN controllers create a VTN (called an Infrastructure SDN Controller), while a new SDN controller is instantiated to manage this VTN (called a Tenant SDN Controller).

When an SDN-Enabled VTN deployment takes place, the respective Tenant SDN Controller is manually installed and configured on a dedicated server, which can be a long process. NFV/SDN architectures can be used to virtualize tenant SDN Controllers and provide fast and dynamic VTN provisioning.

The works of Munoz and Vilalta [66–71] proposed an NFV/SDN solution for fast and dynamic deployment of SDN-Enabled Virtual Tenant Networks over multiple Data Centers and WAN domains. Their solution aims at providing geographically distributed cloud services with specific QoS and SLAs.

In Reference [66], the authors used NFV and Cloud to virtualize tenant SDN Controllers (OpenDaylight or Floodlight) to control the underlying SDN-enabled VTNs and provide fast and dynamic VTN provisioning (see Figure 9). They used OpenStack as VIM for each Data Center and an OpenDaylight controller to interconnect a virtual tenant SDN Controller with its respective VTN.

To create the VTNs, they used the Multidomain SDN Orchestrator (MSO) mechanism as a Network Operating System (NOS). The MSO creates an abstraction over multiple domains including different transport network technologies thus enabling the composition of end-to-end services over heterogeneous WAN networks. Also, the authors use the Multidomain Network Hypervisor (MNH) to create end-to-end SDN-enabled VTNs, over the abstraction provided by MSO. Using the Global Cloud and Network Orchestrator, a VIM mechanism, this architecture integrates geographically distributed Data Centers and multiple WAN domains, providing a unified cloud and network operating system for the creation of end-to-end NFV services over VTNs.

Several research articles focus on providing Network Slicing [56, 58, 65, 72, 73] for the new generation of mobile network. The 3GPP has identified Network Slicing as one of the key technologies to achieve the goals in 5G Networks [65], since it is a potential solution to enable suitable flexibility to address the specific requirements of different use cases. In this scenario, mobile operators could share the same physical network substrate, adding their virtual networks with their services (e.g., 3G, 4G services) and a centralized management plane, creating the so-called Mobile Virtual Network Operators (MVNO). These logical networks must be isolated from each other as a way to maintain privacy between operators. In this case, NFV provides the mobile network services per operator as VNFs and, in turn, SDN creates the slice as well as establishes network functions interconnectivity.

Mwangama et al. (2015) [73] designed an NFV/SDN architecture to support MVNOs in a federated cloud environment. A prototype was implemented using the non-open source FOKUS

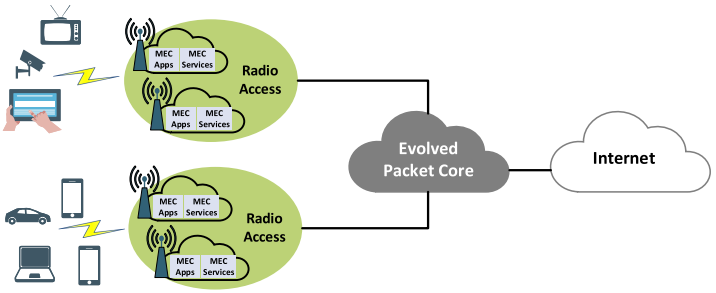

Fig. 10. Mobile edge computing scenario.

OpenSDNCore Orchestrator [17] to coordinate the network services between MVNOs. The Orchestrator uses OpenStack as VIM to create the virtual tenant networks and to instantiate the following VNFs per mobile operator: EPC (non-open source FOKUS OpenEPC[13] platform), IMS–IP Multimedia Subsystem (open source FOKUS OpenIMSCore[14]), M2M (non-open source FOKUS OpenMTC[15]).

Li et al. (2017) [65] proposed a three-layer Network Slicing framework model for 5G networks considering NFV and SDN technologies. The bottom layer is the 5G Software-Defined Infrastructure (5G-SDI), which comprises multiple administrative and physical domains (e.g., RAN, transport and core networks, etc.) with SDN-based control and management. Their SDN-based approach uses hierarchically organized SDN controllers to provide abstraction and distributed dynamic allocation of resources. Furthermore, RAN and MEC can be deployed to enable a cloud-based infrastructure. The Virtual Resource layer creates network slices with virtual resources (radio, computing, and network) and VNFs that are customized to meet the requirements of different types of services. The Application and Service layer includes the per-tenant services (e.g., connected vehicles, virtual reality, etc.) that will use these slices to perform their functionalities. Also, the life cycle of network slices is managed and orchestrated by the Slicing MANO that acts as VIM, VNF Manager, and Slice Orchestrator.

Recently Munoz and Vilalta (2016 and 2017) [74–79] have adapted the previously defined NFV/SDN architecture [66] to 5G scenarios, including the entire mobile network (e.g., radio access network) for fast and dynamic deployment of MVNOs. Besides, in References [78, 79] the authors proposed the ADRENALINE Testbed for 5G and IoT services on top of an NFV/SDN platform.

5.5.3 Mobile Edge Computing (MEC). Mobile Edge Computing (MEC) or Multi-access Edge Computing has been a trend in mobile networks. Like NFV, the MEC architecture has been standardized by ETSI through Group Specification (GS) MEC [80, 81] since 2016. MEC provides IT and Cloud Computing capabilities within the Radio Access Network (RAN). For this, a set of computer and storage resources (e.g., data centers, clusters, etc.) are deployed at the edges of a mobile operator's network to assist the core data center in supporting computing and communication (see Figure 10) [82]. MEC focuses on delivering the services closest to the user, as a way to meet certain critical application (e.g., video analytics, Internet-of-Things, augmented reality, and data caching) requirements that are not supported only by Cloud Computing, such as high bandwidth, low latency and jitter, context awareness, and mobility support.

[13] http://www.openepc.com/.
[14] http://www.openimscore.org/.
[15] http://www.open-mtc.org/.

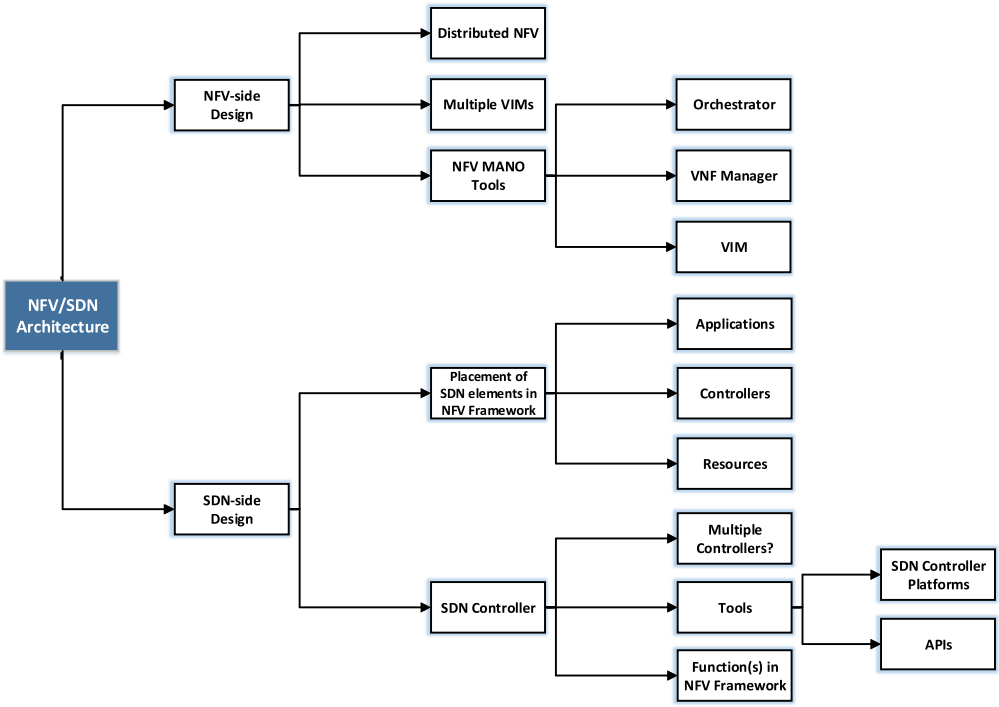

Fig. 11. A taxonomy to classify NFV/SDN architectures design.

According to 5G-PPP, MEC is vital technological component to enable 5G networks [83]. NFV/SDN architectures are in line with current trends for MEC solutions. Because it is a new technology, only a few studies have been found in this SLR. As an example, the EU H2020 SELFNET project [84] proposes the design and implementation of an Autonomic Management Framework for 5G networks, using technologies such as SDN, NFV, Self-Organizing Network (SON), Cloud Computing, and Artificial Intelligence. This framework aims at reducing OPEX and at improving QoE of the end users, addressing the following self-organizing capabilities: (i) self-protection against distributed cyber-attacks, (ii) self-healing against network failures, and (iii) self-optimization of the network traffic. In this context, Neves et al. proposed a SELFNET approach to support SFC in MEC scenarios [85, 86], to meet 5G requirements defined by the 5G-PPP initiative [46]. They considered a federated cloud infrastructure (i.e., multiples edge NFVI-PoP and a core NFVI-PoP) to provide IT and network resources to execute VNFs that support some management elements and network services. The WAN Infrastructure Management (WIN) uses SDN Controllers to provide connectivity between edge NFVI-PoPs and the core NFVI-PoP through the creation of virtual tenant networks.

6 TAXONOMY OF NFV/SDN ARCHITECTURES DESIGN

This section provides support to answer the third research question, by describing a taxonomy to organize the various decision-making levels for the design of NFV/SDN architectures.

Figure 11 depicts our proposed taxonomy that provides a conventional architectural design for using SDN in an NFV framework. It was derived from architectures and implementations found in the selected studies and published NFV/SDN reference architectures [15, 87]. This taxonomy is

useful as a guide for simplifying the work of researchers when studying NFV/SDN architectures or providing new solutions.

In this taxonomy, the NFV/SDN architectures design was divided into two sides: NFV-side and SDN-side. In the NFV-side, we must decide whether or not to use two features inherent to the architecture design. We describe these features as follows.

Distributed NFV (D-NFV): In D-NFV, the MANO framework places Virtual Network Functions (VNFs) where they could be most efficiently and economically be deployed, such as in data centers, forwarding devices, or the CPEs [36].

Multiple VIMs: A designer could place Multiple VIMs in different NFVI-PoPs to support the multi-domain administration or in the same NFVI-PoP to provide scalability and performance [15].

As far as we are concerned with NFV, it is important to identify the NFV Management and Orchestration (MANO) tools. Tools such as OpenMANO [88] and OpenBaton [89] can provide complete solutions for MANO. However, the OpenStack enables VIM implementation to provide support for existing or new VNF Managers and NFV Orchestrators.

On the SDN-side, a first step is the placement of SDN elements in the NFV Framework [15]. These elements are described below:

SDN Resources: Comprise of both physical and virtual switches and routers;

SDN Controllers: Responsible for controlling the SDN resources, determining the behavior of network traffic;

SDN Applications: Interfaces with one or multiple SDN controllers to enforce high-level network policy, such as firewall, network address translation, QoS, and network management.

We list some possible locations for the placement of SDN Resources in the NFV Framework, as follows [15]:

- Physical switch or router;
- Virtual switch or router;
- E-switch, software-based SDN-enabled switch in a server NIC;
- Switch or router as a VNF.

There are also some possible locations for the placement of SDN Controllers in NFV Framework [15]. They are the following:

- Merged with the Virtualized Infrastructure Manager (VIM);
- Virtualized as a VNF;
- As part of the NFVI and not as a VNF;
- As part of the OSS/BSS;
- As a Physical Network Function (PNF).

Finally, some locations for the placement of SDN Applications in NFV Framework are listed below [15]:

- As part of a PNF;
- As part of the VIM;
- Virtualized as a VNF;
- As part of an EMS;
- As part of the OSS/BSS.

There are also some architectural decisions to be made when one considers the SDN Controller, as follows:

- **Does the solution implement multiple SDN Controllers? If so, then what is the main objective?** Multiple SDN Controllers are hierarchically distributed to provide performance, scalability, reliability, administrative domains interaction, or Network as a Service (NaaS) management in an NFV Framework.
- **What is the function of SDN Controller in the NFV Framework?** Network Connectivity in the NFVI, Control of Virtual Networks, Interconnecting VNFCs, and Interconnecting VNFs are some of these functions (extracted from the studies selected in this SLR).

Finally, we must identify the SDN Controller tools, including its underlying software and the used bound interfaces (at South, North, West, and East). As SDN Controllers we can cite OpenDaylight [32], Floodlight [90], ONOS [91], Ryu [92], and POX [33].

The next Sections (7 and 8) aim to answer research question 3, using this taxonomy to organize the description and the differences between the NFV/SDN solutions extracted from articles selected in SLR.

7 NFV-SIDE DESIGN

This section uses the NFV-side of the taxonomy described in Section 6 to organize the NFV/SDN solutions extracted from articles selected in the SLR.

We organized the studies according to how they implemented the components of MANO (i.e., the NFV Orchestrator, the VNF Manager, and the Virtualized Infrastructure Manager). For each component, the research studies are classified as follows:

- Real Implementation: The study proposes and implements its own component;
- Theoretical: The study proposes its own component, but it is not implemented;
- Vendor-specific: The study uses a proprietary tool to implement the component.

The majority of the articles were classified as *Real Implementation* (see Figure 12). These studies adopt modern tools to assist in the implementation of solutions, mainly the VIM component (e.g., OpenStack [31]). However, it is worth mentioning that most of them implement the orchestration functions without the support of existing NFV MANO frameworks, such as OpenStack Tacker, OpenMANO, OpenBaton, Open-O, ECOMP, Hurtle, and the like. However, three articles [16, 58, 73] developed their solutions on top of the OpenSDNCore Orchestrator [17], from Fraunhofer FOKUS Institute. Last, we highlight that the Vendor-specific solutions are only used in some PoCs from ETSI NFV ISG.

Furthermore, some studies have provided architectures that deal with Distributed NFV (D-NFV) and Multiple VIMs. With D-NFV we could place VNFs wherever they may be most effective (performance and scalability) and least expensive. However, Multiple VIMs are often used to perform management of several administrative domains or NFVI-PoPs. In this scenario, VIMs are hierarchically distributed. So-called secondary VIMs are responsible for managing NFVI-PoPs. The primary VIM controls the secondary VIMs to create an abstraction layer on all NFVI-PoPs and performs a centralized management. Such a hierarchy enables the creation of end-to-end network services, involving multiple domains (e.g., Cloud and WAN).

Table 2 lists the studies that implement Distributed NFV or Multiple VIMs. It is worth mentioning that most of the studies implemented both designs (see Figure 13). As an example, Reference [93] proposed the vConductor, a Cloud CPE (see Section 5.3) solution for automation of multitenant virtual network provisioning. vConductor deploys all enterprise network functions as VNFs in a Cloud domain composed of multiple data centers. By using a User Portal, the customers can

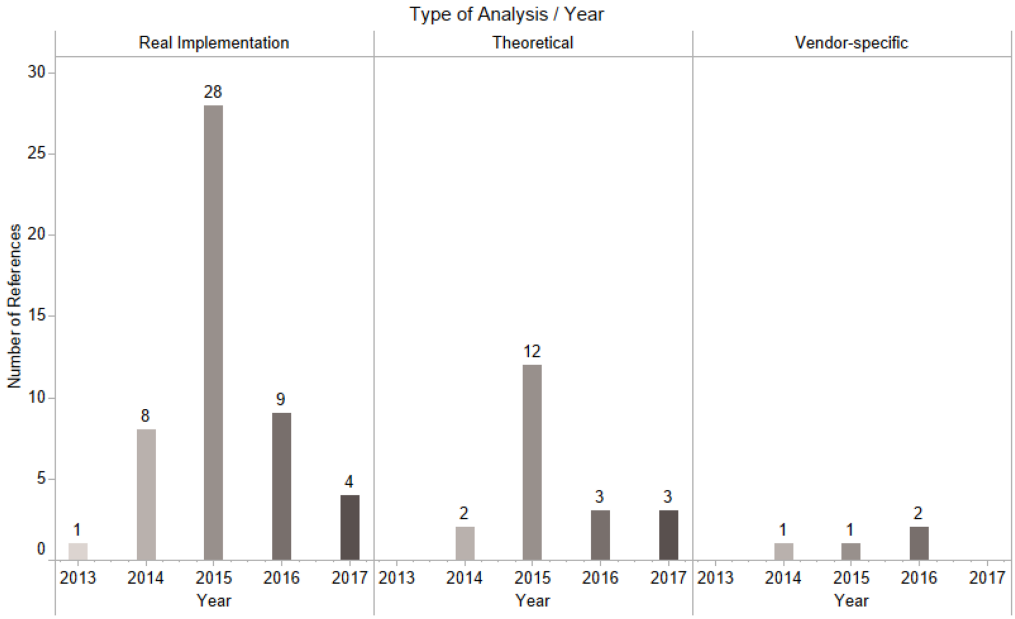

Fig. 12. Number of studies per type of analysis/year.

Table 2. List of Studies Addressing Distributed NFV and/or Multiple VIMs

Implementation	Studies
Distributed NFV	[16, 24, 27, 29, 35, 37, 38, 51, 55, 63, 65–71, 74, 75, 76, 77, 78, 79, 85, 86, 93, 94, 95, 96, 97, 98]
Multiple VIMs	[3, 16, 21, 24, 27, 29, 35, 37, 38, 51, 55, 63, 65–71, 74–79, 85, 86, 93–95, 98, 99]

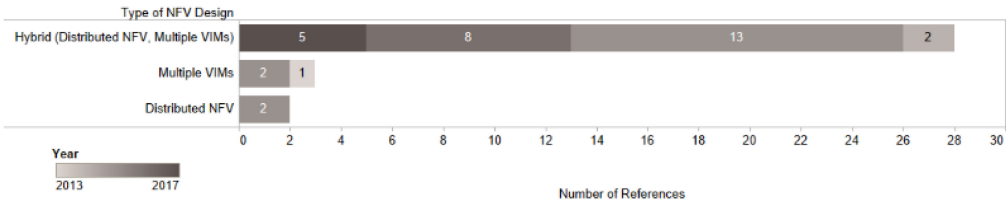

Fig. 13. Number of studies addressing distributed NFV and/or multiple VIMs.

acquire new network functions and define how their VNFs must be chained. A virtual tenant network (VTN) is established connecting the enterprise CPE and the Cloud infrastructure through an OpenFlow-enabled WAN domain. Each data center in a Cloud domain uses an OpenStack as a management platform (secondary VIM). Further, an OpenDaylight Controller (secondary VIM) updates the OpenFlow rules required for VTN management in WAN domain. vConductor acts as NFVO, VNFM, and primary VIM, controlling the multiple secondary VIMs. Finally, vConductor includes the Virtual Network Life Cycle Manager (VNLM) to creates a D-NFV scenario. VNLM implements a multi-objective resource scheduling algorithm (MORSA) that uses a genetic algorithm to provides near-optimal placement of VNFs over different data centers.

Table 3. The Position of Virtual Switches in NFV Framework

Position	Studies
NFVI	[3, 16, 19–29, 35, 37–39, 41, 53, 56–59, 62–79, 85, 86, 94–97, 99, 101–112]
VNF	[54, 62, 85, 86, 113, 114]

However, there are also works that implement only one of these scenarios. As an example of D-NFV scenarios without multiple VIMs, the authors of Reference [96] proposed an NFV/SDN architecture, called NetFATE (Network Functions At an Edge), aimed at allocating VNFs at the CPE nodes (multiple NFVI-PoPs) to minimize end-to-end latency. For this, a general-purpose computer replaces the old proprietary hardware-based CPE. The new CPE operating system is the CentOS 6.4 running a Xen Hypervisor [100] for network function instantiation as VNFs, and an Open vSwitch (OVS) for VNF interconnection. NetFATE works as the MANO framework, using the NFV Coordinator (C++ software) to manage the VNF life cycle and a POX controller to control the OVSs. Finally, the Orchestration Engine determines how to distribute the VNFs and compose the network services.

When we consider the use of multiple VIMs without distributed NFV, we usually have a scenario where there are two secondary VIMs, one to manage a data center for virtualization purposes (Cloud domain) and another to manage a transport network (WAN domain) for end-to-end network services provisioning. As an example, PoC 16 [99] proposed a multi-domain NFV/SDN architecture intended to provide enterprise services (firewall, IPS/IDS, and load balancer) to remote users across an MPLS-based transport network. This PoC uses an OpenStack (secondary VIM) as cloud orchestrator for VNF instantiation while ensuring end-to-end connectivity and SLAs over the WAN by using OpenFlow with Ryu controller (secondary VIM). The NFV Orchestrator acts as both NFVO and primary VIM, controlling the secondary VIMs to instantiate the end-to-end network services.

8 SDN-SIDE DESIGN

This section uses the SDN-side of the taxonomy described in Section 6 to organize the NFV/SDN solutions regarding the SDN.

8.1 Placement of SDN Elements in the NFV Framework

Table 3 shows how the studies place virtual switches in the given NFV Framework. The NFVI is the most used as a location for SDN resources. This scenario is a common approach to providing network programmability and flexibility for connectivity and traffic steering among VNFs.

However, some works also include virtual switches as VNFs. These works intend to provide an SDN-enabled virtual network to different customers. In [54, 113], this placement is possible because they work with the Forwarding and Control Element Separation (ForCES) protocol [119] as SDN Southbound API and consider the Logical Functional Blocks (LFBs) as VNFs. However, Neves et al. [85, 86] created an abstraction for deployment of network services through the instantiation of Virtual Network Elements (VNE). VNEs are VNFs running a virtual switch process that perform packet processing (networking services) over the network traffic. VNEs can be distributed in the core or the edge NFVI-PoPs (see Section 5.5.3).

Table 4 shows how the studies positioned the SDN Controllers in the given NFV Framework. According to Reference [15], an SDN Controller can run in five places: NFVI, VIM, VNF, OSS/BSS, and can be a Physical Network Function (PNF). In this work, we did not find references for the last 2 (two) placements. Table 5 shows how the studies positioned the SDN Applications in the

Table 4. The Position of SDN Controllers in the NFV Framework

Position	Studies
NFVI	[3, 16, 23–25, 27, 28, 35, 37, 38, 58, 59, 63–71, 74–79, 94, 98, 101, 109–111, 115]
VIM	[3, 16, 19–22, 24, 26, 27, 29, 39, 41, 51, 53–56, 61, 72, 73, 85, 86, 93, 95–97, 102–108, 112–114, 116–118]
VNF	[27, 52, 53, 56, 57, 60, 62, 66–71, 74–79, 85, 86, 99]

Table 5. The Position of SDN Applications in the NFV Framework

Position	Studies
VIM	[3, 16, 19–, 22–29, 35, 37–39, 41, 53, 56, 58, 63, 65–71, 74–79, 85, 86, 93–99, 101, 103, 106–112, 114]
VNF	[20, 21, 27, 51–54, 56, 57, 59–62, 64, 72, 85, 86, 102, 104, 105, 115, 116]
OSS/BSS	[3, 22, 55, 66–71, 73–79, 94, 97, 106, 113, 117, 118]

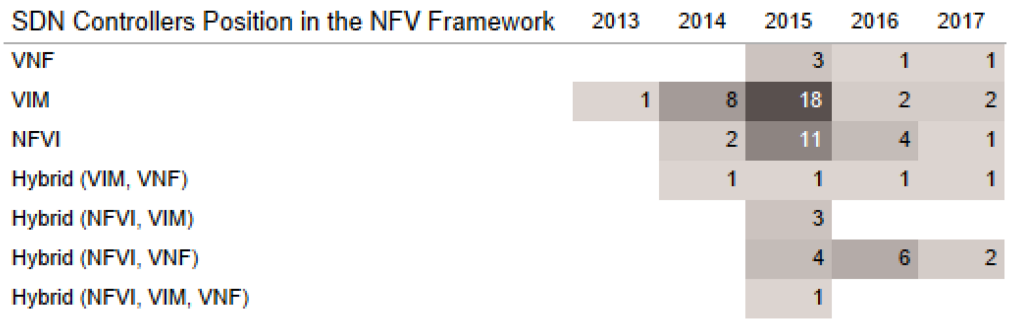

Fig. 14. Number of studies addressing the position of SDN controllers in the NFV framework.

NFV Framework. According to Reference [15], SDN Applications can run in five points: as part of a PNF, as part of the VIM, virtualized as a VNF, as part of an Element Manager (EM), and as part of the OSS/BSS. In this work, we did not find references to SDN Applications as part of a PNF or an EM.

The VIM is the most used as a position for both SDN Controllers and Applications (see Figure 14). The VIM is the best place for these elements because it offers a global view of both NFVI physical and virtual infrastructures and the VNFs. This property allows the implementation of different functionalities, such as VNFs or VNFCs interconnections, network connectivity in the NFVI, and the control of virtual networks as shown in Table 6.

However, the NFVI have also been widely adopted as SDN Controller placement. According to [15], this scenario is a classic case of the SDN controller providing network connectivity in the NFVI. In this work, we consider the SDN controller as a NFVI component when the VIM and its functions are distinguishable, as in the case of OpenStack controlling OpenDaylight.

A good example to illustrate SDN Controllers and Applications placement is the work of Rossem et al. (2015) [27]. In that work, the authors have used ESCAPE (see Section 5.2) environment to implement a NFV/SDN solution for elastic virtual router provisioning, needed in a VPN service. The main goal is to increase the throughput by load balancing (using Valliant Load Balancing) traffic among multiple virtual switches. In this architecture, the Service Layer receives the VPN requests and define the required VNFs to be instantiated by Orchestration Layer (OL) in an optimal

Table 6. List of Possible Functions for SDN Controllers When Applied in the NFV Framework

Function	Studies
Interconnecting VNFs/VNFCs	[3, 16, 19, 20, 22–24, 26–29, 35, 37–39, 51–56, 58–65, 73, 85, 86, 95–99, 101, 103–115, 117]
Network Connectivity in the NFVI	[3, 16, 21, 22, 24, 25, 27–29, 35, 37–39, 41, 51–53, 55, 57–60, 63–72, 74–79, 85, 86, 93, 94, 96, 98, 99, 101, 102, 107, 109, 111, 114–116, 118]
Control of Virtual Networks	[22, 39, 41, 51, 56, 66–79, 85, 86, 107, 111, 118]

way. There are three VNF types: *Ctrl App*, *OF Ctrl*, and SDN-enabled virtual switches. The *Ctrl App* and *OF Ctrl* are deployed as VMs in OpenStack (Cloud domain), while SDN switches are deployed in an Mininet[16] emulator (representing an OpenFlow domain). On the data plane, an elastic router comprises one or more SDN switches. On the control plane, the *SDN Ctrl* manages the topology creation on the top of SDN switches. Moreover, the SDN application *Ctrl App* monitors the SDN flow statistics and triggers topology changes (if needed), adding more or less SDN switches.

In Reference [27], a hybrid solution for the positioning of SDN controllers was proposed. In this case, the POX Controller [33] was placed on VIM to support the creation and management of network services in OpenFlow domains. The OpenDaylight [32] was placed on NFVI and is used by the OpenStack [31] to provide connectivity in the Cloud domain. Finally, the *SDN Ctrl* is a Ryu Controller created as a VNF to coordinate the SDN-enabled virtual network.

Regarding SDN Applications placement, the *SDN App* also runs as a VNF on top of *SDN Ctrl* (Ryu Controller). For OpenFlow domains, the SDN applications run as a VIM component, on top of POX. Finally, for Cloud domains, the Neutron service (VIM) performs the control of OpenDaylight instance.

The Operations support systems (OSS) and Business Support Systems (BSS) have also been widely adopted as SDN Applications placement. Application at this level enables multiple tenants to control dedicated SDN networks to provide their own services. This scenario is common in works that propose solutions for 5G Cellular Networks [73, 85, 86]. Examples of SDN application placement as OSS/BSS management task are the works of Munoz and Vilalta [66–71, 74–79]. In those works, a tenant SDN Controller runs as a VNF to control an underlying VTN. The end-users or service provider operators (components of OSS/BSS element) have direct access to this controller and can implement customized applications for VTN control and management.

8.2 SDN Controller Functions in the NFV Framework

Table 6 shows the possible functions for SDN Controllers when applied in the NFV Framework. Below, we describe these functions:

Interconnecting VNFs/VNFCs: The SDN Controller might be used to connect and manage the traffic between VNFs and/or VNFCs to enable Network Services, by creating Service Function Chaining (SFC). As seen in Section 2.1, a VNF might be composed of several VNFCs.

Network Connectivity in the NFVI: The SDN Controller is used to provide L2/L3 connectivity among end-to-end devices;

Control of Virtual Networks: The SDN Controller might be responsible for creating and managing virtual networks for different customers.

[16] http://mininet.org/.

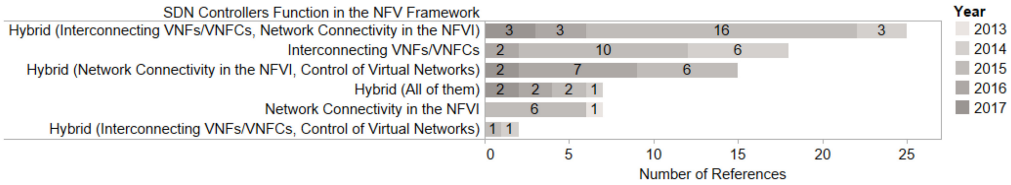

Fig. 15. Number of studies addressing SDN controller functions.

Table 7. List of Objectives for Use Multiple SDN Controllers

Objective	Studies
Distributed Performance	[63]
Scalability	[28]
Reliability	[66–71, 74–79]
Administrative Domains Interaction	[3, 16, 21, 24, 27, 51, 65–71, 74–79, 85, 86, 94, 98, 99, 107]
NaaS Management	[3, 19, 26, 27, 29, 35, 65–72, 74–79, 85, 86, 99, 107]

It is worth emphasizing that interconnecting VNFs is the main objective for using SDN in the NFV Framework (see Figure 15), mainly for automation and monitoring of SFC deployments. The works of Cziva el al. [19, 26, 29] used an OpenDaylight Controller to create OpenFlow rules in Open vSwitch instances and OpenFlow switches, intending to provide interconnectivity and traffic steering for network services (set of linked VNFs). According to Reference [6], there are some problems regarding Interconnecting VNFs, such as network function placement, survivability of VNFs, and dynamic service function chaining (elasticity).

As an example of SDN Controllers performing Network Connectivity in the NFVI and Control of Virtual Networks, Vestin et al. (2015) [41] used OpenDaylight Controller as VIM to manage the communication among physical APs and Virtual Access Points (VAP), as shown in subsection 5.4. In this case, the controller provides Network connectivity in the NFVI by connecting physical AP and Cloud infrastructure, and Control of Virtual Networks by using OpenFlow to connect a mobile client with its respective VAP.

8.3 The Use of Multiple SDN Controllers

Table 7 shows the main objectives for implementing Multiple SDN Controllers in the NFV Framework. As well as the use of Multiple VIMs (see Section 7), multiple SDN controllers are organized hierarchically to provide the following objectives.

Distributed Performance: When the VNFs have to be distributed to the chosen location, they are interconnected by location-specific SDN Controllers;
Scalability: Multiple controllers manage an NFVI-PoP infrastructure;
Reliability: Fault tolerance, disaster recovery, and full isolation management.
Administrative Domains Interaction: Communication management of different scenarios (e.g., Cloud and WAN) using different SDN Controllers integrated hierarchically in a unique platform;
Network as a Service (NaaS) Management: The concept of NaaS is related to the provision of virtualized network services to customers with different requirements [120]. This function includes network virtualization (Network Slicing).

In PoC 34 [63], the developers implement Distributed Performance by putting EPC Virtual Functions to different NFVI-PoPs, each NFVI-PoP managed by its own SDN Controller (OpenDaylight). These controllers communicate with each other to allow the programming of GPRS Tunneling Protocol (GTP) tunnels interconnecting VNFs placed in different locations.

Scalability is handled in Callegati et al. (2015) [28]. In this work, the authors used two SDN controllers to manage the virtual networks in an OpenStack-based Cloud environment. They create OpenFlow rules using a Neutron Open vSwitch Agent for connectivity in NFVI and interconnection of VNFs, and monitor the throughput of OpenFlow rules using a POX [33] controller for traffic steering mechanisms.

Regarding Reliability, the works of Munoz et al. (2015) [66, 67] used NFV and Cloud to virtualize tenant SDN Controllers to control the underlying VTNs. The authors pointed out that a clear advantage of using cloud virtualization for SDN controllers is the reliability achieved with the lack of hardware maintenance downtime and the decreasing recovery time.

However, it is worth noting that Administrative Domains Interaction and NaaS Management are the main objectives for using multiple SDN controllers in the NFV Framework. As an example, Rossem et al. (2015) [27] used three SDN controllers in their NFV/SDN architecture to provide elastic virtual router provisioning in a multi-domain scenario (described in Section 8.1). For Administrative Domains Interaction, the authors used OpenDaylight and POX controllers to provide an SDN-enabled virtual network on top of a Cloud (OpenStack-based) and a WAN domain (OpenFlow-based), respectively. Finally, for NaaS Management, the authors instantiate SDN Controllers (Ryu) as VNFs to control the underlying virtual networks.

9 LESSONS LEARNED

This section summarizes our view from the literature review described in the previous sections (Sections 5, 7, and 8) and points out some trends for the design and implementation of NFV/SDN architectures.

Cloud Computing is the dominant scenario for implementing NFV/SDN solutions (72% of the studies found). According to Reference [6], Cloud Computing and Software-Defined Networking (SDN) are two concepts closely related to NFV. Most of the proposed NFV solutions have been implemented and tested in cloud-based environments. It has been the primary choice for the creation of NFV infrastructures (NFVI) mainly due to its flexibility, rapid deployment of new services, and inherent elasticity. The VNFs of a specific SFC are deployed as functions in dedicated Virtual Machines (VMs), which can be instantiated on devices placed in different geographic locations. Cloud Computing allows NFV/SDN solutions to provide better services for users by simplifying the provision of network services and enabling the quick deployment, management, and optimization of physical infrastructure dynamically, using resource virtualization mechanisms.

At the SDN-side, the SDN elements have been placed in different points of the NFV framework. It is clear that SDN Switches are most present in NFVI, whereas SDN Controllers are deployed in VIM and at the NFVI. Also, SDN Applications are usually placed in VIMs. The VIM is most used as a position for both SDN Controllers and Applications. The VIM seems to be the best place for these elements because it offers a global view of both NFVI physical and virtual infrastructures and the VNFs. It allows the implementation of different functionalities, such as VNFs or VNFCs interconnections, network connectivity in the NFVI, and the control of virtual networks.

Another important observation is that the most used elements in the VIM component are the OpenStack [31] and the SDN Controller OpenDaylight (ODL) [32] due to their soft integration. The Neutron service uses the Module Layer 2 (ML2) Plugin [31] to provide networking services in a Cloud. The ML2 might control an ODL instance using the Neutron API, a REST API provided by ODL. However, the ONOS SDN Controller has gained space in both academia and industry and is

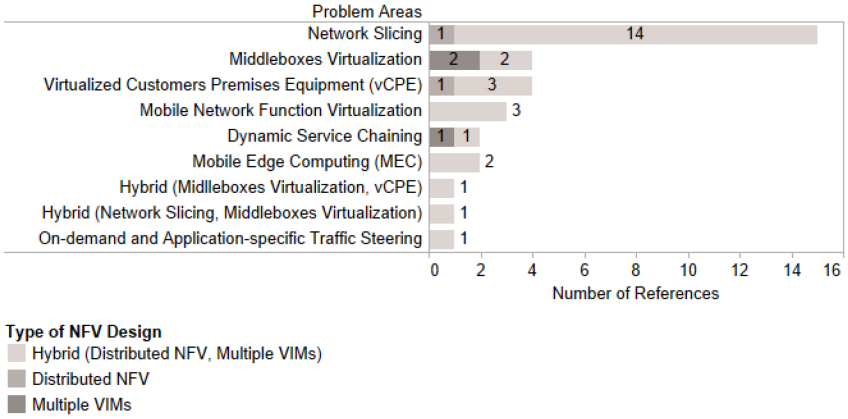

Fig. 16. Statistics related to the problem areas versus NFV design.

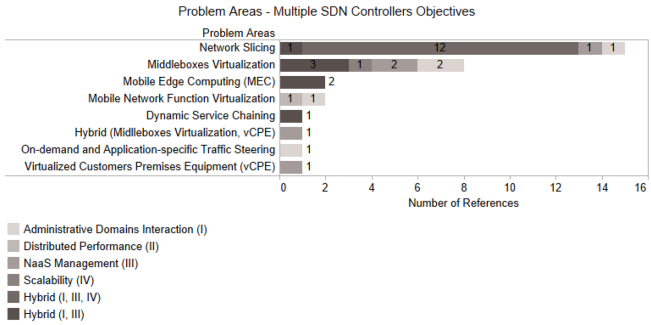

Fig. 17. Statistics related to the problem areas versus objectives of multiple SDN controller.

currently the leading competitor of ODL. ONOS is the official distribution of the Open Network Foundation (ONF). Some industrial use case projects have been used ONOS, such as the Central Office Re-architected as a Datacenter (CORD), an NFV/SDN platform supported by major service providers (e.g., Google and Verizon).

The number of studies using OpenFlow 1.0 in NFV/SDN architectures has been steadily declining over the years (there is only one study in 2016) [118]. As expected, there is a clear trend to use the most current OpenFlow version in recent studies [29, 39, 52, 63, 99, 110]. ForCES does not seem to have attracted the interest of the research community, since only a few studies have used it as the southbound protocol [54, 113].

In Section 7, the reader can observe that most solutions rely on both Distributed NFV and Multiple VIMs (see Figure 16), except for Wireless LAN and Wireless Mesh Networks. Particularly, 80% of the studies for Network Slicing have used this type of NFV design, mainly because they are deployed under multiple administrative domains (see Figure 17) including different types of network infrastructures (e.g., RAN, transport and core networks).

As far as we are concerned to the SDN Controller Functions (see Figure 18), the primary focus for all areas was *Interconnecting VNFs/VNFCs*, which can be considered indispensable for any NFV/SDN architectures. As SDN can deliver intelligent traffic steering, service chains can indeed benefit from this integration. Also, although the *Control of Virtual Networks* function has been widely adopted in areas (e.g., Network Slicing with 75% of studies), we have not identified its implementation in other scenarios such as On-demand and Application-specific Traffic Steering,

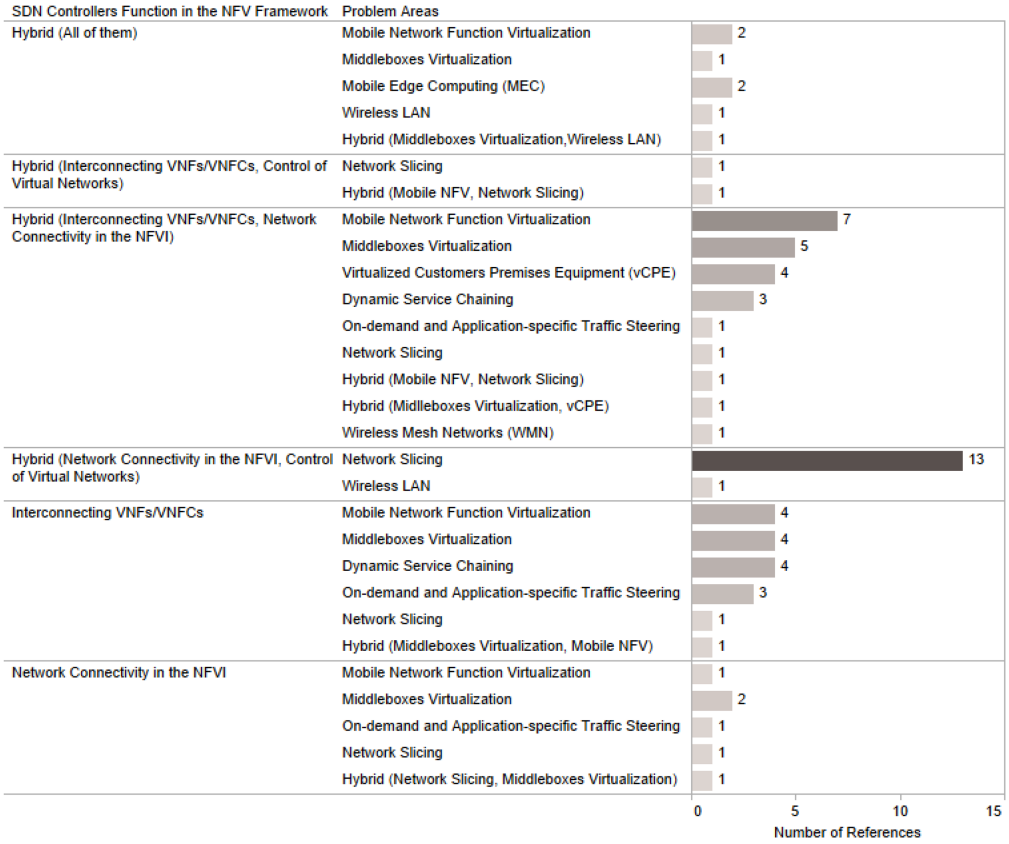

Fig. 18. Statistics related to the problem areas versus SDN controller function.

Dynamic Service Chaining, and vCPE. We argue that the absence of this functionality is not an impediment to the implementation of solutions for traffic steering and dynamic SFC and it should be a functional requirement for vCPE. Please recall that vCPE must provide multi-tenant services, leaving the client responsible for the selection and configuration of their VNFs.

As shown in Figure 17, the use of multiple SDN controllers has not been explored in Wireless LAN and Wireless Mesh Networks solutions, which may be a gap to be addressed for new NFV/SDN architectures with multiple controllers. Furthermore, we have identified a few studies dealing with *Scalability* (elasticity mechanisms) and *Reliability* (fault tolerance mechanisms) problems (see Section 7 and subsection 8.3), which require orchestrators to manage environments with multiple VIMs and SDN controllers to provide support to perform D-NFV management in several administrative domains or NFVI-PoPs. According to the 5GPPP, such characteristics are essential for the 5G network, since novel 5G technologies (e.g., Network Slicing) might impact the entire mobile network including mobile devices; radio access, transport, and core networks; and the cloud (local, regional, or global).

Finally, the use of NFV/SDN architectures has been a growing trend for providing fast delivery of network services in a flexible and automated way for 5G networks. A certain NFV/SDN architecture creates an abstraction layer that unifies both computer and network resources and enables dynamic and application-specific traffic steering. Most studies have demonstrated a growing attention to the cost problem where they tried to reduce both CAPEX and OPEX costs. However, the

use of NFV/SDN architectures for the Mobile Edge Computing (MEC) is incipient. Current solutions are still in their infancy, and better schemes are needed to provide distributed and dynamic SFC as well as to meet flow requirements. The problem of how to integrate MEC and network slicing in a unique NFV/SDN architecture has been slightly addressed [56, 58, 72, 73, 85].

10 FUTURE RESEARCH DIRECTIONS: OPPORTUNITIES AND CHALLENGES

We have analyzed the selected studies and the surveys to have a clear view of the directions for future research efforts. We have identified some challenges in the design and implementation of NFV/SDN architectures. They are described in the following subsections.

10.1 Deployment of Network Services

Service Function Chaining (SFC) provides an ordered list for service processing of traffic flows [9]. The fast SFC deployment and provisioning of NFV and the centralized and flexible control of SDN have enabled new opportunities regarding this topic. However, solutions that provide better performance and optimal resource utilization in function deployment are still needed. Below, we describe some challenges related to NFV/SDN solutions regarding the deployment of network functions.

VNF Performance: Virtualized network functions should meet the user's performance requirements, especially when the SDN Controller is a VNF. Current hypervisors must be optimized for fast packet processing in standard servers as a way to obtain high I/O speed, short transmission delays, and so on. Some initiatives are DPDK [121], NetVM [122], and ClickOS [123]. Further, the use of container-based virtualization, such as Docker containers, is of paramount importance in environments that require high-performance with low resource consumption such as edge devices (e.g., vCPEs, MEC, etc.), since they have relatively low capabilities compared to traditional NFV servers. However, there are some challenges to be considered when choosing the containers technologies for deploying VNFs in NFV/SDN architectures [29, 124]:

- Orchestration: all containers share the same kernel as well as its services and configurations. This characteristic increases the complexity of orchestration and management platforms, since the allocation process must take into account whether a VNF has unique needs in kernel, such as a particular module or configuration;
- Security: container-based virtualization has a broader attack surface than other virtualization techniques, since its interface is more sophisticated than hardware emulation interfaces. Besides, it provides weaker functional isolation among instances, since containers may require different kernel configurations that can conflict. It also offers more vulnerable performance isolation, since containers in the same host can be placed under resource pressure (e.g., memory or CPU overconsumption) by external attacks or new instances.

VNFs Scheduling and Placement: The scheduling and placement of VNFs impact the performance of Service Chaining significantly. For better performance, the physical resources should be used efficiently. Also, energy-efficient hardware and energy-aware network service placement remain some of the main challenges in NFV and solutions are still limited [6]. Therefore, optimization and machine learning techniques are necessary to achieve optimal, automatic, and dynamic resource reservation, allocation, and migration of VNFs, considering a global view of the resources and the customer requirements. Integer programming and heuristic approaches can be used for VNFs Scheduling [93] and Placement

[25], considering resource constraints. Tools such as Google's Borg [125], Omega [126], and Apache Mesos [127] may be considered for scheduling of VNFs.

High-level Policies: The definition of high-level policies is necessary to simplify the configuration of NFV Orchestrator operations, such as resource allocation and optimization mechanisms, and to meet the customers' requirements (interfaces to OSS/BSS). In this case, OpenStack's HOT (Heat Orchestration Template) and TOSCA (Topology and Orchestration Specification for Cloud Applications) template languages could be used [31].

Traffic Steering: In NFV/SDN solutions, traffic steering and network function deployment should be optimized jointly, providing a network-aware scheduling mechanism [35, 101] that deploys VNFs considering both the paths expressed in the forwarding graph and the network behavior (available bandwidth, latency, jitter, etc.). As a consequence, more variables are introduced, and heuristic algorithms should be created to reduce computing complexity.

Elastic Network Function: The dynamic service scaling at runtime provides better resource utilization, reducing both CAPEX and OPEX, and maintains service level requirements [3]. It is necessary that NFV/SDN solutions can scale (in/out or up/down) networking services and monitor both servers' and networks' resources to offer elastic, pay-as-used services.

Orchestration: Orchestration services are necessary for elastic, adaptable, and autonomic network function deployment, provisioning, and management. Tools such as OpenMANO [88] and OpenBaton [89] might be used as a solution for NFV MANO (Management and Orchestration).

10.2 Improving the Programmability

SDN and NFV are the critical enablers for realizing some of the expected features in 5G networks, such as network programmability, flexibility (e.g., network abstraction, infrastructure sharing, and reconfigurability), adaptability (e.g., self-healing, self-configuration, self-protection, and self-optimization) and capabilities (e.g., network slicing and MEC) [47]. However, some improvements should be provided so that the existing SDN standards such as OpenFlow can be applied in this type of scenario.

OpenFlow is the most used protocol for the Southbound API in NFV/SDN solutions, as described in Section 8. However, currently, it does not support application layer packet processing. The application layer inspection and classification is necessary to provide fine-grained flow distribution for different network services, and thus to provide intelligent service chaining.

Finally, OpenFlow is not suitable for Wireless Networks (e.g., WiFi, LTE, etc.), since flow tables just include rules for Ethernet-based switches. Wireless communication is more complex, as wireless links are time-varying and vulnerable to interference. For this, extensions must be implemented to allow WiFi programming rules, enabling the matching and monitoring of wireless frames [22]. Besides, SDN must provide support to Radio Access Network (RAN) virtualization infrastructures [128]. In this case, SDN approaches must support both legacies (e.g., 3G and 4G) and new radio access technologies (e.g., 5G and narrowband Internet of Things, NB-IoT), ensuring radio resource isolation.

10.3 Multi-Tenant, Multi-Service, and Multi-Domain Support

An NFV/SDN architecture that supports multiple domains or NFVI-PoPs is necessary for the provision of quality of service (QoS) and SLA enforcement in multi-tenant environments with end-to-end services. However, it remains a challenge, since the orchestration functions must support the following features [47]:

- Multi-domain orchestration of diverse programmable infrastructure technologies (e.g., RAN, transport and core networks, data centers, etc.), possibly belonging to different operators;
- Northbound interface for Network Slicing management, providing multi-tenancy and multi-service support;
- End-to-end network slices that are flexible to the dynamic requirements of different services (e.g., IoT, smart cities, etc.) and mobile operators, providing a multi-service and context-aware adaptation of network functions;
- Advanced autonomic network management platforms to address complexity in such scenarios.

Furthermore, studies are still being carried out to evaluate the impact of end-to-end slices on the RAN design. RAN Virtualization is currently under investigation and is one of the major obstacles to creating NFV/SDN architectures for 5G networks [128].

10.4 Multiple SDN Controllers

NFV/SDN solutions could be used for the control and management of heterogeneous network resources (Optical, MPLS, IP, etc.), distributed in different geographical locations. Therefore, hierarchical and federated SDN Controllers must be used to meet scalability, availability, reliability, and end-to-end (multi-domain) provisioning requirements. Tools such as FlowVisor as well as North/East/WestBound interfaces from popular SDN Controllers (OpenDayLight [32] and ONOS [91]) can be used to provide such solutions.

10.5 Security

In addition to current security problems that are unique to each technology [5, 6], the NFV/SDN solutions also have security challenges related to the integration process, such as the lack of authentication and authorization mechanisms in the communication interfaces between SDN and NFV modules. Besides, by exploring network programmability, security services should be developed to deal with malfunctioning software (e.g., detecting and preventing exploits) or attacks caused by malicious adversaries (e.g., intrusion detection and prevention systems) such as Distributed Denial of Service (DDoS) [82]. Such services should take into account attack surfaces at all levels of the infrastructure, including network, edge and core data centers, virtualization, and user devices.

Furthermore, regarding 5G networks, security in network slicing is a complex task, since there is resource sharing among slices and they may have different security policy requirements. This problem gets worse when we consider multi-domain scenarios. In this context, security solutions in the NFV/SDN architecture should provide mechanisms for resource isolation between slices, considering their impact on the entire infrastructure and providing security policy coordination among different domain infrastructures [129].

10.6 Extensibility and the Expressiveness of NFV/SDN Models

It is important to use a single model (framework) to address both NFV and SDN issues, instead of a combination that focuses on one problem at a time. This type of model eases implementation and the learning curve as well as reduces interdependencies (plugins to interconnect different frameworks) [54].

10.7 Standardization

Even with the existence of reference architectures defined by industry [87], an effort should be made towards standardization of an architecture that integrates the NFV and SDN technologies to

simplify the work of researchers when providing new NFV/SDN solutions. This reference architecture must include standardized interfaces and resource catalogs [93] so that new VNFs can be rapidly integrated and deployed into the system.

11 CONCLUSION

Even with different purposes, NFV and SDN are complementary paradigms and technologies capable of providing one consolidated solution that offers the best of both technologies. NFV/SDN architectures are of paramount importance for a passage from the static design of conventional networks to an intelligent, open network environment. Therefore, this work proposed a Systematic Literature Review (SLR) for NFV/SDN architectures, intending to provide a profound understanding of such integrated designs. We aimed to identify the current trend in this field. For this, a total of 74 articles have been studied in-depth according to our predefined SLR protocol. Through comprehensive analysis and interpretation of the collected data, this SLR achieved three goals. First, we described the main characteristics (target environment and problems to solve) of integrated NFV/SDN solutions practices. Second, we compared their architecture designs (NFV framework design and tools, SDN APIs and place of SDN elements) and classified them against the presented taxonomy. Then, we discussed some opportunities and challenges for research work in the next generation of NFV/SDN architectures.

ELECTRONIC APPENDIX

The electronic appendix for this article can be accessed in the ACM Digital Library.

REFERENCES

[1] ETSI. 2012. Network functions virtualisation—An introduction, benefits, enablers, challenges and call for action. *White Paper* (Out. 2012).

[2] ETSI. 2015. Network functions virtualisation (NFV)—Network operator perspectives on industry progress. *White Paper* (Jan. 2015).

[3] Róbert Szabó, Mario Kind, Fritz-joachim Westphal, Hagen Woesner, Dávid Jocha, and András Császar. 2015. Elastic network functions: Opportunities and challenges. *IEEE Netw.* 29, 3 (Jun. 2015), 15–21. DOI: http://dx.doi.org/10.1109/MNET.2015.7113220

[4] Nick McKeown, Tom Anderson, Hari Balakrishnan, Guru Parulkar, Larry Peterson, Jennifer Rexford, Scott Shenker, and Jonathan Turner. 2008. OpenFlow: Enabling innovation in campus networks. *SIGCOMM Comput. Commun. Rev.* 38, 2 (Mar. 2008), 69–74. DOI: http://dx.doi.org/10.1145/1355734.1355746

[5] D. Kreutz, F. M. V. Ramos, P. E. Veríssimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig. 2015. Software-defined networking: A comprehensive survey. *Proc. IEEE* 103, 1 (Jan. 2015), 14–76. DOI: http://dx.doi.org/10.1109/JPROC.2014.2371999

[6] R. Mijumbi, J. Serrat, J. L. Gorricho, N. Bouten, F. De Turck, and R. Boutaba. 2016. Network function virtualization: State-of-the-art and research challenges. *IEEE Commun. Surv. Tutor.* 18, 1 (2016), 236–262. DOI: http://dx.doi.org/10.1109/COMST.2015.2477041

[7] Daniel M. Batista, Gordon Blair, Fabio Kon, Raouf Boutaba, David Hutchison, Raj Jain, Ramachandran Ramjee, and Christian Esteve Rothenberg. 2015. Perspectives on software-defined networks: Interviews with five leading scientists from the networking community. *J. Internet Serv. Appl.* 6, 1 (2015), 1–10. DOI: http://dx.doi.org/10.1186/s13174-015-0035-3

[8] J. d. J. Gil Herrera and J. F. Botero Vega. 2016. Network functions virtualization: A survey. *IEEE Latin Am. Trans.* 14, 2 (Feb. 2016), 983–997. DOI: http://dx.doi.org/10.1109/TLA.2016.7437249

[9] Y. Li and M. Chen. 2015. Software-defined network function virtualization: A survey. *IEEE Access* 3 (2015), 2542–2553. DOI: http://dx.doi.org/10.1109/ACCESS.2015.2499271

[10] L. I. Barona López, Á L. Valdivieso Caraguay, L. J. García Villalba, and D. López. 2015. Trends on virtualisation with software defined networking and network function virtualisation. *IET Netw.* 4, 5 (2015), 255–263. DOI: http://dx.doi.org/10.1049/iet-net.2014.0117

[11] Barbara Kitchenham, O. Pearl Brereton, David Budgen, Mark Turner, John Bailey, and Stephen Linkman. 2009. Systematic literature reviews in software engineering—A systematic literature review. *Inf. Softw. Technol.* 51, 1 (2009), 7–15. DOI: http://dx.doi.org/10.1016/j.infsof.2008.09.009

[12] Barbara Kitchenham. 2004. *Procedures for Performing Systematic Reviews.* Technical Report TR/SE-0401. Keele University, Keele.
[13] ETSI. 2014. Network functions virtualisation (NFV)—Architectural framework. *ETSI GS NFV 002 V1.2.1* (Dec. 2014).
[14] ONF. 2015. OpenFlow Switch Specification, Version 1.3.5.
[15] ETSI. 2015. Network functions virtualisation (NFV), Ecosystem: Report on SDN usage in NFV architectural framework. *ETSI GS NFV-EVE 005 V1.1.1*
[16] G. Carella, J. Yamada, N. Blum, C. Lück, N. Kanamaru, N. Uchida, and T. Magedanz. 2015. Cross-layer service to network orchestration. In *Proceedings of the 2015 IEEE International Conference on Communications (ICC'15).* 6829–6835. DOI: http://dx.doi.org/10.1109/ICC.2015.7249414
[17] Fraunhofer FOKUS. 2016. OpenSDNCore—Reasearch and testbed for the carrier-grade nfv/sdn environment. Retrieved July 25, 2016 from http://www.opensdncore.org/.
[18] Justine Sherry, Shaddi Hasan, Colin Scott, Arvind Krishnamurthy, Sylvia Ratnasamy, and Vyas Sekar. 2012. Making middleboxes someone else's problem: Network processing as a cloud service. In *Proceedings of the ACM SIGCOMM 2012 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication (SIGCOMM'12).* ACM, New York, NY, 13–24. DOI: http://dx.doi.org/10.1145/2342356.2342359
[19] R. Cziva, S. Jouet, K. J. S. White, and D. P. Pezaros. 2015. Container-based network function virtualization for software-defined networks. In *Proceedings of the 2015 IEEE Symposium on Computers and Communication (ISCC'15).* 415–420. DOI: http://dx.doi.org/10.1109/ISCC.2015.7405550
[20] SK Telecom, Hewlett Packard, Samsung, and Telcoware. 2014. *PoC#23—E2E Orchestration of Virtualized LTE Core-network Functions and SDN-based Dynamic Service Chaining of VNFs Using VNF FG.* Technical Report. The European Telecommunications Standards Institute.
[21] J. Batalle, J. Ferrer Riera, E. Escalona, and J. A. Garcia-Espin. 2013. On the implementation of NFV over an OpenFlow infrastructure: Routing function virtualization. In *Proceedings of the IEEE SDN for Future Networks and Services (SDN4FNS'13).* 1–6. DOI: http://dx.doi.org/10.1109/SDN4FNS.2013.6702546
[22] Julius Schulz-Zander, Carlos Mayer, Bogdan Ciobotaru, Stefan Schmid, and Anja Feldmann. 2015. OpenSDWN: Programmatic control over home and enterprise WiFi. In *Proceedings of the 1st ACM SIGCOMM Symposium on Software Defined Networking Research (SOSR'15).* ACM, New York, NY, 16:1–16:12. DOI: http://dx.doi.org/10.1145/2774993.2775002
[23] Y. D. Lin, P. C. Lin, C. H. Yeh, Y. C. Wang, and Y. C. Lai. 2015. An extended SDN architecture for network function virtualization with a case study on intrusion prevention. *IEEE Netw.* 29, 3 (2015), 48–53. DOI: http://dx.doi.org/10.1109/MNET.2015.7113225
[24] B. Sonkoly, R. Szabo, D. Jocha, J. Czentye, M. Kind, and F. J. Westphal. 2015. UNIFYing cloud and carrier network resources: An architectural view. In *Proceedings of the 2015 IEEE Global Communications Conference (GLOBECOM'15).* 1–7. DOI: http://dx.doi.org/10.1109/GLOCOM.2015.7417869
[25] J. Deng, H. Hu, H. Li, Z. Pan, K. C. Wang, G. J. Ahn, J. Bi, and Y. Park. 2015. VNGuard: An NFV/SDN combination framework for provisioning and managing virtual firewalls. In *Proceedings of the 2015 IEEE Conference on Network Function Virtualization and Software Defined Network (NFV-SDN'15).* 107–114. DOI: http://dx.doi.org/10.1109/NFV-SDN.2015.7387414
[26] R. Cziva, S. Jouet, and D. P. Pezaros. 2015. GNFC: Towards network function cloudification. In *Proceedings of the 2015 IEEE Conference on Network Function Virtualization and Software Defined Network (NFV-SDN'15).* 142–148. DOI: http://dx.doi.org/10.1109/NFV-SDN.2015.7387419
[27] S. Van Rossem, W. Tavernier, B. Sonkoly, D. Colle, J. Czentye, M. Pickavet, and P. Demeester. 2015. Deploying elastic routing capability in an SDN/NFV-enabled environment. In *Proceedings of the 2015 IEEE Conference on Network Function Virtualization and Software Defined Network (NFV-SDN'15).* 22–24. DOI: http://dx.doi.org/10.1109/NFV-SDN.2015.7387398
[28] F. Callegati, W. Cerroni, C. Contoli, and G. Santandrea. 2015. Implementing dynamic chaining of virtual network functions in OpenStack platform. In *Proceedings of the 2015 17th International Conference on Transparent Optical Networks (ICTON'15).* 1–4. DOI: http://dx.doi.org/10.1109/ICTON.2015.7193561
[29] R. Cziva and D. P. Pezaros. 2017. Container network functions: Bringing NFV to the network edge. *IEEE Commun. Mag.* 55, 6 (2017), 24–31. DOI: http://dx.doi.org/10.1109/MCOM.2017.1601039
[30] Docker Inc. 2016. Docker Documentation. Retrieved July 25, 2016 from https://docs.docker.com/.
[31] Rackspace Cloud Computing. 2016. OpenStack Open Source Cloud Computing Software. Retrieved July 25, 2016 from https://www.openstack.org/.
[32] Linux Foundation. 2016. The OpenDaylight Plataform. Retrieved July 25, 2016 from http://www.opendaylight.org.
[33] NOXRepo.org. 2016. The POX Controller. Retrieved July 25, 2016 from http://www.noxrepo.org/pox/about-pox/.
[34] ETSI. 2013. Network functions virtualisation (NFV)—Use cases. *ETSI GS NFV 001 V1.1.1* (Out. 2013).

[35] Ivano Cerrato, Alex Palesandro, Fulvio Risso, Marc Suñé, Vinicio Vercellone, and Hagen Woesner. 2015. Toward dynamic virtualized network services in telecom operator networks. *Comput. Netw.* 92, 2 (2015), 380–395. DOI: http://dx.doi.org/10.1016/j.comnet.2015.09.028

[36] Yuri Gittik. 2014. White article—Distributed network functions virtualization (RAD).

[37] J. Soares, M. Dias, J. Carapinha, B. Parreira, and S. Sargento. 2014. Cloud4NFV: A platform for virtual network functions. In *Proceedings of the 2014 IEEE 3rd International Conference on Cloud Networking (CloudNet'14)*. 288–293. DOI: http://dx.doi.org/10.1109/CloudNet.2014.6969010

[38] J. Soares, C. Gonçalves, B. Parreira, P. Tavares, J. Carapinha, J. P. Barraca, R. L. Aguiar, and S. Sargento. 2015. Toward a telco cloud environment for service functions. *IEEE Commun. Mag.* 53, 2 (2015), 98–106. DOI: http://dx.doi.org/10.1109/MCOM.2015.7045397

[39] J. Schulz-Zander, C. Mayer, B. Ciobotaru, S. Schmid, and A. Feldmann. 2017. Unified programmability of virtualized network functions and software-defined wireless networks. *IEEE Trans. Netw. Service Manage.* 14, 4 (2017), 1–1. DOI: http://dx.doi.org/10.1109/TNSM.2017.2744807

[40] Lalith Suresh, Julius Schulz-Zander, Ruben Merz, Anja Feldmann, and Teresa Vazao. 2012. Towards programmable enterprise WLANS with odin. In *Proceedings of the 1st Workshop on Hot Topics in Software Defined Networks (HotSDN'12)*. ACM, New York, NY, 115–120. DOI: http://dx.doi.org/10.1145/2342441.2342465

[41] J. Vestin and A. Kassler. 2015. QoS enabled WiFi MAC layer processing as an example of a NFV service. In *Proceedings of the 2015 1st IEEE Conference on Network Softwarization (NetSoft'15)*. 1–9. DOI: http://dx.doi.org/10.1109/NETSOFT.2015.7116164

[42] P. Dely, J. Vestin, A. Kassler, N. Bayer, H. Einsiedler, and C. Peylo. 2012. CloudMAC: An OpenFlow based architecture for 802.11 MAC layer processing in the cloud. In *Proceedings of the 2012 IEEE Globecom Workshops*. 186–191. DOI: http://dx.doi.org/10.1109/GLOCOMW.2012.6477567

[43] A. Gupta and R. K. Jha. 2015. A survey of 5G network: Architecture and emerging technologies. *IEEE Access* 3 (2015), 1206–1232. DOI: http://dx.doi.org/10.1109/ACCESS.2015.2461602

[44] 2017. ITU towards "IMT for 2020 and beyond." Retrieved September 28, 2017 from http://www.itu.int/en/ITU-R/study-groups/rsg5/rwp5d/imt-2020/Pages/default.aspx.

[45] 2017. Verizon 5G Technical Forum. Retrieved September 28, 2017 from http://www.5gtf.org/.

[46] 2017. 5G Infrastructure Public Private Partnership–5G PPP. Retrieved September 28, 2017 https://5g-ppp.eu/.

[47] 5G PPP Architecture Working Group. 2016. *View on 5G Architecture*. Technical Report.

[48] 5G PPP Architecture Working Group. 2016. *5G PPP Use Cases and Performance Evaluation Models*. Technical Report.

[49] Huawei Technologies. 2015. *5G Network Architecture-A High Level View*. Technical Report.

[50] 5G PPP Architecture Working Group. 2017. *Vision on Software Networks and 5G*. Technical Report.

[51] Y. Kyung, T. M. Nguyen, K. Hong, J. Park, and J. Park. 2015. Software defined service migration through legacy service integration into 4G networks and future evolutions. *IEEE Commun. Mag.* 53, 9 (Sep. 2015), 108–114. DOI: http://dx.doi.org/10.1109/MCOM.2015.7263353

[52] Telecom Italia, Nokia Networks, EXFO, Coriant, and Aalto University. 2015. *PoC#26—Virtual EPC with SDN Function in Mobile Backhaul Networks*. Technical Report. The European Telecommunications Standards Institute.

[53] China Unicom, ZTE Corporation, and Hewlett-Packard. 2015. *PoC#27—VoLTE Service Based on vEPC and vIMS Architecture*. Technical Report. The European Telecommunications Standards Institute.

[54] E. Haleplidis, D. Joachimpillai, J. H. Salim, D. Lopez, J. Martin, K. Pentikousis, S. Denazis, and O. Koufopavlou. 2014. ForCES applicability to SDN-enhanced NFV. In *Proceedings of the 2014 3rd European Workshop on Software Defined Networks*. 43–48. DOI: http://dx.doi.org/10.1109/EWSDN.2014.27

[55] Arsany Basta, Andreas Blenk, Marco Hoffmann, Hans Jochen Morper, Klaus Hoffmann, and Wolfgang Kellerer. 2014. SDN and NFV dynamic operation of LTE EPC gateways for time-varying traffic patterns. In *Mobile Networks and Management*, Ramón Agüero, Thomas Zinner, Rossitza Goleva, Andreas Timm-Giel, and Phuoc Tran-Gia (Eds.). Number 141 in Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering. Springer International Publishing, 63–76.

[56] V. G. Nguyen and Y. H. Kim. 2014. Slicing the next mobile packet core network. In *Proceedings of the 2014 11th International Symposium on Wireless Communications Systems (ISWCS'14)*. 901–904. DOI: http://dx.doi.org/10.1109/ISWCS.2014.6933481

[57] J. Costa-Requena, J. L. Santos, V. F. Guasch, K. Ahokas, G. Premsankar, S. Luukkainen, O. L. Pérez, M. U. Itzazelaia, I. Ahmad, M. Liyanage, M. Ylianttila, and E. M. de Oca. 2015. SDN and NFV integration in generalized mobile network architecture. In *Proceedings of the 2015 European Conference on Networks and Communications (EuCNC'15)*. 154–158. DOI: http://dx.doi.org/10.1109/EuCNC.2015.7194059

[58] A. M. Medhat, G. Carella, J. Mwangama, and N. Ventura. 2015. Multi-tenancy for virtualized network functions. In *Proceedings of the 2015 1st IEEE Conference on Network Softwarization (NetSoft'15)*. 1–6. DOI: http://dx.doi.org/10.1109/NETSOFT.2015.7116177

[59] I. Ahmad, M. Liyanage, S. Namal, M. Ylianttila, A. Gurtov, M. Eckert, T. Bauschert, Z. Faigl, L. Bokor, E. Saygun, O. L. Akyildiz, H. A. and, M. U. Itzazelaia, B. Ozbek, and A. Ulas. 2016. New concepts for traffic, resource and mobility management in software-defined mobile networks. In *Proceedings of the 2016 12th Annual Conference on Wireless On-demand Network Systems and Services (WONS'16)*. 1–8.

[60] A. Tawbeh, H. Safa, and A. R. Dhaini. 2017. A hybrid SDN/NFV architecture for future LTE networks. In *Proceedings of the 2017 IEEE International Conference on Communications (ICC'17)*. 1–6. DOI: http://dx.doi.org/10.1109/ICC.2017.7997391

[61] X. An, W. Kiess, and D. Perez-Caparros. 2014. Virtualization of cellular network EPC gateways based on a scalable SDN architecture. In *Proceedings of the 2014 IEEE Global Communications Conference*. 2295–2301. DOI: http://dx.doi.org/10.1109/GLOCOM.2014.7037150

[62] Telefonica, Vodafone, Radware, HP, and Melanox. 2016. *PoC#13—SteerFlow: Multi-Layered Traffic Steering for Gi-LAN*. Technical Report. The European Telecommunications Standards Institute.

[63] Telenor, Vodafone, Hewlett Packard Enterprise, ImVision Tech, Mavenir, Redhat, and Altiostar. 2016. *PoC#34—SDN Enabled Virtual EPC Gateway*. Technical Report. The European Telecommunications Standards Institute.

[64] P. Grønsund, K. Mahmood, G. Millstein, A. Noy, G. Solomon, and A. Sahai. 2015. A solution for SGi-LAN services virtualization using NFV and SDN. In *Proceedings of the 2015 European Conference on Networks and Communications (EuCNC'15)*. 408–412. DOI: http://dx.doi.org/10.1109/EuCNC.2015.7194108

[65] J. Ordonez-Lucena, P. Ameigeiras, D. Lopez, J. J. Ramos-Munoz, J. Lorca, and J. Folgueira. 2017. Network slicing for 5G with SDN/NFV: Concepts, architectures, and challenges. *IEEE Commun. Mag.* 55, 5 (May 2017), 80–87. DOI: http://dx.doi.org/10.1109/MCOM.2017.1600935

[66] R. Munoz, R. Vilalta, R. Casellas, R. Martinez, T. Szyrkowiec, A. Autenrieth, V. Lopez, and D. Lopez. 2015. Integrated SDN/NFV management and orchestration architecture for dynamic deployment of virtual SDN control instances for virtual tenant networks [invited]. *IEEE/OSA J. Opt. Commun. Netw.* 7, 11 (Nov. 2015), B62–B70. DOI: http://dx.doi.org/10.1364/JOCN.7.000B62

[67] R. Muñoz, R. Vilalta, R. Casellas, R. Martínez, T. Szyrkowiec, A. Autenrieth, V. López, and D. López. 2015. SDN/NFV orchestration for dynamic deployment of virtual SDN controllers as VNF for multi-tenant optical networks. In *Proceedings of the Optical Fiber Communications Conference and Exhibition (OFC'15), 2015*. 1–3.

[68] R. Vilalta, A. Mayoral, R. Muñoz, R. Casellas, and R. Martínez. 2015. The SDN/NFV cloud computing platform and transport network of the ADRENALINE testbed. In *Proceedings of the 2015 1st IEEE Conference on Network Softwarization (NetSoft'15)*. 1–5. DOI: http://dx.doi.org/10.1109/NETSOFT.2015.7116150

[69] R. Vilalta, A. Mayoral, R. Muñoz, R. Casellas, and R. Martínez. 2015. Multi-tenant transport networks with SDN/NFV. In *Proceedings of the 2015 European Conference on Optical Communication (ECOC'15)*. 1–3. DOI: http://dx.doi.org/10.1109/ECOC.2015.7341931

[70] R. Vilalta, A. Mayoral, R. Muñoz, R. Casellas, and R. Martínez. 2016. Multitenant transport networks with SDN/NFV. *J. Lightwave Technol.* 34, 6 (Mar. 2016), 1509–1515. DOI: http://dx.doi.org/10.1109/JLT.2015.2508044

[71] R. Vilalta, A. Mayoral, V. Lopez, V. Uceda, R. Casellas, R. Martinez, R. Munoz, A. Aguado, J. Marhuenda, R. Nejabati, D. Simeonidou, N. Yoshikane, T. Tsuritani, I. Morita, T. Szyrkowiec, and A. Autenrieth. 2016. Peer SDN orchestration: End-to-end connectivity service provisioning through multiple administrative domains. In *Proceedings of the 42nd European Conference on Optical Communication (ECOC'16)*. 1–3.

[72] Ian F. Akyildiz, Shih-Chun Lin, and Pu Wang. 2015. Wireless software-defined networks (W-SDNs) and network function virtualization (NFV) for 5G cellular systems: An overview and qualitative evaluation. *Comput. Netw.* 93, 1 (2015), 66–79. DOI: http://dx.doi.org/10.1016/j.comnet.2015.10.013

[73] J. Mwangama, N. Ventura, A. Willner, Y. Al-Hazmi, G. Carella, and T. Magedanz. 2015. Towards mobile federated network operators. In *Proceedings of the 2015 1st IEEE Conference on Network Softwarization (NetSoft'15)*. 1–6. DOI: http://dx.doi.org/10.1109/NETSOFT.2015.7116187

[74] R. Casellas, R. Muñoz, R. Vilalta, and R. Martínez. 2016. Orchestration of IT/cloud and networks: From inter-DC interconnection to SDN/NFV 5G services. In *Proceedings of the 2016 International Conference on Optical Network Design and Modeling (ONDM'16)*. 1–6. DOI: http://dx.doi.org/10.1109/ONDM.2016.7494060

[75] R. Vilalta, A. Mayoral, R. Casellas, R. Martínez, and R. Muñoz. 2016. SDN/NFV orchestration of multi-technology and multi-domain networks in cloud/fog architectures for 5g services. In *Proceedings of the 2016 21st OptoElectronics and Communications Conference (OECC'16) held jointly with 2016 International Conference on Photonics in Switching (PS'16)*. 1–3.

[76] A. Mayoral, R. Vilalta, R. Casellas, R. Martinez, and R. Munoz. 2016. Multi-tenant 5G network slicing architecture with dynamic deployment of virtualized tenant management and orchestration (MANO) instances. In *Proceedings of the ECOC 2016; 42nd European Conference on Optical Communication*. 1–3.

[77] R. Martínez, A. Mayoral, R. Vilalta, R. Casellas, R. Muñoz, S. Pachnicke, T. Szyrkowiec, and A. Autenrieth. 2017. Integrated SDN/NFV orchestration for the dynamic deployment of mobile virtual backhaul networks over a

multilayer (packet/optical) aggregation infrastructure. *IEEE/OSA J. Opt. Commun. Netw.* 9, 2 (Feb. 2017), A135–A142. DOI: http://dx.doi.org/10.1364/JOCN.9.00A135

[78] R. Muñoz, L. Nadal, R. Casellas, M. S. Moreolo, R. Vilalta, J. M. Fàbrega, R. Martínez, A. Mayoral, and F. J. Vílchez. 2017. The ADRENALINE testbed: An SDN/NFV packet/optical transport network and edge/core cloud platform for end-to-end 5G and IoT services. In *Proceedings of the 2017 European Conference on Networks and Communications (EuCNC'17)*. 1–5. DOI: http://dx.doi.org/10.1109/EuCNC.2017.7980775

[79] R. Vilalta, A. Mayoral, R. Casellas, R. Martínez, and R. Muñoz. 2016. Experimental demonstration of distributed multi-tenant cloud/fog and heterogeneous SDN/NFV orchestration for 5G services. In *Proceedings of the 2016 European Conference on Networks and Communications (EuCNC'16)*. 52–56. DOI: http://dx.doi.org/10.1109/EuCNC.2016.7561003

[80] ETSI. 2016. Mobile edge computing (MEC): Technical requirements. *ETSI GS MEC 002 v1.1.1* (Mar. 2016).

[81] ETSI. 2016. Mobile edge computing (MEC): Framework and reference architecture. *ETSI GS MEC 003 v1.1.1* (Mar. 2016).

[82] Rodrigo Roman, Javier Lopez, and Masahiro Mambo. 2018. Mobile edge computing, Fog et al.: A survey and analysis of security threats and challenges. *Future Generation Computer Systems* 78, 2 (2018), 680–698. DOI: https://doi.org/10.1016/j.future.2016.11.009

[83] 5G PPP Architecture Working Group. 2015. *5G Vision*. Technical Report.

[84] EU SELFNET Project. 2016. Framework for Self-Organized Network Management in Virtualized and Software Defined Networks, Project reference: ICT-2014-2/671672. Funded under H2020. Retrieved July 25, 2016 from http://www.selfnet-5g.eu/.

[85] Pedro Neves, Rui Calé, Mário Rui Costa, Carlos Parada, Bruno Parreira, Jose Alcaraz-Calero, Qi Wang, James Nightingale, Enrique Chirivella-Perez, Wei Jiang, Hans Dieter Schotten, Konstantinos Koutsopoulos, Anastasius Gavras, and Maria João Barros. 2016. The SELFNET approach for autonomic management in an NFV/SDN networking paradigm. *Int. J. Distrib. Sen. Netw.* 2016, Article 2 (Jan. 2016), 1 pages. DOI: http://dx.doi.org/10.1155/2016/2897479

[86] Pedro Neves, Rui Calé, Mário Costa, Gonçalo Gaspar, Jose Alcaraz-Calero, Qi Wang, James Nightingale, Giacomo Bernini, Gino Carrozzo, Ángel Valdivieso, Luis Javier García Villalba, Maria Barros, Anastasius Gravas, José Santos, Ricardo Maia, and Ricardo Preto. 2017. Future mode of operations for 5G—The SELFNET approach enabled by SDN/NFV. *Comput. Standards Interfaces* 54, Part 4 (2017), 229–246. DOI: http://dx.doi.org/10.1016/j.csi.2016.12.008SI: Standardization SDN & NFV.

[87] Verizon. 2016. SDN-NFV reference architecture. *Verizon Network Infrastructure Planning* (Feb. 2016).

[88] Telefonica I+D. 2016. OpenMANO - A ETSI NFV compliant Management and Orchestration (MANO). Retrieved July 25, 2016 from https://github.com/nfvlabs/openmano.

[89] Fraunhofer FOKUS. 2016. OpenBaton—A ETSI NFV compliant Network Function Virtualization Orchestrator (NFVO). Retrieved July 25, 2016 from http://openbaton.github.io/.

[90] Big Switch Networks. 2016. The Floodlight Project. July 25, 2016 from http://www.projectfloodlight.org/floodlight/.

[91] Open Network Foundation (ONF). 2017. The ONOS Project. October 2, 2017 from http://onosproject.org/.

[92] Nippon Telegraph and Telephone (NTT). 2016. Ryu SDN Framework. Retrieved July 25, 2016 from https://osrg.github.io/ryu/.

[93] W. Shen, M. Yoshida, K. Minato, and W. Imajuku. 2015. vConductor: An enabler for achieving virtual network integration as a service. *IEEE Commun. Mag.* 53, 2 (2015), 116–124. DOI: http://dx.doi.org/10.1109/MCOM.2015.7045399

[94] R. Vilalta, R. Muñoz, A. Mayoral, R. Casellas, R. Martínez, V. López, and D. López. 2015. Transport network function virtualization. *J. Lightwave Technol.* 33, 8 (Apr. 2015), 1557–1564. DOI: http://dx.doi.org/10.1109/JLT.2015.2390655

[95] A. Mohammadkhan, G. Liu, W. Zhang, K. K. Ramakrishnan, and T. Woodv. 2015. Protocols to support autonomy and control for NFV in software defined networks. In *Proceedings of the 2015 IEEE Conference on Network Function Virtualization and Software Defined Network (NFV-SDN'15)*. 163–169. DOI: http://dx.doi.org/10.1109/NFV-SDN.2015.7387422

[96] A. Lombardo, A. Manzalini, G. Schembra, G. Faraci, C. Rametta, and V. Riccobene. 2015. An open framework to enable NetFATE (network functions at the edge). In *Proceedings of the 2015 1st IEEE Conference on Network Softwarization (NetSoft'15)*. 1–6. DOI: http://dx.doi.org/10.1109/NETSOFT.2015.7116179

[97] L. Mamatas, S. Clayman, and A. Galis. 2015. A service-aware virtualized software-defined infrastructure. *IEEE Commun. Mag.* 53, 4 (Apr. 2015), 166–174. DOI: http://dx.doi.org/10.1109/MCOM.2015.7081091

[98] Q. Duan, N. Ansari, and M. Toy. 2016. Software-defined network virtualization: An architectural framework for integrating SDN and NFV for service provisioning in future networks. *IEEE Netw.* 30, 5 (Sep. 2016), 10–16. DOI: http://dx.doi.org/10.1109/MNET.2016.7579021

[99] AT&T, Telecom Italia, Netronome, Intel, ServiceMesh, PLUMgrid, and Cisco Systems. 2015. *PoC#16—NFVIaaS with Secure, SDN-controlled WAN Gateway*. Technical Report. The European Telecommunications Standards Institute.

[100] Linux Foundation. 2016. The Xen Project. Retrieved July 25, 2016 from https://www.xenproject.org/.

[101] F. Lucrezia, G. Marchetto, F. Risso, and V. Vercellone. 2015. Introducing network-aware scheduling capabilities in OpenStack. In *Proceedings of the 2015 1st IEEE Conference on Network Softwarization (NetSoft'15)*. 1–5. DOI: http://dx.doi.org/10.1109/NETSOFT.2015.7116155

[102] K. Giotis, Y. Kryftis, and V. Maglaris. 2015. Policy-based orchestration of NFV services in software-defined networks. In *Proceedings of the 2015 1st IEEE Conference on Network Softwarization (NetSoft'15)*. 1–5. DOI: http://dx.doi.org/10.1109/NETSOFT.2015.7116145

[103] W. Ding, W. Qi, J. Wang, and B. Chen. 2015. OpenSCaaS: An open service chain as a service platform toward the integration of SDN and NFV. *IEEE Netw.* 29, 3 (2015), 30–35. DOI: http://dx.doi.org/10.1109/MNET.2015.7113222

[104] J. Lai, Q. Fu, and T. Moors. 2015. Rapid IP rerouting with SDN and NFV. In *Proceedings of the 2015 IEEE Global Communications Conference (GLOBECOM'15)*. 1–7. DOI: http://dx.doi.org/10.1109/GLOCOM.2015.7417318

[105] H. Wang, S. Chen, H. Xu, M. Ai, and Y. Shi. 2015. SoftNet: A software defined decentralized mobile network architecture toward 5G. *IEEE Netw.* 29, 2 (Mar. 2015), 16–22. DOI: http://dx.doi.org/10.1109/MNET.2015.7064898

[106] M. Xia, M. Shirazipour, Y. Zhang, H. Green, and A. Takacs. 2015. Optical service chaining for network function virtualization. *IEEE Commun. Mag.* 53, 4 (Apr. 2015), 152–158. DOI: http://dx.doi.org/10.1109/MCOM.2015.7081089

[107] Telefonica, Sprint, 6WIND, Dell, EnterpriseWeb, Mellanox, Metaswitch, Overture Networks, Qosmos, and Aeroflex. 2014. *PoC#1—CloudNFV Open NFV Framework Project.* Technical Report. The European Telecommunications Standards Institute.

[108] NTT, Cisco, HP, and Juniper Networks. 2014. *PoC#2—Service Chaining for NW Function Selection in Carrier Networks.* Technical Report. The European Telecommunications Standards Institute.

[109] Deutsche Telekom, Ericsson, x-ion GmbH, and Deutsche Telekom Innovation Laboratories. 2014. *PoC#8—Automated Network Orchestration.* Technical Report. The European Telecommunications Standards Institute.

[110] Linda Dunbar and Cathy Zhang. 2015. *PoC#28 - SDN Controlled VNF Forwarding Graph.* Technical Report. The European Telecommunications Standards Institute.

[111] Telstra, Hewlett-Packard, Alcatel Lucent, and F5 Networks. 2016. *PoC#38 - Full ISO 7-layer Stack Fulfilment, Activation and Orchestration of VNFs in Carrier Networks.* Technical Report. The European Telecommunications Standards Institute.

[112] F. Callegati, W. Cerroni, C. Contoli, and F. Foresta. 2017. Performance of intent-based virtualized network infrastructure management. In *Proceedings of the 2017 IEEE International Conference on Communications (ICC'17)*. 1–6. DOI: http://dx.doi.org/10.1109/ICC.2017.7997431

[113] Evangelos Haleplidis, Jamal Hadi Salim, Spyros Denazis, and Odysseas Koufopavlou. 2014. Towards a network abstraction model for SDN. *J. Netw. Syst. Manage.* 23, 2 (Jul. 2014), 309–327. DOI: http://dx.doi.org/10.1007/s10922-014-9319-3

[114] Wooseong Kim. 2015. Toward network function virtualization for cognitive wireless mesh networks: A TCP case study. *J. Wireless Commun. Netw.* 2015, 1 (Oct. 2015), 1–16. DOI: http://dx.doi.org/10.1186/s13638-015-0450-y

[115] Evelyne Roch. 2015. *PoC#21 - Network Intensive and Compute Intensive Hardware Acceleration.* Technical Report. The European Telecommunications Standards Institute. Retrieved from https://docbox.etsi.org/ISG/NFV/TST/05-CONTRIBUTIONS/2015//NFVTST(15)000111r1_PoC_21__Network_Intensive_and_Compute_Intensive_Hardware_Acc.docx.

[116] Jon Matias, Jokin Garay, Nerea Toledo, Juanjo Unzilla, and Eduardo Jacob. 2015. Toward an SDN-enabled NFV architecture. *IEEE Commun. Mag.* 53, 4 (2015), 187–193. DOI: http://dx.doi.org/10.1109/MCOM.2015.7081093

[117] Guozhen Cheng, Hongchang Chen, Hongchao Hu, Zhiming Wang, and Julong Lan. 2015. Enabling network function combination via service chain instantiation. *Comput. Netw.* 92, 2 (2015), 396–407. DOI: http://dx.doi.org/10.1016/j.comnet.2015.09.015

[118] G. M. Saridis, S. Peng, Y. Yan, A. Aguado, B. Guo, M. Arslan, C. Jackson, W. Miao, N. Calabretta, F. Agraz, S. Spadaro, G. Bernini, N. Ciulli, G. Zervas, R. Nejabati, and D. Simeonidou. 2016. Lightness: A function-virtualizable software defined data center network with all-optical circuit/packet switching. *J. Lightwave Technol.* 34, 7 (Apr. 2016), 1618–1627. DOI: http://dx.doi.org/10.1109/JLT.2015.2509476

[119] A. Doria, J. Hadi Salim, R. Haas, H. Khosravi, W. Wang, L. Dong, R. Gopal, and J. Halpern. 2010. *Forwarding and Control Element Separation (ForCES) Protocol Specification.* RFC 5810. RFC Editor. https://www.rfc-editor.org/info/rfc5810.

[120] H. D. Mustafa, B. M. Baveja, S. Vijayan, S. N. Merchant, and U. B. Desai. 2015. Replicating the geographical cloud: Provisioning omnipresence, omniscience and omnipotence. *Fut. Generat. Comput. Syst.* 47 (2015), 1–15. DOI: http://dx.doi.org/10.1016/j.future.2014.12.004

[121] Linux Foundation. 2016. Data Plane Development Kit (DPDK) Documentation. Retrieved July 25, 2016 from http://dpdk.org/doc.

[122] J. Hwang, K. K. Ramakrishnan, and T. Wood. 2015. NetVM: High performance and flexible networking using virtualization on commodity platforms. *IEEE Trans. Netw. Service Manage.* 12, 1 (Mar. 2015), 34–47. DOI: http://dx.doi.org/10.1109/TNSM.2015.2401568

[123] Joao Martins, Mohamed Ahmed, Costin Raiciu, Vladimir Olteanu, Michio Honda, Roberto Bifulco, and Felipe Huici. 2014. ClickOS and the art of network function virtualization. In *Proceedings of the 11th USENIX Conference on Networked Systems Design and Implementation (NSDI'14)*. USENIX Association, Berkeley, CA, 459–473. http://dl.acm.org/citation.cfm?id=2616448.2616491.

[124] ETSI. 2016. Network functions virtualisation (NFV)—Virtualisation technologies. Report on the application of different virtualisation technologies in the NFV framework. *ETSI GS NFV-EVE 004 V1.1.1* (Mar. 2016).

[125] Abhishek Verma, Luis Pedrosa, Madhukar R. Korupolu, David Oppenheimer, Eric Tune, and John Wilkes. 2015. Large-scale cluster management at Google with Borg. In *Proceedings of the European Conference on Computer Systems (EuroSys'15)*.

[126] Malte Schwarzkopf, Andy Konwinski, Michael Abd-El-Malek, and John Wilkes. 2013. Omega: Flexible, scalable schedulers for large compute clusters. In *Proceedings of the SIGOPS European Conference on Computer Systems (EuroSys'13)*. 351–364.

[127] Benjamin Hindman, Andy Konwinski, Matei Zaharia, Ali Ghodsi, Anthony D. Joseph, Randy Katz, Scott Shenker, and Ion Stoica. 2011. Mesos: A platform for fine-grained resource sharing in the data center. In *Proceedings of the 8th USENIX Conference on Networked Systems Design and Implementation (NSDI'11)*. USENIX Association, Berkeley, CA, 295–308. http://dl.acm.org/citation.cfm?id=1972457.1972488

[128] X. Foukas, G. Patounas, A. Elmokashfi, and M. K. Marina. 2017. Network slicing in 5G: Survey and challenges. *IEEE Commun. Mag.* 55, 5 (May 2017), 94–100. DOI: http://dx.doi.org/10.1109/MCOM.2017.1600951

[129] X. Li, M. Samaka, H. A. Chan, D. Bhamare, L. Gupta, C. Guo, and R. Jain. 2017. Network slicing for 5G: Challenges and opportunities. *IEEE Internet Comput.* 21, 5 (2017), 20–27. DOI: http://dx.doi.org/10.1109/MIC.2017.3481355

Received December 2016; revised October 2017; accepted December 2017