SDN/NFV-Based Mobile Packet Core Network Architectures: A Survey

Van-Giang Nguyen, *Student Member, IEEE*, Anna Brunstrom, *Member, IEEE*,
Karl-Johan Grinnemo, *Senior Member, IEEE*, and Javid Taheri, *Member, IEEE*

***Abstract*—The emergence of two new technologies, namely, software defined networking (SDN) and network function virtualization (NFV), have radically changed the development of network functions and the evolution of network architectures. These two technologies bring to mobile operators the promises of reducing costs, enhancing network flexibility and scalability, and shortening the time-to-market of new applications and services. With the advent of SDN and NFV and their offered benefits, the mobile operators are gradually changing the way how they architect their mobile networks to cope with ever-increasing growth of data traffic, massive number of new devices and network accesses, and to pave the way toward the upcoming fifth generation networking. This survey aims at providing a comprehensive survey of state-of-the-art research work, which leverages SDN and NFV into the most recent mobile packet core network architecture, evolved packet core. The research work is categorized into smaller groups according to a proposed four-dimensional taxonomy reflecting the: 1) architectural approach, 2) technology adoption, 3) functional implementation, and 4) deployment strategy. Thereafter, the research work is exhaustively compared based on the proposed taxonomy and some added attributes and criteria. Finally, this survey identifies and discusses some major challenges and open issues, such as scalability and reliability, optimal resource scheduling and allocation, management and orchestration, and network sharing and slicing that raise from the taxonomy and comparison tables that need to be further investigated and explored.**

***Index Terms*—Software defined networking, network function virtualization, mobile packet core, evolved packet core, future mobile networking, 5G networking, network slicing.**

I. Introduction

OVER the last decade, we have witnessed an explosion of mobile devices along with the appearance and emergence of new types of applications and services such as augmented reality, virtual reality, etc. Having these new services deployed over the mobile network along with a massive number of mobile devices have caused an exponential increase in mobile data traffic usage. According to a Cisco forecast, global mobile data traffic is expected to reach approximately 31 Exabytes per month by 2020, i.e., roughly a ten-time increase since 2015 [1]. This significant growth in mobile traffic and new services are pushing mobile network operators to upgrade their systems and invest in the infrastructure in order to meet new requirements and to satisfy their customers' demands. Such new requirements include requirements for the next generation of mobile networks the so-called fifth generation (5G) network, which is expected to achieve an extremely high data rate, ultra-low latency, high user mobility, ultra-reliable communication, etc. [2]. Some examples of newly defined services and use cases which will appear in the 5G ecosystem are autonomous driving, augmented and virtual reality, tactile Internet, smart city, smart environment, etc.

Historically, mobile cellular communication networks have been evolved through four generations, starting from being a circuit-based analog telephony system in 1G, to become a partially packet-based system in 2G and 3G, and finally became an all-IP packet-based 4G system a few years ago. During this evolution process, there are a number of changes that have been made to be able to provide users better quality of service and experience. One of the most recent mobile cellular network technologies in 4G is the Long Term Evolution (LTE) developed by the 3rd Generation Partnership Project (3GPP) organization [3]. With the development of LTE in the radio access network part, the 3GPP also introduced a new mobile core network architecture called Evolved Packet Core (EPC), which is able to interoperate with the legacy 2G and 3G systems. Taken together, the development of LTE and the introduction of EPC allows mobile users to access to multimedia resources in external packet data networks such as the Internet. Although the EPC has been simplified in comparison to its predecessors in 2G and 3G, it has still a number of limitations that impose challenges to the mobile network operators to upgrade their architectures. (1) All EPC entities, including Mobility Management Entity (MME), Serving Gateway (SGW), Packet Data Network Gateway (PGW), Home Subscriber Server (HSS) and Policy control and Charging Rule Functions (PCRF), are typically based on customized hardware which are usually configured, deployed and provisioned in a static and cost-ineffective manner. This type of functional design and configuration results in inflexibility of network management while hardware-based deployment results in increasing capital expense for the mobile network operators. (2) Being tightly integrated into a hardware-based platform also limits the elasticity, on-demand provisioning process, and network deployment cycle.

Manuscript received September 23, 2016; revised February 2, 2017; accepted March 23, 2017. Date of publication April 5, 2017; date of current version August 21, 2017. This work was supported by the High Quality Networked Services in a Mobile World Project through the Knowledge Foundation of Sweden. *(Corresponding author: Van-Giang Nguyen.)*

The authors are with the Department of Mathematics and Computer Science, Karlstad University, 65188 Karlstad, Sweden (e-mail: giang.nguyen@kau.se; anna.brunstrom@kau.se; karl-johan.grinnemo@kau.se; javid.taheri@kau.se).

Digital Object Identifier 10.1109/COMST.2017.2690823

Indeed, the current EPC and its entities are being dimensioned and over-provisioned based on peak-load demands which are predicted and foreseen for a long-term period, typically few years. (3) The control and data planes in the EPC network architecture have not been completely decoupled; they are still tightly coupled at SGW and PGW. Such a coupled design contributes to the inflexibility of network management, and limits the scalability of the network. Since the control plane and data plane have different performance requirements, the control plane requires low latency for processing signaling messages, whereas the data plane requires high throughput for processing user data traffic, it is necessary to decouple these planes to be able to get them scaled independently and efficiently during the provisioning process. (4) The data plane of the current LTE/EPC architecture is too centralized. Indeed, all uplink traffic from user equipments (UEs) has always to traverse along a north-south path through the radio access, mobile backhaul and then enters IP networks via a small number of centralized PGWs even if some UEs are just communicating with local application servers. Such a hierarchical deployment results in inefficiency of data packet forwarding and mobility management, high latency, thus not suitable to accomplish the aforementioned 5G requirements. Recently, the advent of some cutting-edge technologies such as cloud computing, mobile edge computing, network virtualization, Software Defined Networking (SDN) [13], and Network Function Virtualization (NFV) [14] have changed the way in which network functions and devices are implemented, and also changed the way in which the network architectures are constructed. More specifically, the network equipment or device is now changing from closed, vendor specific to open and generic with SDN technology, which enables the separation of control and data planes, and allows networks to be programmed by using open interfaces. With NFV, network functions previously realized in costly hardware platforms are now implemented as software appliances placed on low-cost commodity hardwares or running in the cloud computing environment. These two technologies, together with cloud computing and network virtualization, bring to mobile operators the promises of reducing Capital Expense (CAPEX) [15] and Operation Expense (OPEX) [16], enhancing network flexibility and scalability, and shortening the time-to-market of new applications and services. In addition, with the combination of SDN and NFV, it is possible to bring parts of the mobile packet core network closer to the edge or users, thus shortening the end-to-end network latency. More importantly, SDN and NFV have been defined as key drivers in the design of the 5G network architecture [2], [17]–[21]. In 5G systems, networks will be further abstracted into different network slices forming end-to-end logically isolated networks dedicated to different types of services with different characteristics and requirements such as a slice of massive IoT devices, a slice of smartphones or a slice of autonomous cars, etc. [2]. This capability of slicing network is driven by means of SDN and NFV [22], [23] and the network slicing technology is now set to play a key role in meeting the demands of 5G use cases and underlying cost requirements.

Aligned with on-going SDN and NFV research activities in some open and standard organizations such as Open Networking Foundation (ONF) [24], European Telecommunications Standards Institutes (ETSI) NFV Industry Specification Group (ISG) [25], recently the 3GPP mobile standards organisation has also shifted their focus towards SDN and NFV in the development of the next generation mobile network architecture [26]. They have started working on these concepts to be released in Release 14, which is the first 3GPP standard release to introduce 5G. On the aspect of SDN, the 3GPP architecture working group SA2 has initiated a study item, control and user plane separation (CUPS) [27], on the separation of the control and user plane for SGW and PGW entities, so that the user plane functions can be placed flexibly while the control plane functions still remain centralized. On the aspect of NFV, 3GPP telecoms management working group SA5 in liaison with ETSI NFV have established a study item on network management of virtualized networks [28]. It focuses on the end-to-end management and orchestration solutions for mobile packet core networks and covers lifecycle, configuration, fault and performance management of 3GPP virtualized network functions. For network slicing, 3GPP SA2 is in the early stages of a study on an architecture for next generation system in which a network slicing architecture and related issues are being defined [26]. Thus, there is an urgent need to study the fundamental architectural principles and approaches underlying a new generation of mobile packet core (MPC)[1] network architectures with the adoption of SDN and NFV technologies. To this end, we present in this paper a literature review of all current SDN- and NFV-based MPC research initiatives by approaching them from different points of view: architectural approach, technology adoption, functional implementation and deployment strategy, which we believe are the most important aspects while designing and developing a system.

A. Scope and Contributions

The main objectives of this paper are to provide a comprehensive survey of the up-to-date solutions adopted SDN and NFV into the current EPC network architecture, and to provide several guidelines for future relevant investigations in this field. We focus on the EPC network architecture because it is the most recent core network architecture that is serving 4G services. Another reason is that the evolution of the EPC architecture is currently being considered to be one of two major options for designing the core network part of the upcoming 5G network within the 3GPP standardization group. In the past, different papers have been proposed to survey the benefits and adoption of SDN and NFV for wireless networks in general and for Mobile Cellular Networks (MCN) in particular. The scope of these survey papers in comparison to our paper (in red circle) is illustrated in Fig. 1. As shown in this figure, most of the related surveys [4]–[9]

[1] In this paper, we aim at describing the research work showing developments of the current MPC with SDN and NFV. The current MPC refers to the most recent MPC, named Evolved Packet Core or EPC. However, we intend to keep the use of MPC as a general terminology. For that reason, sometimes we use these two terminologies interchangeably throughout the paper.

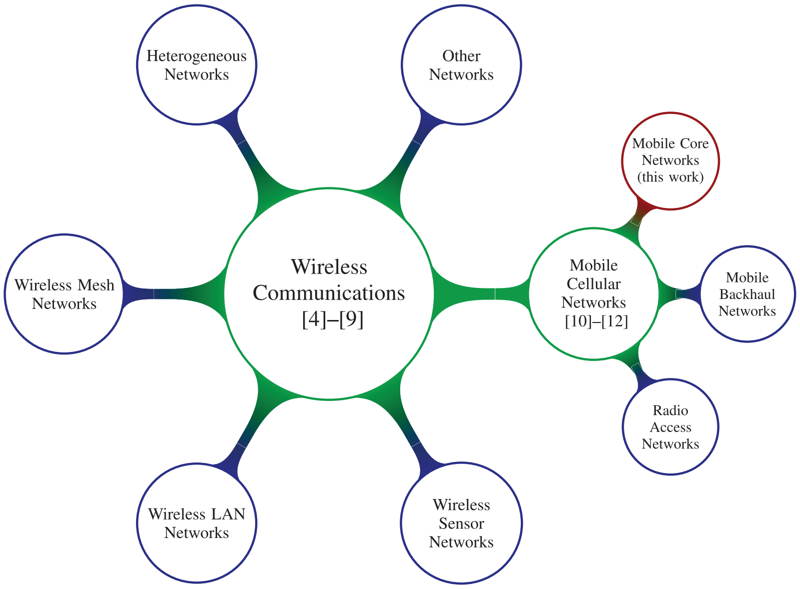

Fig. 1. Scope of the survey paper in red circle within the scope of wireless communication system and the comparison to other survey papers.

TABLE I
A SUMMARY OF RELATED SURVEY PAPERS

Survey Papers	Year	Covered Scope	Technologies	Comparing current research works in terms of	# papers in MPC
Qadir et al. [4]	2014	WLAN, WSN, MCN	SDN	Summarizing main idea	2
Tomovic et al. [10]	2014	MCN	SDN	Requirements, key benefits	2
Yang et al. [5]	2015	WLAN, MCN	SDN	Year, network type, network position	4
Jagadeesan et al. [6]	2015	WLAN, WSN, Mesh Networks, MCN	SDN	OpenFlow compatibility, research thrust	2
Chen et al. [11]	2015	MCN	SDN	No comparison table	5
Akyildiz et al. [7]	2015	WLAN, MCN	SDN, NFV	SDN architecture, scalability, network virtualization, software-defined traffic engineering, research community	5
Bizanis et al. [8]	2016	WSN, MCN	SDN, NFV	Year, technology, summarizing main idea	4
Haque et al. [9]	2016	WSN, Mesh Networks MCN, Home Networks	SDN	Features, data plane, virtualization, OpenFlow compatibility	7
Nguyen et al. [12]	2016	MCN	SDN, NFV	Architecture type, user traffic routing, compatibility, virtualization model, main components, southbound interface	30
This work	2016	MPC	SDN, NFV	Architectural model, technology adoption, functional implementation, deployment strategy, user traffic routing, southbound interface, compatibility, network slicing, scalability	60+

WLAN = Wireless Local Area Network; WSN = Wireless Sensor Network; MCN = Mobile Cellular Network; MPC = Mobile Packet Core Network

cover the whole or multiple components of the wireless communications system ranging from Wireless Sensor Networks (WSN), wireless local area networks (WLAN) such as Wi-Fi, wireless mesh networks, heterogeneous networks, and MCNs such as 3G, 4G. Qadir *et al.* [4] presented an architectural survey on programmable wireless networks which covers proposals applying SDN and virtualization into different kind of wireless networks including WLAN, WSN, MCN, and cognitive wireless networks. Yang *et al.* [5] mainly surveyed current efforts applying SDN and virtualization into MCN and WLAN networks. Jagadeesan and Krishnamachari [6] focused on surveying the adoption of SDN into WLAN, mesh networks, WSNs, and MCNs. Akyildiz *et al.* [7] provided an overview and a qualitative evaluation of several research works on SDN and NFV for 5G systems, where most of examined works are about SDN based radio access and WLAN networks. Haque and Abu-Ghazaleh [9] evaluated the use of SDN in MCN, WSN, mesh and wireless home networks. Bizanis and Kuipers [8] presented a survey about SDN and virtualization for Internet of Things (IoT), which covers proposals adopting these two concepts into MCN and WSNs. Also from this figure, Tomovic *et al.* [10], Chen *et al.* [11], and Nguyen *et al.* [12] are the related surveys, which only focus on the MCN architecture, which is typically composed of Radio Access Networks (RANs), mobile backhaul networks, and the MPC network. Tomovic *et al.* [10] and Chen *et al.* [11] summarize research works on leveraging SDN into both RANs and the MPC network parts of the MCN. The work closest to this article is proposed by Nguyen *et al.* [12]. There, the authors surveyed the use of SDN and NFV in the RAN, mobile backhaul, and MPC network parts of LTE MCN architecture by providing a hierarchical taxonomy corresponding to these parts. In the MPC part, the authors simply classified the surveyed works based on architectural approaches either revolutionary or evolutionary and did not answer important questions such as how network functions in those architectures are implemented and deployed. Therefore, they failed to classify them in a fully comprehensive way.

As summarized in Table I, all existing surveys have different scopes, in general wider than our work. Although there are some survey papers which also covered the MPC network part, the number of surveyed works is very limited. This table also points out how the authors compared the surveyed works in their papers by naming the criteria that they used in comparison tables. In contrast, in this paper we focus on surveying and analyzing the research work in the area of MPC network with the assist of a proposed four-dimensional taxonomy, which helps to classify the current research proposals by approaching them from different points of view, and to compare them by using a large number of criteria that has not been covered in the previous works. To the best of our knowledge, this is the most comprehensive and intensive survey on SDN and NFV in the MPC network architecture.

The main contributions of this paper include:

- A description of the most typical ways to re-architect the current MPC network architecture by adopting SDN and NFV technologies, and a discussion of their advantages and disadvantages;
- A definition and presentation of a four-dimensional taxonomy showing the trade-offs between different evolution approaches, different technology adoptions, different implementation options and different deployment strategies;
- An elaborate classification of the most relevant and up-to-date research proposals supporting the adoption and the advancement of SDN and NFV into the MPC network, and an exhaustive comparison of these proposals according to the proposed taxonomy plus other added attributes and criteria;
- An identification of new research challenges and issues raised from the taxonomy and comparison tables, and a discussion of potential research directions that might be conducted in the future to achieve a complete solution on the SDN and NFV based MPC network architecture.

B. Organization

The rest of this paper is organized as follows. Section II presents an overview of the current LTE/EPC system with focus on the MPC part and identifies the major problems faced by mobile network operators. A review of basic concepts of SDN and NFV, and their impacts on the development of the current MPC network architecture is also provided in this section. In Section III, we describe and analyze the four most common migration directions that the mobile network operators are currently following to re-architect their MPC networks. The definition of our high-level four-dimensional taxonomy is detailed in Section IV. Sections V–VIII describe current research initiatives organized according to four proposed dimensions: architectural approach, technology adoption, functional implementation, and deployment strategy, respectively. Also, these sections provide a comprehensive and exhaustive comparison of all initiatives are listed and analyzed. Based on the comparison of the state-of-the-art in the previous sections, we identify key open research problems for further investigation and exploration in Section IX. Finally, we conclude our survey in Section X.

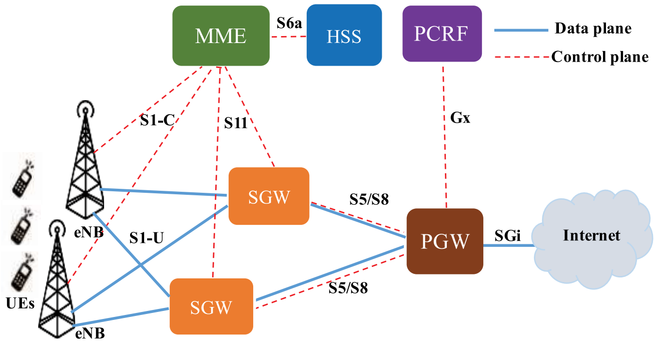

Fig. 2. The current EPS system [3].

II. Background

The first part of this section summarizes the background of the current mobile packet core architecture and its major problems. The second part provides an introduction of SDN and NFV, and their benefits in EPC, how they can be considered as enablers of future mobile packet core networks.

A. Evolved Packet Core Architecture

The most recent MPC network is the EPC, the core of the LTE system [29]. The EPC architecture first appeared in 3GPP Release 8 of the standard and has been widely deployed all over the world. The EPC together with LTE forms the Evolved Packet System (EPS) which is a flat, all-IP network, and dedicated to support only packet-switched connectivity. The current EPS system is depicted in Fig. 2. The LTE or the E-UTRAN (Evolved Universal Terrestrial Radio Access Network) is the access part of the EPS, which includes eNodeBs. The EPC architecture has five main functional entities: the Mobility Management Entity (MME), the Serving Gateway (SGW), the Packet Data Network Gateway (PGW), the Home Subscriber Server (HSS) and the Policy control and Charging Rules Function (PCRF). The MME serves as the key control entity in the EPS system and is responsible for handling all signaling events including mobility management, paging, as well as managing bearers setup, subscriber information, etc. The SGW is responsible for forwarding and routing user-data packets between eNodeBs and the PGW, and it acts as the local mobility anchor point for the inter-eNodeB handover. The PGW is a termination point for the user data packets from the mobile network towards external networks

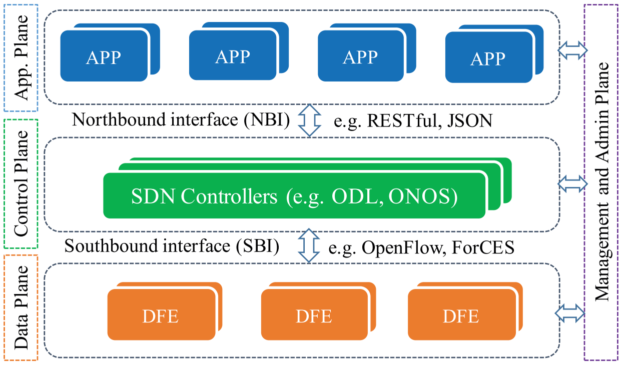
App. Plane
APP
APP
APP
APP
Northbound interface (NBI)
e.g. RESTful, JSON
Control Plane
SDN Controllers (e.g. ODL, ONOS)
Southbound interface (SBI)
e.g. OpenFlow, ForCES
Data Plane
DFE
DFE
DFE
Management and Admin Plane

Fig. 3. A high-level overview of SDN architecture [31].

and vice versa. Its main functions include device IP address allocation, policy enforcement, packet filtering and charging, etc. The HSS is responsible for subscription management. The PCRF provides QoS profiles and charging rules to the PGW. The number of interfaces and protocols are also specified in order to provide an end-to-end connectivity between UEs and the external networks with different QoS levels. The general packet radio service (GPRS) tunneling protocol (GTP) is used by the control plane to setup tunnels in the data plane[2] to forward the user data packets from the eNodeB to the PGW and vice versa. Interested readers can seek more detailed information about the LTE technology and the EPS architecture from [29] and [30].

Although having been deployed worldwide over the last few years, the EPC still has many issues and problems: the high deployment and dimensioning costs due to dedicated, vendor locking hardware implementation; the poor scalability and low flexibility due to incomplete separation between the control and data planes; the inefficient resource provisioning and allocation due to manual and static network configurations; the suboptimal data packet forwarding and routing due to a hierarchical architecture design. These problems impact the mobile network operators' revenue and slow down their time-to-market for new innovations. In order to address the described problems, it is necessary to have a radical change in the architecture design.

B. *Software Defined Networking*

Software Defined Networking (SDN) [13] is essentially a centralized networking paradigm, in which the network intelligence (i.e., the control function or the control plane) is logically centralized at one or a set of control entities (i.e., SDN controllers) while the data forwarding plane is simplified and abstracted for applications and networks services requesting through the SDN controllers. In the first generation of its development, SDN is based on OpenFlow [32], which is the most widely-used protocol between the control and data planes, and currently being maintained by ONF [24]. Since then, many other protocols such as ForCES [33] have been integrated into the SDN architecture. The high-level overview of the SDN architecture with four planes is shown in Fig. 3. The application plane consisting of applications, such as routing, and load balancing, communicates with the SDN controller in the control plane through northbound interfaces (e.g., REST and JSON). The control plane consists of one or a set of SDN controllers (e.g., ONOS [34], OpenDayLight [35]), which logically maintain a global and dynamic network view, provide control tasks to manage the network devices in the data plane via southbound interfaces (e.g., OpenFlow [32], ForCES [33]) based on requests from the applications. The controllers communicate with each other using east-westbound interfaces. The data plane is composed of Data Forwarding Elements (DFEs) such as virtual/physical switches and routers, which forward and route the data packets based on rules installed by the SDN controllers. The management and administration plane is recently considered by ONF [31] and IETF [36]. This plane is responsible for all activities related to provisioning and monitoring of the networks. Interested readers can find more about SDN and its applications in [37]–[41].

Nowadays, SDN has gained a lot of attention from both academia and industry in many networking areas, not only wired networks such as campus or data center [37], [38] but has also been expanding quickly in the field of mobile and wireless networks [12]. While talking about the core of the LTE mobile network, EPC, the SDN concept is initially applied to achieve a clear separation between the control (C) and user planes (U) in SGW and PGW entities. By splitting the gateways in this manner (i.e., from SGW to SGW-C and SGW-U and from PGW to PGW-C and PGW-U), it is possible to scale these components independently and it also enables a range of deployment options. The protocol used between the control and user plane can be either an extension of the existing OpenFlow protocol, which is being developed by the ONF Wireless and Mobile Working Group (WMWG) [42] or new interfaces, namely Sxa and Sxb, which are being defined and specified in 3GPP CUPS working item [27]. Another option for evolution, where the entire EPC is completely substituted by the SDN components, is being actively discussed.

C. *Network Function Virtualization*

Network Function Virtualization (NFV) [14], [43] is essentially the relocation of network functions from standalone boxes based on dedicated hardware to software appliances running in the cloud environment or on general-purpose commodity servers. By using NFV, each conventional network function (NF) is now running on a virtual machine (VM) as a 1:1 mapping model or is decomposed into smaller components called Virtual Network Function Component (VNFC) running on multiple VMs as a 1:N mapping model [44]. The NFV architectural framework is shown in Fig. 4. In this figure, Virtual Network Functions (VNF), which represents the implementation of NFs, are deployed and executed on a NFV Infrastructure (NFVI). The NFVI consists of virtual resources, which are abstracted and logically partitioned from underlying hardware resources (computing, storage, and networking) through a virtualization layer. The NFV management and orchestrator (NFV MANO) [45] is

[2] In this paper, we use the terms data plane and user plane interchangeably.

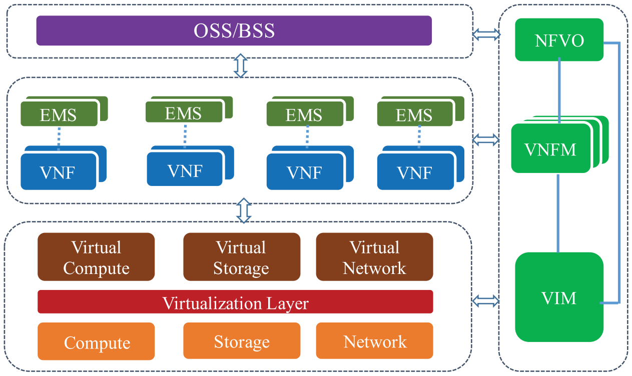

Fig. 4. NFV architectural framework [14].

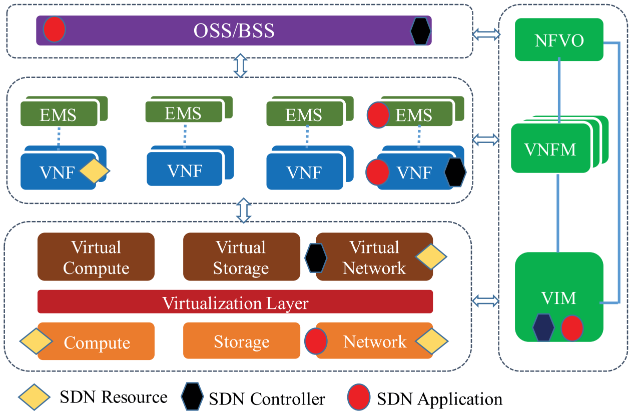

Fig. 5. A mapping SDN elements to NFV architectural framework [47].

responsible for orchestrating and managing VNFs (through a set of VNF Managers [VNFMs]) and NFVI (through a Virtualized Infrastructure Manager [VIM]). The NFV orchestrator (NFVO) is in charge of network services (NS) lifecyle management, on-boarding of new NS, etc. In addition, the NFV MANO also allows the integration with external Operational and Business Support Systems (OSS/BSS).

Among a variety of NFV use cases covered in [46], virtualizing the functionalities within the EPC is one of the most important use cases and has been attracting a lot of attention, especially from mobile network operators (MNO). MNOs have seen the potential benefits brought by NFV including reduction of their CAPEX and OPEX costs, better flexibility of management, dynamic scaling of resources, services agility, which hence increase their revenue. As the first option, all five main EPC entities including MME, HSS, SGW, PGW and PCRF are virtualized as VNFs, and deployed in their cloud data center. Each type of VNF then could form a pool (e.g., a MME pool and a HSS pool) and get scaled independently according to their specific resource requirements. Another option of EPC virtualization is to virtualize only part of EPC (e.g., MME and HSS) while keeping the rest (i.e., SGW and PGW) as physical appliances due to performance issues.

D. SDN Versus NFV

Being born at different times and promoted by different communities and organizations, SDN and NFV share many properties and are highly complementary to each other. They both aim to accelerate the innovation of new services towards a software-driven networked ecosystem [47], [48]. More specifically, NFV can serve SDN by virtualizing SDN elements such as the SDN controller, SDN data forwarding entities (which can be seen as network functions) to run in the cloud, thus allows the dynamic migration of these components to their optimal locations. In turn, SDN serves NFV by providing programmable network connectivity between VNFs to achieve optimized traffic engineering and steering [49]. Although SDN and NFV are mutually beneficial to each other, the frameworks are not dependent on each other. It means that the network functions can be virtualized and deployed without SDN and vice-versa. Figure 5 gives an example of mapping SDN elements to the NFV architectural framework. In this figure, SDN elements (i.e., SDN resource, SDN controller, and SDN application) can be positioned in different places in the NFV framework. For example, an SDN application can be implemented as a VNF, or can be part of a physical network function, or can be part of an Element Management System (EMS), etc.

The combination of SDN and NFV enables dynamic, flexible deployment and on-demand scaling of network functions, which are necessary for the development of the future mobile packet core towards a 5G system. Such characteristics have also encouraged the development of network slicing and service function chaining. From a UE perspective, slicing a network is to group devices with similar performance requirements (transmission rate, delay, throughput, etc.) into a “slice”. From network perspective, slicing a network is to divide an underlying physical network infrastructure into a set of logically isolated virtual networks. This concept is considered as an important feature of a 5G network, and also being standardized by 3GPP [26]. Service Function Chaining (SFC) [50] or network service chaining allows traffic flows to be routed through an ordered list of network functions (firewall, load balancers, etc.). The best practical use case of SFC is to chain network functions (i.e., middleboxes in this case) placed in the interface between PGW and the external networks [51], [52].

III. SDN/NFV-Based MPC Architectures

In this section, we aim at describing major migration directions of the MPC network architecture, and how SDN and NFV are part of these directions. We take the EPC architecture, which is the most recent MPC network architecture, as the starting point. It should be noted that in reality, deployed EPC entities are interconnected by means of an intermediate transportation network. However, in the scope of this survey, we focus on the change of the EPC and its entities with SDN and NFV technologies. Therefore, the EPC architecture and its evolutions are represented as logical architectures by omitting the transportation network.

Based on our observation from a collection of research proposals and industry talks, we have seen three major ways to re-architect the EPC architecture: (1) virtualizing EPC with

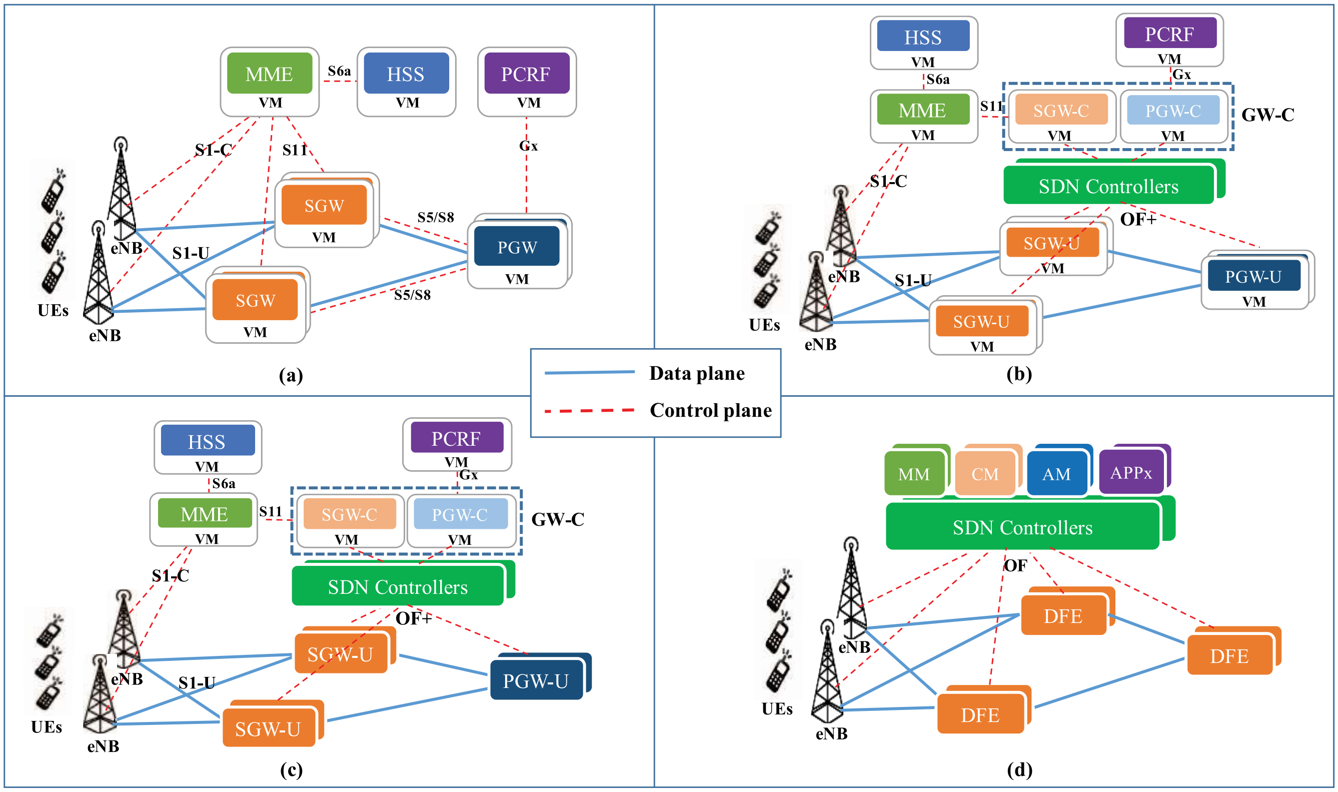

Fig. 6. Typical ways of re-architecturing the MPC architecture with SDN and NFV: a) Fully NFV-based EPC architecture, b) SDN/NFV-based EPC architecture with virtualized data/user plane, c) SDN/NFV-based EPC architecture with non-virtualized data/user plane, d) Fully SDN-based MPC architecture.

NFV technology (NFV-based EPC or vEPC), (2) decoupling control and user planes in vEPC with SDN technology, and (3) fully SDN realized MPC. These three ways in turn may form a three-step evolution roadmap or evolution path of the current MPC (i.e., EPC). However, it is not mandatory to follow that evolution path since, for example, one can start designing a full-SDN MPC architecture as in (3) without considering the existing EPC architecture as in (1) and (2). At the end of each subsection, we analyze and discuss advantages and disadvantages of each evolution direction.

A. NFV-Based EPC Architecture

The first design of an EPC architecture which is purely based on the NFV concept is illustrated in Fig. 6 a). In this figure, all conventional EPC entities are migrated from dedicated hardware platforms and implemented as software appliances running on Virtual Machines (VMs) or containers on a cloud system (e.g., OpenStack [53]) without any functional modifications. The interfaces (e.g., S11, S6a, etc.) and protocols (e.g., GTP and DIAMETER) used to communicate between those entities are still the ones standardized by 3GPP. These VMs or containers are instantiated and managed by the cloud controller or using some recent MANO tools such as OpenStack Tacker [54] or OpenBaton [55]. The resource (i.e., computing, storage, networking) for the VNFs are provided by NFVI (see Fig. 4). Although Fig. 6 a) shows a full virtualized scenario, a scenario in which only control plane entities (i.e., MME and HSS) are virtualized, while user plane entities (i.e., SGW and PGW) are non-virtualized has also been considered due to some strict requirements on data processing.

Apart from anticipated benefits such as cost reduction, and flexibility brought by NFV, i.e., benefits previously mentioned, this type of EPC evolution seems to be the most practically feasible approach to realize in the current EPC as it requires no major changes to the current EPC deployment. In addition, it should be noted that each conventional entity can be virtualized as multiple VMs, thus it brings up the usage of the multi-tenancy concept, which opens up for the deployment of multiple NFV-based EPCs (vEPC) simultaneously. Moreover, since standard interfaces and protocols are maintained, it allows mobile operators to easily interwork vEPCs with their existing EPC. However, this approach still has several limitations and drawbacks. Keeping all VNFs tightly within 3GPP standards imposes challenges in the management and the orchestration process when adding new VNFs (i.e., scaling out) because these VNFs are required to be configured and instantiated in a coherent way. In addition, the scaling and provisioning process are still inefficient due to the tight coupling between the control plane and the user plane at gateways, which have different resource requirements (i.e., control plane requires low latency, while the user plane requires high throughput). Moreover, User Equipment (UE) contexts and information which are currently kept inside the EPC entities are now kept inside the EPC VNFs. This information can be affected or even lost during the scaling procedures, in particular removing a VNF from the system. Therefore, it results in reliability and resiliency problems which will be further discussed in Section IX.

Currently, the vEPC has been commercially offered by many leading mobile operators, service providers, and equipment vendors. For example, NEC Corporation launched the world's first vEPC solution in 2013 [56]; Ericsson's complete vEPC solution was demonstrated in 2014 [57]; Cisco offers a solution called Virtual Packet Core (VPC), which covers all packet core services [58]; Nokia and Alcatel-Lucent offers its vEPC application software in [59]; and NTT DoCoMo has recently completed development of the world's first development of multi-vendor EPC software [60], etc.

B. SDN/NFV-Based EPC Architecture

This subsection presents the second design of EPC architecture which is based on SDN and NFV concepts. In this approach, the EPC gateways (SGW, PGW) are first partitioned into the control and user planes. Then, the control functions (SGW-C, PGW-C) are virtualized as VNFs like other virtualized control entities, while the user plane functions (SGW-U, PGW-U) are either virtualized or non-virtualized as illustrated in Fig. 6 b) and 6 c), respectively. It should also be noted that it is possible to combine SGW and PGW as a single entity where SGW-C and PGW-C can be merged into an unified control entity, a so-called GW-C, while SGW-U and PGW-U can be merged into an unified user-plane entity, a so-called GW-U. In all cases, an SDN controller is introduced to bridge between the control and user planes. This SDN controller, which can be either virtualized or non-virtualized, is in charge of interpreting the signaling messages received from the control plane and responsible for installing the forwarding rules (e.g., GTP tunnel establishment) into the user plane via an open API. The open API in this case could be an extension of the OpenFlow protocol which is identified as OF+ in Fig. 6 b) and 6 c). These extensions typically are GTP matching fields and an action set which tells the user plane how to handle (e.g., encapsulate/decapsulate) with packets which have headers. The user plane (SGW-U, PGW-U) can be implemented as OpenFlow switches capable of GTP encapsulation and decapsulation. Nevertheless, the extensions are still under discussion and have not been standardized yet.

Compared to the vEPC architecture described in the previous section, this approach not only has advantages such as flexibility, and backward compatibility but also overcomes drawbacks of the vEPC architecture with the introduction of SDN. Indeed, the control and user planes of EPC are now completely separated, thus they can get scaled independently in a cost-effective manner. In addition, SDN brings flexibility of flow distribution over the infrastructure, and thus provides better UE mobility management. Moreover, being decoupled from each other, control and user plane functions can be flexibly placed around the network, for example, closer to the edge or users, thus shortening the network latency. This encourages the development of mobile edge computing [61] and its use cases including traffic offloading or local breakout, distributed content and service caching, augmented reality, etc. However, introducing a new SDN controller and its interfaces to communicate with the control and user planes exposes more latency in the network. In addition, keeping the use of GTP tunneling protocol is also another factor contributing to signaling latency and packet header overhead. Last but not least, the scalability of the SDN controller is also a major problem. It could be overcome by using multiple controllers or through a hierarchical design of controllers [62], [63], but in the context of mobile network, these designs are still unclear.

This type of development was first considered by Ericsson as the concept of implementing EPC in a cloud computer with an OpenFlow data plane described in Kempf *et al.* [64], which was then patented in [65]. Since then, many other efforts from mobile operators, service providers, and equipment vendors have been done through some Proof of Concepts (PoCs). Nokia Networks has demonstrated a scenario showing the feasibility of an SDN and virtualized based EPC solution in case of large crowd events (e.g., a football match or a music festival) [66]. Many other PoCs have been done by collaborations between companies and under the sponsorship of ETSI NFV ISG group in [67]. For example, Telenor, Vodafone, Hewlett Packard Enterprise (HPE), and Redhat have collaborated to demonstrate the capabilities of an SDN-enabled virtual EPC architecture in [68]. This type of design is currently being commercialized by HPE as an SDN-enabled MPC platform [69]. An alternative design where the control plane functions and data plane functions of gateways are converged into unified GW-C function and GW-U functions, respectively, is envisioned by ZTE Corporation in [70] or to be released by SK Telecom (SKT) in [71].

C. Full-SDN MPC Architecture

This subsection presents the third design proposal of an EPC architecture, which is purely based on the SDN concept. The architecture is illustrated in Fig. 6 d). In this architecture, all conventional EPC entities no longer exist or are collapsed. Instead, the user plane entities are replaced by data forwarding entities (DFEs) (e.g., switches and middleboxes), while control plane entities are replaced by a set of software applications implemented on top of an SDN controller. These applications could be newly defined or simply decomposed from functionalities of conventional EPC entities. For example, the MME and the SGW are traditionally sharing similar functionalities such as connectivity management, mobility management, while the MME and the HSS are sharing similar functionalities like authentication, attachment management. These functionalities can be formed or merged together as unified control elements or modules such as connectivity management (CM), mobility management (MM), and authentication management (AM), etc. (as depicted in Fig. 6). In this architecture, the GTP tunneling protocol is eliminated, instead, the user data packet is routed on the basis of flow entries in the DFE which are configured from the SDN controller via OpenFlow or other southbound interfaces. Although this approach is seen as a full realization of the SDN concept, it should be noted that the network functions such as CM, MM and the SDN controller can also be virtualized as VNFs.

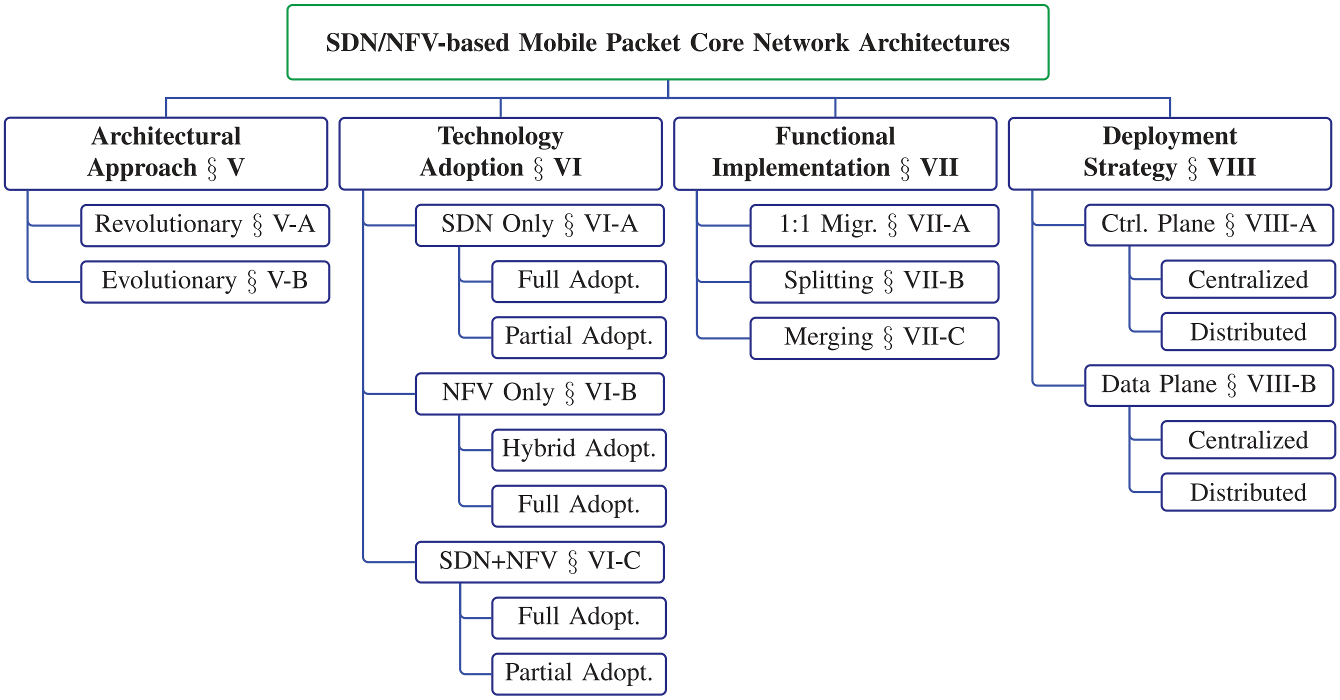

Fig. 7. Classification taxonomy tree with four dimensions: architectural approach, technology adoption, functional implementation, and deployment strategy.

The biggest advantage of this revolution approach is the capability of a fully programmable and flat network architecture. In addition, eliminating the use of GTP tunneling helps overcome the drawbacks discussed in the two previous approaches. More importantly, fully leveraging SDN can empower the network slicing technology which is one of the key features of 5G networks. However, it is worth noting that this option, while achieving the highest flexibility and programmability, presents functional implementation problems, and the complexity of the control plane due to the porting of a large number of atomic network element. In addition, since the PCRF entity is collapsed, it results in challenges while enforcing QoS policies in such new architecture. Moreover, this approach is not compatible at all with the existing MPC since it eliminates the usage of standard interfaces and protocols. Finally, similar to the previous approach in Section III-B, the scalable design of the SDN controller and control plane still needs to be further investigated.

While talking about industry-related activities, this “clean-slate” approach has appeared in scientific research papers from Huawei Telecom research center such as Guerzoni *et al.* [72] and Trivisonno *et al.* [73], [74]. Currently, this type of design has been presented as one of the key design principles for the development of an on-going 5G project named 5G CONFIG [75] led by Huawei Telecom in a consortium of several network operators (e.g., Telenor, Orange Telecom), vendors (e.g., NEC, Thales), etc. The feasibility of this approach has also been demonstrated in [76].

IV. Taxonomy Description

In the previous section, we have discussed the major directions of re-architecting the MPC network architecture by using SDN and NFV technologies. The main objective is to provide our readers a brief tutorial on the topic before getting involved in reviewing a collection of ways of leveraging SDN and NFV into the MPC. More importantly, it serves as the base for our taxonomy tree, which helps us classify research work into categories in a certain way, thus providing the readers a complete and comparative view of all current research proposals. For example, the way of using SDN and NFV in the three described architectures inspired us to classify the research work based on the choice of technology, and the way of constructing network functions in the full-SDN based MPC architecture inspired us to come up with the classification based on different options to implement the network function.

As presented in Fig. 7, our taxonomy for analyzing SDN/NFV-based MPC related research is constructed in four main dimensions including (1) *architectural approach*, (2) *technology adoption*, (3) *functional implementation*, and (4) *deployment strategy*. These main dimensions will be further categorized into different smaller groups which will be elaborated in detail later in this section. Although there is an overlap between some of the dimensions, for example, between the architectural approach and the technology adoption, our purpose is to allow the readers to approach the research topic from different angles. They can then have an observation of the trade-offs between choices for which in terms of technologies are selected, architectural and functional design as well as deployment options. Moreover, this observation can provide some insights to mobile operators so that they can make a decision on what the most suitable implementation and deployment options are. The definition and description of each dimension and their subcategories are elaborated as follows.

A. Architectural Approach: Revolutionary vs Evolutionary

The purpose of the first dimension is to classify the research work into different categories defined upon the approach taken to construct the architecture. Two major architectural approaches can be adopted to re-architect the current MPC network with SDN and NFV technologies: revolutionary and evolutionary.

A research work is classified as revolutionary or "clean-slate" if it entails the complete replacement of the entire legacy MPC with SDN and NFV. It means that all legacy MPC entities and interfaces between them no longer exist. Fig 6 d) is an illustration of a revolutionary or "clean-slate" architecture. Replacing the entire architecture including modification of functionalities and protocols, this architecture is backwards incompatible to the legacy MPC.

In contrast, we classify a research work as evolutionary when the architecture presented in that work is an incremental deployment of SDN and NFV into the existing MPC. It means that the legacy MPC entities can be either virtualized or "software-defined" but all or a subset of internal functionalities and existing interfaces still exist. Fig 6 a), b) and c) in Section III are examples of evolutionary architectures. In these figures, the interfaces and protocols between MME and other entities (either virtualized like HSS or "software-defined" like SGW-C) are still kept as standard ones (e.g., Diameter over S6a and GTP over S11). This characteristic allows to keep the use of the traditional GTP tunnel-based mechanism in the data plane to route and forward the user data packets even if the control and data planes have been separated. In addition, this characteristic would create more chances to interoperate with the legacy MPC architecture (i.e., backward compatibility).

B. Technology Adoption: SDN vs NFV

The second dimension is to classify the research work into different categories defined on the basis of the technology used. A research work is classified as SDN-Only when they utilize only SDN, i.e., does not use NFV. Fig 6 d) could be an example of this category. In contrast, a research work is classified as NFV-Only when they utilize only NFV (not SDN), as shown in Fig 6 a). Complementary to these two, a research work is classified as SDN and NFV when they employ both two concepts in their proposals, as shown in Fig 6 b) and 6 c).

Furthermore, each category is further divided into two subgroups. For the SDN-Only category, we classify a research work that makes use of SDN into a part of the MPC architecture only (e.g., SDN-based gateways or user plane) as partial adoption while a research work in which SDN is fully adopted is classified as full adoption. For the NFV-Only category, a research work is classified as hybrid if the virtualized architecture and legacy architecture exist simultaneously, while a research work which virtualizes the entire MPC architecture is classified as full. For the SDN and NFV category, a research work is classified as partial if NFV is used to virtualize only a part of the MPC architecture (e.g., the control plane), whereas, a research work that virtualizes the entire MPC architecture is classified as full.

C. Functional Implementation: Splitting vs Merging

The third dimension focuses on the implementation options of network functions proposed by the research to be classified. From a software development perspective, a network function could either be implemented as a set of modules decomposed from the original function (which we call the s̈plitting model) or as a single, one-size-fits-all, multi-purposed entity (which we call the "merging" model).

The splitting model, which also refers to the modularization or decomposition of network functions, is considered as one of the key principles in the design of 5G core network architecture [75]. The functionality of control plane entity (MME as an example) is modularized as single-purpose elements such as connectivity management, mobility management, authentication management, etc. These modularized elements could be implemented, as applications (APPs) on top of the SDN controller as shown in Fig. 6 d). The data plane entity (e.g., PGW) is decomposed into a chain of simplified network functions (e.g., forwarding, charging). As such, the "splitting" principle enables great flexibility and dynamic of network function deployment according to different service requirements and more importantly encourages the use of network slicing [22], [23], [26]. It should be noted that the splitting of the control and data planes at EPC gateways (i.e., SGW, PGW) is not viewed as the "splitting" implementation model in our taxonomy.

In contrast, the "merging" model, also referred to as grouping of network functions, is another implementation option to overcome the drawbacks of the "splitting" model such as the design complexity, and extra latency exposed between atomic elements. From the SDN perspective, this implementation model presumes that all the control plane entities (i.e., MME, HSS, S/PGW-C) or data plane entities (i.e., SGW/PGW-U) are merged into a single or some multi-purposed control and data plane components, respectively. From the NFV perspective, VNF entities (e.g., MME, HSS) could be divided into groups based on their interactions and workload. As such, the "merging" approach would improve performance in terms of signaling latency by allowing the entities communicate internally, and it could also reduce the complexity of network design as well as simplify many operational tasks. However, it results in the low scalability and inflexible network management. Tying back to the evolution directions described in Section III, the full-SDN MPC architecture can employ both the "splitting" and "merging" models as APPs can be either modularized or multi-purposed.

Last but not least, the simplest implementation option is "1:1 migration". From the NFV perspective, each network function is implemented as an individual VM, without modifying internal functionalities or interfaces, and is not grouped together with others. For example, SGW and PGW are implemented as VMs without separating the control and user planes. From the SDN perspective, the "1:1 migration" means that a network function is still kept unmodified or is implemented as a corresponding application on top of the SDN controller. For example, the PGW is reused without being virtualized and its control and user planes are not decoupled.

This approach brings the most simplicity, but still results in scalability problems.

Based on the definition above, we classify a research work into "1:1 migration", "splitting", and "merging" subgroups if a network function presented in that research follows the "1:1 migration", "splitting", and "merging" implementation models, respectively.

D. Deployment Strategy: Centralized vs Distributed

The fourth dimension covers the deployment strategy that is presented in the research work to be classified. The deployment strategy or function placement strongly depends upon the operator's requirements. The operators decide how to place their MPC entities based on what kinds of services they are providing. There are several ways to place the network functions, but they all converge into two main streams: centralized placement and distributed placement. Centralizing all the network functions allow operators to manage and monitor their network easily, but introduces high end-to-end latency (either control or data planes), which is not suitable to serve new services which require ultra-low latency such as autonomous driving, smart-grid or automated factory [2]. By deploying the network function close to the edge or users, especially the data plane, could eliminate network delay and could enable traffic optimization, thus improving user quality of experience. In addition, the distributed approach help to promote the development and the use of mobile edge computing and fog computing in the mobile network, which will significantly close the gap between the existing 4G systems (i.e., LTE/EPC) and services [20]. However, this approach introduces the difficulty of management and orchestration of the network.

As presented in [77] and [78], potential deployment strategies for EPC could be (1) centralizing the data plane while distributing the control plane, (2) centralizing the control plane while distributing the data plane, (3) completely centralizing both the control and data planes, and (4) completely distributing both the control and data planes. These four deployment strategies are inherently used as subcategories under the deployment strategy dimension in our survey taxonomy. From the SDN perspective, the control plane can be fully centralized or decentralized while the data plane is often deployed in a distributed manner as a collection of distributed SDN switches. From the NFV perspective, the control and data planes can be (i) virtualized and distributed together at distributed data centers or (ii) only the data plane is distributed while the control plane remains centrally or (iii) centralizing both the control and data planes.

We consider a research work to have a centralized control plane if all network control functions presented in that work are fully deployed in a centralized manner (e.g., at a centralized data center or a centralized SDN controller). The research work is classified as distributed control plane group if in that research work, either a part of control plane is offloaded to the edge of the network (e.g., to an access controller) or the control plane is hierarchically constructed. For the data-plane deployment, we classify all research work that purely employs SDN as belonging to the distributed data plane group. The reason is that, by being separated from the control plane, the data plane becomes a network of simple forwarding devices and can be deployed any where in a distributed fashion as long as they connect to the SDN controller. A research work in which the data plane functions are virtualized instances deployed in distributed data centers also belongs to this group. A research work is classified as centralized data plane group if the data plane functions presented in that work are either virtualized instances located at a central data center or conventional data plane functions (i.e., SGW/PGW) which are reused from the conventional EPC architecture.

E. Classification of Contemporary Work

From the following sections, we will present a survey on the most recent research initiatives on SDN and NFV-based MPC architectures. Since many proposals may belong to more than one category in the taxonomy, we only consider the most relevant papers from each category and then describe them in detail. However, we also briefly mention other work. For each research work in one category, we highlight the main contributions, and characteristics. Next, the research work is listed into two comparison tables. The purpose of the comparison is to give a summary of the differences between existing solutions on SDN/NFV-based MPC network architectures. In addition, it not only helps to observe the main points and strengths, but also the limitations and drawbacks of each proposed solution. Table II compares all research work in terms of architectural approach and technology adoption. Table III provides a comparison of all research work in terms of functional implementation and deployment strategy. In addition, in order to have a comprehensive comparison, we add some more attributes including southbound interfaces, compatibility, network slicing capability and scalability. These newly defined attributes will be described and explained in detail while describing the comparison tables.

V. Architectural Approaches

In the following, we will review the research work in terms of the architecture approach. As defined in Section IV-A, there are two types of architectural approaches: the revolutionary approach refers to the complete replacement of all conventional MPC entities and the standard interfaces and protocols used between them, while in the evolutionary approach, the whole entities or some parts of the existing MPC architecture, standard interfaces, and protocols still exist or remain as before.

A. Revolutionary Approaches

CellSDN [79] and its successor SoftCell [80] are the two earliest proposals aiming at completely re-designing the current MPC network architectures by incorporating SDN principles. As "clean-slate" designs, CellSDN and SoftCell simplify both control and data planes of the MPC network by using SDN components. Therefore, conventional MPC entities like MME, HSS, S/PGW are eliminated. Inspired from CellSDN and SoftCell, Moradi *et al.* [81], [82] proposed SoftMoW to address challenges imposed in a very large-scale

TABLE II
COMPARISON OF CURRENT RESEARCH INITIATIVES IN TERMS OF ARCHITECTURAL APPROACH AND TECHNOLOGY ADOPTION

References	Architectural Approach		Technology Adoption						User-Traffic Routing		Compa tibility	Southbound Interface
			SDN Only		NFV Only		SDN+NFV					
	Revol.	Evol.	Full	Partial	Full	Hybrid	Full	Partial	Non-GTP	GTP		
CellSDN [79], SoftCell [80]	✓	–	✓	–	–	–	–	–	✓	–	–	OF
SoftMow [81], [82]	✓	–	✓	–	–	–	–	–	✓	–	–	OF
Yazici et al. [83]	✓	–	✓	–	–	–	–	–	✓	–	–	OF
Lindholm et al. [84]	✓	–	✓	–	–	–	–	–	✓	–	–	–
Chourasia et al. [85]	✓	–	✓	–	–	–	–	–	✓	–	–	OF
Marquezan et al. [86]–[88]	✓	–	✓	–	–	–	–	–	✓	–	–	OF
SoftAir [7], [89]	✓	–	–	–	–	–	✓	–	✓	–	–	OF
Yang et al. [90]	✓	–	–	–	–	–	✓	–	✓	–	–	OF
SoftNet [91]	✓	–	–	–	–	–	✓	–	✓	–	–	–
CleanG [92]	✓	–	–	–	–	–	✓	–	✓	–	–	–
Einsiedler et al. [93]	✓	–	–	–	–	–	–	✓	✓	–	–	–
Roozbeh [94]	✓	–	–	–	–	–	–	✓	✓	–	–	SDN API
Trivisonno et al. [72]–[74]	✓	–	–	–	–	–	–	✓	✓	–	✓	–
Hampel et al. [95]	–	✓	–	✓	–	–	–	–	–	✓	✓	SDN API
Said et al. [96]	–	✓	–	✓	–	–	–	–	–	✓	✓	OF v1.3+
Sama et al. [97]	–	✓	–	✓	–	–	–	–	–	✓	✓	OF v1.3+
Nguyen et al. [98], [99]	–	✓	–	✓	–	–	–	–	–	✓	✓	OF v1.4+
Mahmoodi et al. [100]	–	✓	–	✓	–	–	–	–	–	✓	✓	OF+
Page et al. [101]	–	✓	–	✓	–	–	–	–	✓	–	✓	OF
Shanmugalingam et al. [102]	–	✓	–	✓	–	–	–	–	✓	–	–	OF
Mueller et al. [103]	–	✓	–	✓	–	–	–	–	–	✓	✓	OF v1.4+
Yongli et al. [104]	–	✓	–	✓	–	–	–	–	–	✓	✓	OF+
Osmani et al. [105]	–	✓	–	✓	–	–	–	–	–	✓	–	JSON-RPC
Jain et al. [106]	–	✓	–	✓	✓	–	–	–	–	✓	–	OF+
SoftEPC [107]	–	✓	–	–	✓	–	–	–	–	✓	✓	–
Taleb et al. [108]	–	✓	–	–	✓	–	–	–	–	✓	✓	–
KLEIN [109]	–	✓	–	–	✓	–	–	–	–	✓	✓	–
Baba et al. [110]	–	✓	–	–	✓	–	–	–	–	✓	✓	–
Hawilo et al. [111]	–	✓	–	–	✓	–	–	–	–	✓	✓	–
Kiess et al. [112]	–	✓	–	–	✓	–	–	–	–	✓	✓	–
Medhat et al. [113]	–	✓	–	–	✓	–	–	–	–	✓	✓	–
Jeon et al. [114]	–	✓	–	–	✓	✓	–	–	–	✓	✓	–
FME [115], [116]	–	✓	–	–	–	✓	–	–	–	✓	✓	–
Taleb et al. [117]	–	✓	–	–	–	✓	–	–	–	✓	✓	–
Ren et al. [118], [119]	–	✓	–	–	–	✓	–	–	–	✓	✓	–
MobileFlow [120]	–	✓	–	–	–	–	✓	–	–	✓	✓	Smf
Basta et al. [121]	–	✓	–	–	–	–	✓	–	–	✓	✓	SDN API
Hahn et al. [122]	–	✓	–	–	–	–	✓	–	–	✓	✓	OF+
Haleplidis et al. [123]	–	✓	–	–	–	–	✓	–	–	✓	✓	ForCES
i-Net [124]	–	✓	–	–	–	–	✓	–	–	✓	✓	–
An et al. [125]	–	✓	–	–	–	–	✓	–	–	✓	✓	OF v1.3+
Kempf et al. [64]	–	✓	–	–	–	–	–	✓	–	✓	✓	OF v1.2+
Hahn et al. [126]	–	✓	–	–	–	–	–	✓	–	✓	✓	OF+
Sama et al. [127]	–	✓	–	–	–	–	–	✓	–	✓	✓	OF-mpc
Kaippallimalil et al. [128]	–	✓	–	–	–	–	–	✓	–	✓	–	OF
Ameigeiras et al. [129]	–	✓	–	–	–	–	–	✓	✓	–	–	OF
Cattoni et al. [130]	–	✓	–	–	–	–	–	✓	✓	–	–	OF
MobiSDN [131]	–	✓	–	–	–	–	–	✓	✓	–	–	OF
Bradai et al. [132]	–	✓	–	–	–	–	–	✓	✓	–	–	OF
Costa-Requena et al. [133]	–	✓	–	–	–	–	–	✓	✓	–	–	OF
Hasegawa et al. [134]	–	✓	–	–	–	–	–	✓	–	✓	✓	OF+
Heinoen et al. [135]	–	✓	–	–	–	–	✓	✓	–	✓	✓	OF v1.3+
Basta et al. [136]	–	✓	–	–	–	–	✓	✓	–	✓	–	SDN API
Sama et al. [137]	–	✓	–	–	–	–	✓	✓	–	✓	✓	OF+
Taleb et al. [138]	–	✓	–	–	✓	–	✓	✓	–	✓	✓	OF+

"✓" indicates that the attributes are provided or applicable in the research work
"–" indicates that the attributes are unspecified or non applicable in the research work
Note: Some research works, which have more than one "✓" in one category meaning that they have more than one proposed solution

cellular mobile network, so-called mobile wide area networks. In this architecture, all conventional MPC entities also no longer exist.

Other proposals including Yazıcı *et al.* [83], Lindholm *et al.* [84], Chourasia and Sivalingam [85], Marquezan *et al.* [86], SoftAir [7], [89], SoftNet [91], Guerzoni *et al.* [72], Trivisonno *et al.* [73], [74], Yang *et al.* [90], Roozbeh [94], etc. are also following the same design principles. These are summarized in Table II. As observed from this table, the number of proposals in this category is small, less than one-fourth of the total number of proposals. The user data packets in these proposals are routed based on IP flow entries. Also from the table, we can see that, all revolutionary approaches take SDN as the key technology to reshape the current MPC architecture. Section VI will describe these work in detail to see how SDN changes the current MPC architecture.

B. Evolutionary Approaches

In research proposals such as Kempf *et al.* [64], Nguyen and Kim [98], [99], Hampel *et al.* [95], MobileFlow [120], Basta *et al.* [121], [136], are keeping all control plane entities unchanged. In some research proposals such as SoftEPC [107], KLEIN [109], Baba *et al.* [110], Hawilo *et al.* [111], Kiess *et al.* [112], Jeon *et al.* [114], and FME [115], [116], the control plane entities and user plane entities are kept unchanged, they are only migrated from dedicated hardware to commodity servers. Said *et al.* [96] and Sama *et al.* [97], the control plane entities and the PGW are unchanged. All proposals in this category are summarized in Table II. As observed from the table, the major of proposals adhere to the evolutionary approach. In addition, most of them keep using GTP tunneling protocol as a routing mechanism to route the user data packets. It can also be seen from the table that SDN and NFV and its combination play key roles in re-architecting the current MPC architecture in evolutionary approaches. Section VI will describe these works in detail to see how these technologies change the current MPC architecture.

VI. Technology Adoption

In this subsection, we will review the research work in terms of technology used in their proposed architectures. As mentioned in the taxonomy description in Section IV-B, there are three categories: adopting only SDN technology, adopting only NFV technology, and adopting both SDN and NFV technologies.

A. SDN Only

Adopting SDN technology into the current MPC architecture results in two approaches: Full SDN adoption and partial SDN adoption. In the following, we first describe the research work according to the full SDN adoption category and then the partial scheme.

1) Full SDN Adoption: As mentioned in Section V-A, CellSDN [79] and its successor SoftCell [80] use SDN as a key technology to simplify the MPC network architecture. In both architectures, the data plane is composed of commodity switches including access switches and core switches performing data packet forwarding between UEs and the Internet as shown in Fig. 8. In addition, a set of commodity middleboxes (e.g., transcoders and firewalls) is supported in order to handle complicated processing tasks relegated from the switches or to enforce QoS and service policies. In the control plane, a SDN controller consists of a network operating system running a collection of application modules such as mobility management, subscriber information base, and routing. The controller is in charge of computing paths and installing switch-level rules to direct the traffic through chains of switches and middleboxes based on high-level service policies. The most important contribution of SoftCell is the introduction of a scalable service routing mechanism by aggregating forwarding rules along multiple dimensions such as location-based aggregation, user mobility-aware aggregation. This proposed multi-dimensional aggregation takes advantages of traditional location-based routing and tag-based routing to scale to large networks with large service policies. The performance of SoftCell architecture was demonstrated through a prototype implementation and a large-scale simulation setup. Although, the evaluation results showed promising performance values of SoftCell in terms of number of service policies it can support, it has not been deployed or integrated into a real cellular network.

Considering the case that the MCN is organized into very large and rigid regional scale, e.g., a country, Moradi *et al.* [81], [82] proposed an architecture called SoftMoW, which basically applies the principles of SDN to re-design the architecture of such large-scale MCNs. Similar to CellSDN [79] and SoftCell [80], the data plane is also composed of programmable switches and a set of middleboxes. However, these components are distributed over a large geographical area. The control plane of SoftMoW also differs from the control plane of CellSDN [79] and SoftCell [80] in which it is hierarchically built up through recursive and reconfigurable abstraction mechanisms. SoftMoW's controllers are geographically distributed and logically organized in a multilevel tree structure and at each level, a controller is able to abstract the network topology it manages and then exposes it to the parent controller at upper level. The introduction of the concept of recursive constructions of the control plane distinguished the work and improves the flexibility as well as the scalability of the network. In order to solve the problem of having large numbers of policies and paths need to be enforced and computed, SoftMoW leverages a scalable recursive label swapping, which forwards the user data packets based on labels pushed from controllers, similar to SoftCell's design. With these design principles in mind, the authors developed a prototype as well as trace-based simulations to show the performance gains of SoftMoW compared with the current network in terms of inter-region handover optimization. Although SoftMoW gave promising performance figures, it is very hard to deploy this architecture in a real environment.

Another hierarchically constructed control plane of SDN-based MPC network architecture is proposed by Yazici *et al.* [83]. Similar to SoftMoW [81], [82], the control plane architecture is constructed in the way that the functions of lower-layer controllers are constrained by the upper-layer decisions. In addition, multiple control applications (e.g., failover, traffic optimization) for the same functionality can be realized at the same or different controller hierarchies. A newly defined device controller (also UE controller) distinguishes it to the previous architecture. This controller is able to communicate with a network controller in the MPC to offer an end-to-end connectivity management as a service (CMaaS). The authors illustrated the benefits of CMaaS through a use case of joining mobility management and routing management for device-to-device communications. However, this paper lacks of detailed design of the network controller compared to SoftMoW.

As an alternative approach, Lindholm *et al.* [84] envisioned an approach to re-factor the MPC architecture with the support of SDN. In their work, they first provided a state-space analysis of MPC network functions based on events generated by UE (e.g., attach, idle, wake-up, mobility). Based on the output of the state-space analysis, they constructed an SDN-based MPC architecture with a publish-subscribe control plane. It means that events generated from UE are subscribed and published within a controller in the MPC or between this controller and its agent in the RAN, according to the changes of UE state. The controller or mobile core controller contains functional modules corresponding to the UE events and programs the forwarding elements in the data plane. The shortcoming of this architecture is the possibility of signaling overload in the control plane due to the update of states during the publish-subscribe procedure. In addition, there is no performance evaluation in this work.

By using the same design principles of SDN concept as previously described works, Chourasia and Sivalingam [85] presented an OpenFlow-enabled EPC architecture with the goal of solving the signaling overhead problem of UE's handover. A detailed description of different procedures for both intra-LTE (between eNodeBs) and inter radio technologies (inter-RAT) handover of UE is provided and analyzed. The authors evaluated by using both analytical modeling and simulation methods and illustrated the performance gains of the proposed scheme over the traditional one in terms of significantly reduced signaling load. The shortcoming of this work is the scalability problem due to a single centralized EPC controller. Another study on understanding processing latency of SDN-based mobility management in MPC networks is presented in Marquezan *et al.* [87].

Most of presented works relies on the use of OpenFlow as a communication protocol between the control applications (APPs) and SDN switches. It means that in a normal OpenFlow protocol operation, whenever the switch receives unknown packets, it will always send a special OpenFlow message called PACKET_IN to the SDN controller to trigger the appropriate APPs on top of it. However, none of them takes in-depth consideration of how to use PACKET_IN in the context of mobile network since applications themselves also communicate to each other to exchange information (e.g., UE states). Marquezan *et al.* [86] explicitly address this problem by enabling PACKET_IN context interpretation at the SDN controller to determine exactly which APPs to invoke in order to process network events (e.g., attachment, mobility, etc.) sent from the SDN switches. The results from experimental evaluations showed the feasibility of the approach since the time for such dispatching process is only in the order of microseconds.

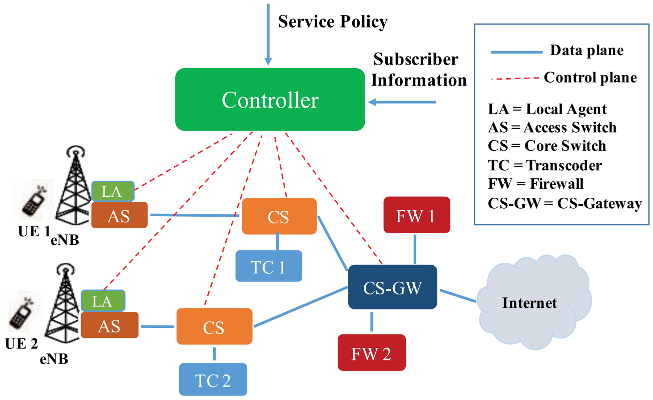

Fig. 8. SoftCell network architecture [80].

2) Partial SDN Adoption: So far, we have described the research work which fully employs SDN technology. There are other approaches that partially apply SDN to separate control and user planes of gateways (i.e., SGW/PGW) [95]. Said *et al.* [96] and Sama *et al.* [97] proposed OpenFlow-enabled EPC architectures, which mainly focus on the separation of control and data planes at SGWs while the PGW is kept unchanged. With the design, the authors claim the benefits of supporting on-demand connectivity services (e.g., load balancing, resiliency) [96] and reducing control signaling load compared to the traditional LTE/EPC architecture [97]. A similar approach is proposed by Pagé and Dricot [101], where the PGW is also reused from the conventional architecture.

As a complementary work, Nguyen and Kim [98], [99] proposed the OEPC architecture which aims at fully separating the control and user planes of both SGW and PGW. By doing so, the signaling load is significantly reduced compared to [97]. A highlighting contribution of this work is to show how the modified OpenFlow protocol operates in the proposed architecture by providing a detailed description of most common procedures that happen in the LTE/EPC network. However, the authors did not solve the problem of scalability due to having a single controller. A similar effort, which applies the SDN concept to minimize the control signaling load is proposed in Mahmoodi and Seetharaman [100].

As a practical approach, Mueller *et al.* [103], Zhao *et al.* [104], and Jain *et al.* [106] provided proof of concept of SDN-based EPC with detailed design, implementation and evaluation. Mueller *et al.* [103] evaluated the proposed system within an existing EPC implementation software, namely OpenEPC [139] while Jain *et al.* [106] validated their proposed system in a self-developed software.Having similar design principles as in [103] and [106], validation tests in [104] are done in the optical network environment. However, these papers still lack of intensive performance evaluation.

3) Summary: We have described all works which use SDN to redesign the current MPC architecture. In these works, by one way or another the authors have already illustrated the benefits of SDN in the MPC architecture and the feasibility of this approach. For example, SDN can help reduce signaling load as in [85] and [97]–[100]. To tackle the problem of having a single centralized controller and to improve the scalability of the network, some solutions have been proposed including constructing a hierarchical control plane architecture such as [83] and [84] or multi-dimensional aggregation mechanisms to reduce the number of forwarding rules such as SoftCell [80], and SoftMoW [81], [82]. However, there is a lack of detailed design of these controllers and how the controllers communicate to each other. Although in this category, there are prototypes implemented to evaluate the proposed schemes, these prototypes are conducted in a small-scale and lack of intensive assessment. The rest is evaluated by using simple analytical models and by simulation. Therefore, more research works need to be done in order to deploy this approach in a real environment.

Table II summarizes all proposals leveraging only SDN into the current MPC architecture. As observed from this table, all research work which fully utilizes SDN have user traffic forwarding based on IP flow entries instead of using GTP tunneling mechanism. While, in partially SDN-enabled approaches, the user traffic routing is done either by using traditional GTP tunneling or by using IP flow entries. In terms of interfaces used between the control and user/data planes, OpenFlow and its variants are the most common protocols. Only research work presented in [105] used JSON-RPC to communicate between the control and user planes. For those using GTP tunnels as the mechanism to route the user data packets, the OpenFlow protocol needs to be extended with some more features like GTP matching fields and GTP-related actions (e.g., encapsulate/decapsulate). For clarity, in the southbound interface column, we use "+" to identify that the OpenFlow version is an extension of the original one. For those using normal IP flow entries, these extensions are not needed. For some proposals that use a general SDN API instead of specific OpenFlow, the requirements for this API are similar as described for OpenFlow.

B. NFV Only

In the following, we will describe the research work, which adopts NFV technology into the current MPC architecture. This type of technology adoption results in two approaches: Full NFV adoption and hybrid NFV adoption. In the following, we first describe the research work according to the full NFV adoption category and then the hybrid scheme will be described.

1) Full NFV Adoption: SoftEPC [107] presented a virtual network of EPC functions over a physical transport network topology. SoftEPC followed the concept of NFV by decoupling the network services and functions from the special purpose hardware. SoftEPC is composed of a collection of General Purpose Nodes (GPN), which typically are core-class commodity servers running hypervisors. A GPN runs virtual instances of EPC entities, e.g., MME, S/PGW. A load-aware algorithm to dynamically place S/PGW functions over the infrastructure is proposed to show that the flexibility and elasticity of SoftEPC outweigh the conventional EPC. However, the authors did not discuss in detail how the instances are instantiated and GPNs are managed.

Taleb [108] envisioned an end-to-end carrier cloud architecture, where all EPC entities are virtualized as VMs running in a distributed manner at different data center (DC) locations. The most distinguishing contributions of this paper is the step-by-step description of how to instantiate VMs and to deploy an entire mobile network including RAN and MPC on the cloud. The VMs and their locations are launched by the mobile service provider through means of a carrier cloud service platform resource controller, based on requirements of the number of subscribers that need to be served at each location. In order to achieve an optimal end-to-end connectivity for UEs, the Follow-Me-Cloud (FMC) concept is introduced. The key idea of FMC is to allow contents and services to follow the user during his/her movement, thus enabling the service continuity and reducing the end-to-end network latency. In addition, the authors briefly described roles of main functional units that are necessary to build an end-to-end carrier cloud architecture integrated with FMC concept such as resource controllers, resource assessors, VNF managers. However, the shortcoming of this work is the lack of evaluation to illustrate how it works in reality. In addition, the detailed design of each functional unit and the interfaces used to communicate between them are not provided.

Similarly, the concept of having distributed DCs to accommodate EPC functions is also introduced in KLEIN [109]. Compared to [108], KLEIN also enables the placement of the data plane entities in a distributed manner. In order to do that, KLEIN proposed a three-level hierarchical resource manager that helps distribute network load across the DCs in an optimal and dynamic manner. In addition, an orchestrator is introduced to allocate network resources and to assign UE's data and network traffic to correct locations. By using a data-driven analysis, the authors proved that KLEIN can almost optimally achieve the benefits of "clean-slate" approaches such as SoftCell [80] and SoftMoW [81], [82] while working within the operational constraints of existing 3GPP standards. A prototype based on OpenAirInterface software [140] is provided to validate the feasibility of KLEIN.

As an attempt to cope with the increase of Machine-to-Machine (M2M) or Machine Type Communications (MTC), Baba *et al.* [110] proposed a multi-vEPC architecture, which is able to provide optimized mobile communication service according to various requirements of M2M services. The M2M services are classified based on their requirements such as policy-based service, mobility required service or IP reachability required service. With different service requirements, the number of EPC VNFs that need to be instantiated is different. An EPC selector is used to select an adequate vEPC to accommodate M2M devices according to their service properties. By doing an experimental validation, the authors showed that the multi-vEPC architecture can significantly help to reduce resource consumption compared to the conventional EPC architecture. However, the authors did not describe in detail the design of the EPC selector, how it works and how they classify the M2M services into different groups and in a static or a dynamic manner.

Another NFV-based EPC is presented in [138]. In this paper, the authors introduced the concept of EPC as a service

(EPCaaS) in which each EPC entity is virtualized as an individual VM communicating to each other using 3GPP standard interfaces. As a practical realization of EPCaaS, Jain *et al.* [106] developed an open source software, which implements most of the conventional EPC functions and run them as VMs in a cloud system. Although these are the simplest ways of virtualizing EPC, Hawilo *et al.* [111] argued that such design can significantly impact the performance, for example, result in a longer communication delay between EPC VNFs. In order to solve that problem, Hawilo *et al.* [111] have grouped several VNFs together on the basis of their interaction and workload and internalize communication between these VNFs, thus reducing the network latency.

While all presented works assume the use of VMs to implement EPC VNFs without considering the performance aspect, Kiess *et al.* [112] provided a comparison of different implementation models of PGW (also applicable to other VNFs) such as device model, cloud-aware model, and software-as-a-service model. Through a cost-based evaluation, they find that the two last models have cost advantages in terms of OPEX saving.

2) Hybrid NFV Adoption: Besides the full virtualization approaches described above, there are several research studies that proposed hybrid approaches that run vEPC along side with a conventional physical EPC architecture. The benefits of NFV by deploying the vEPC along side with a physical legacy EPC system is illustrated in [114]. In this paper, the authors propose several architectural models to offload the mobile traffic from the legacy EPC to vEPC in an on-demand manner. The key idea is to dynamically create a vEPC network architecture and allocate needed vEPC components (e.g., vMME, vSGW, vPGW) when the legacy EPC network capacity is reaching a defined threshold. The three architectural models for offloading includes fully offloading (i.e., create a full vEPC), data plane only offloading (i.e., create a vSGW and a vPGW), and signaling only offloading (i.e., create only a vMME). However, there is no evaluation method of any kind available in this paper. Considered as a fully offloading approach, Gomez *et al.* [115], [116] introduced a flexible management entity (FME), which is composed of a virtual EPC and several necessary functions that are used to let the FME co-work with physical EPCs. The authors analyzed the behavior of the FME by using simulations and showed that the network coverage and capacity can be improved. With this design, the FME can also improve disaster resilience of mobile communication by placing the FME in the RAN area [116].

Alternatively, Taleb *et al.* [117] envisioned an architecture called LightEPC, which is a NFV-based dedicated MPC network for MTC traffic. Unlike [110], the LightEPC operates in parallel with a physical EPC. In the LightEPC architecture, the MTC traffic is classified according to the service types at the edge of the network by a classifier, which is similar to the EPC selector in [110]. A service orchestrator will initiate a LightEPC including virtualized EPC entities for such MTC services. Also, it is possible to dynamically scale LightEPC by a policy orchestration based on changes in the MTC service and the behavior of its devices. A simple analytical model is provided to show the increased number of MTC attach requests that LightEPC can handle compared to the traditional scenario.

While all presented works did not study on how to efficiently manage resources of vEPCs while operating along side a physical EPC, Ren *et al.* [118] and Phung-Duc *et al.* [119] proposed several dynamic auto-scaling algorithms, which are used to dynamically scale EPC VNFs (e.g., MME, S/PGW). By developing analytical and simulation models, the authors showed that these algorithms can significantly reduce operation cost in terms of average response time per user request while providing acceptable levels of performance. However, in order to see actual benefits of these proposed algorithms, it is necessary to evaluate them in a real scenario instead of simulation and analytical models.

3) Summary: All proposals leveraging NFV into the current EPC architecture are summarized in Table II. As observed from this table, all research work which only utilizes NFV have user-traffic routing based on the traditional GTP tunneling mechanism since the functionalities, interfaces and protocols remain unchanged. It also can be seen from the table that fully virtualizing all EPC entities is the most common way of adopting NFV into the EPC architecture. Since the architectures presented in this group did not utilize the SDN technology, the southbound interface is not available. Although this approach is the most simple way of re-designing the current MPC architecture, it is necessary to investigate in more detail the impact of the network performance while moving the EPC entities into the cloud environment. In addition, there is also a need to develop more efficient algorithms such as in Ren *et al.* [118] and Phung-Duc *et al.* [119] and lifecycle management to achieve better resource management and provision while deploying NFV-based EPC systems.

C. SDN and NFV

With the benefits of deployment flexibility and on-demand scalability, many research proposals have been proposed to bring SDN and NFV into the design the current MPC network architecture. As defined in the taxonomy in Section IV-B, such adoption can result in two virtualization paradigms based on the implementation of the data plane: full virtualization and partial virtualization. In the following, we will describe the research work belonging to the full virtualization category and then the research work belonging to the partial virtualization category. It should be noted that some research proposals have considered both partial and full paradigms. These will be described afterwards.

1) Full Virtualization: Akyildiz *et al.* [7], [89] proposed SoftAir, a revolutionary and complete software defined architecture for the next generation (5G) wireless networks which covers elaborated designs for both the RAN and the MPC network. The data plane of MPC is also simplified as a collection of SDN-capable switches. These switches can be virtualized as software instances on top of a switch hypervisor. The control plane consists of two main components: customized applications (e.g., mobility management, QoS routing, billing) and essential management tools. These management tools, including mobility-aware control traffic balancing,

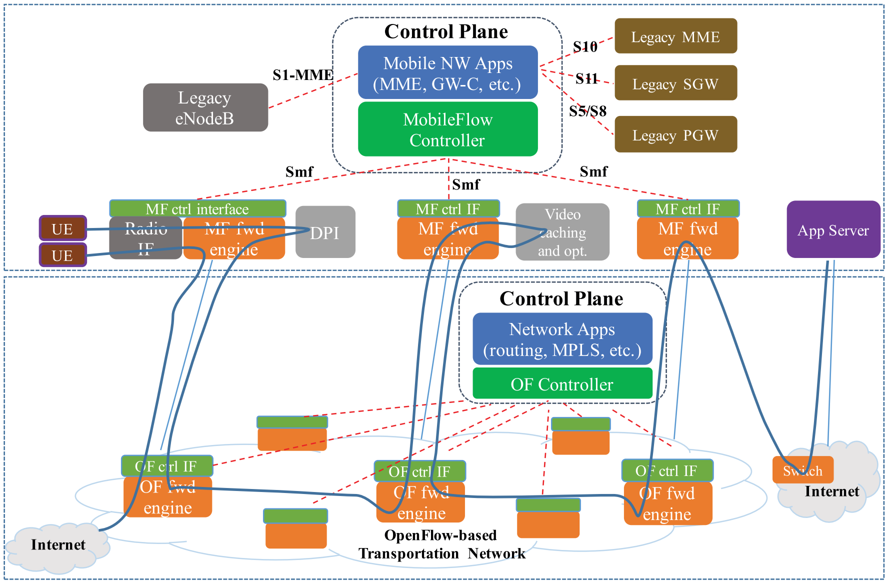

Fig. 9. MobileFlow network architecture [120].

resource-efficient network virtualization, and distributed and collaborative traffic classifier, are proposed to help orchestrate network resource and automate the configuration, management and coordination of software and software interactions. Other important use cases offered by SoftAir including software defined traffic engineering and a mobility management framework are elaborated in the paper. However, the authors did not provide any evaluation in the paper.

SoftNet [91] is another revolutionary architecture towards 5G networking where the core network is designed as an SDN flavor network. The control plane consists of a network controller and a set of network control functions such as a communication control function for mobility management, a policy control function for QoS support, etc. In this architecture, all control functions and data plane entities can be implemented by software. In addition, the control functions of SoftNet are partially offloaded to a server located in the edge of the network which is similar to the concept of local classifier in the SoftAir [7], [89] architecture. However, this server can support the coordination between different RATs (e.g., WLAN, eNB, 5G base stations), while the local classifier mainly classifies and categorizes the incoming traffic into different classes. Unlike SoftAir [7], [89], the authors of this paper have illustrated the benefits of SoftNet over the current LTE network in both qualitative and quantitative ways with a simulation setup.

CleanG [92] is a “clean-slate” simplified software-based architecture for future MPC. The CleanG is designed based on the principles of SDN and NFV. In the CleanG architecture, all conventional EPC components are consolidated on the same host or cluster of hosts as VMs or Docker Containers, and the control plane and data plane are fully separated. These components are implemented in a high performance platform called OpenNetVM and under the management of a NF manager. Compared to SoftAir [7], [89] and SoftNet [91], the authors of this work turn their focus on the design and operation of the control plane protocol in different procedures, which is similar to Nguyen and Kim [99]. A simple analysis has been made to compare the overhead of protocols between CleanG and the traditional EPC.

Pentikousis *et al.* [120] presented the development of a SDN and NFV based carrier-grade mobile network architecture, called MobileFlow as shown in Fig. 9. In this architecture, the MPC network consists of two main components: a MobileFlow Controller (MFC) in the control plane and a collection of MobileFlow Forwarding Engines (MFFEs) in the data plane. The MFC is outlined with necessary functions for managing the entire network such as topology discovery, monitoring and for controlling MFFEs such as tunnel processing, routing and charging. The MFFEs are required to support carrier-grade functionality such as tunnel processing and charging. A new southbound interface called Smf is introduced to communicate between the MFC and the MFFEs. In the NFV context, MobileFlow is a full virtualization approach where the MFC and its control applications are virtualized and the MFFEs employ network virtualization technology. The shortcoming of this work is that the authors did not describe in detail the design of the Smf interface as well as the lack of performance evaluation. A similar architecture is called i-Net, which is proposed in [124]. In this paper, the authors described the evolution of the mobile network from the existing one to i-Net through three main stages. The virtualization of the entire EPC as a vEPC platform is done in the second stage. Then the separation of control and user planes of vEPC into vEPC-C and vEPC-U is done in the third stage. The usage of the GTP tunneling protocol still remains to forward the user data packets in the i-Net architecture. Field trials have been carried out to demonstrate the feasibility and gains of i-Net architecture in cooperative multi point operation scenario in terms of resource management.

As an alternative approach, Basta *et al.* [121] proposed an architecture of virtualized MPC gateways and SDN-based

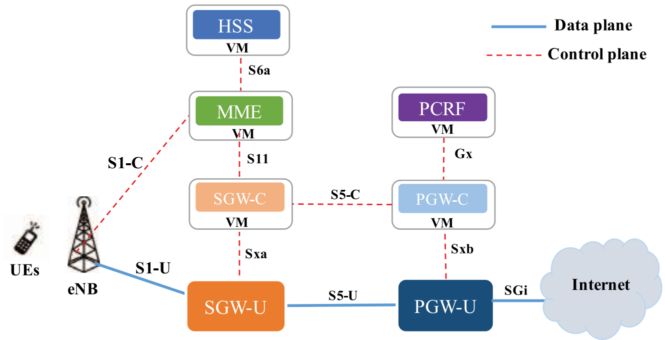

Fig. 10. An example of partial virtualization of 3GPP EPC based CUPS architecture [27].

transport network elements. The control plane is not described in the paper. The data plane entities are virtualized instances running on a data center platform, and they are managed by a data center orchestrator. The SDN-based transport network is used to interconnect these virtualized gateways to the radio access and external IP networks, similar to [120]. As a main contribution, the authors proposed several solutions to find the optimal data center location to host these virtual gateways so that the network load is minimized under a time-varying traffic pattern and a given data plane delay budget. Hahn and Gajic [122] presented an investigation on the elasticity of two different implementation options of SGW and PGW (called GW in general) to adapt to time-varying traffic load and different application profiles: combined GW option, which allow to run a GW in a single VM without splitting the control and user plane, and split GW option, which first separates the control and user plane from each other and then implements them in different VMs. The evaluation results showed that the two schemes have similar performance, but the latter is able to scale resource with finer granularity.

As a practical approach, Haleplidis *et al.* [123] described a proof of concept of an SDN and NFV-based EPC gateways architecture by using a ForCES framework [33]. In this architecture, the control and user planes of SGW and PGW are separated and communicate with each other by using the ForCES protocol. The data plane entities (i.e., SGW-D, PGW-D) are virtualized with a hypervisor. In this proof of concept solution, the GTP tunneling protocol is used to forward the user data packets and its parameters are modeled based on a XML-expressed schema. However, no evaluation results are available from this paper.

2) Partial Virtualization: Fig 10 shows an example of partial virtualization of an EPC-based CUPS architecture, which is currently being developed by 3GPP [27]. In this architecture, the functions of SGW and PGW are split into control functions (SGW-C, PGW-C), which are implemented as VMs, and user plane functions (SGW-U, PGW-U), which are implemented as physical devices.

As an alternative approach, Kempf *et al.* [64] presented an architecture called cloud-based EPC. Compared to the CUPS architecture, the cloud-based EPC introduced a virtualized OpenFlow controller in between the control and data plane. The data plane is realized in OpenFlow switches with GTP tunneling support. The main contribution of this work is the detail description of how OpenFlow protocol should be modified in order to carry GTP packets. Two modifications are made in the OpenFlow flow entry header and OpenFlow v1.2 protocol. However, the authors did not provide any performance results of running the cloud-based EPC with modified OpenFlow protocol. Similar cloud-based EPC approaches are presented in [128]–[133]. Bradai *et al.* [132] proposed a Cellular Software Defined Networking framework, which also aims at "software-defining" and virtualizing the current EPC architecture. The main contribution of this work is the introduction of a new plane called knowledge plane above the application plane. This plane is composed of data blocks which are used to gather and analyze the data received either from the network or the network operator. This plane then communicates with the network controller to exchange significant information for managing and controlling the data plane. Like many other proposals, there is no performance evaluation available in this work.

Sama *et al.* [127] presented a software-defined control architecture of a virtualized MPC network. In this architecture, the SDN concept is also applied to separate the control and user plane of EPC gateways. Instead of being separate components as in the architectures presented above, the control functions are implemented as internal modules called gateway handlers in the centralized controller. These modules are able to perform as a unified handler function in case SGW and PGW are merged. The user plane is a collection of interconnected OpenFlow-enabled switches capable of GTP encapsulation and decapsulation. Other EPC control functions including MME, HSS, PCRF are all virtualized as VNFs. The southbound interface used between the SDN controller and switches is called OF-mpc. The drawbacks of this work are the lack of detail in the design description of OpenFlow extension and the performance evaluation.

Some other SDN and NFV-based MPC architectures employ the Ethrnet-based mechanism to forward the user data packets instead of modifying OpenFlow protocol with GTP tunneling feature such as Kaippallimalil and Chan [128], Ameigeiras *et al.* [129], Cattoni *et al.* [130], and Costa-Requena *et al.* [133]. By using Ethernet-based traffic routing, the packet header size can be significantly reduced, thus reducing the total packet size compared to GTP-based schemes.

As a "clean-slate" approach, which uses both SDN and NFV technologies, Guerzoni *et al.* [72] and Trivisonno *et al.* [73], [74] proposed an SDN-based architecture for 5G networks, which aims to support a heterogeneous set of services efficiently and flexibly. In this architecture, the control plane is made up of SDN platforms and three logical controllers: a device controller, edge controllers, and an orchestration controller. Each controller has a set of control plane modules, which are responsible for different functionalities. For example, the orchestration controller has a resource orchestration module and a topology management module. These modules are virtualized and launched at DCs by using a cloud management platform (e.g., OpenStack). SDN platforms are responsible for managing physical devices to forward the user data packets in the data plane based on IP flow entries instead of GTP tunnels. The data plane has two special nodes: a last hop routing element located at the boundary to the radio access network, and a network entry point located at the boundary to the external IP network. Another major contribution of the paper is the detailed description of protocol operations for each device event such as attachment or service request. In addition, the benefit of this new design in terms of latency reduction has also been shown in [74].

Roozbeh [94] and Hasegawa and Murata [134] provided studies on the signaling overload caused by MTC and M2M services in SDN and NFV-based MPC architectures. In these architectures, the data plane remains in hardware-based devices while the control plane functions are virtualized. In [94], the control plane is composed of virtualized control nodes which contain control functions such as attachment, mobility management, etc. These nodes can be executed separately. Through a mathematical analysis, the authors show that moving the control functions close to the user can significantly reduce control signaling load. Whereas, Hasegawa and Murata [134] has the same architecture as in Fig. 10. By jointly applying SDN principles and bearer aggregation in the EPC architecture, this architecture can significantly increase the number of accommodated M2M terminals without divergence of the data transfer time. The detailed evaluation results are achieved through analytical models.

3) Partial and Full Virtualization: Several proposals describing both partial and full virtualization schemes are presented in [135]–[138]. Heinoen *et al.* [135] presented a solution which brings the cloud computing to the EPC architecture by offering dedicated packet processing resources on-demand for EPC gateway entities. The architecture presented in this paper also has the separation of control and user planes enabled by the SDN concept. The control plane is virtualized in a cloud computing environment while the GTP-enabled user plane processing can be implemented in either general-purpose hardware in the cloud or fast path, dedicated hardware. The main contribution of this paper is the prototype implementation of a switching mechanism, which allows to dynamically switch GTP tunnels between the cloud and the fast path.

Basta *et al.* [136] first analyzed the functionality of SGW and PGW entities and then classified them according to different UE events (e.g., attach, detach, service request, etc.) on the basis of the control- and user-plane separation. Based on that study, the authors proposed four different deployment EPC architectures built on SDN and cloud computing. The first scenario is the full cloud migration in which the control and user plane functions are separated and virtualized in an operator cloud. The second scenario entails migrating only the control plane to the cloud architecture while the user plane keeps running as standard-alone switching fabrics. The third scenario implies shifting a part of the control plane (control signaling) into the cloud environment while the rest is offloaded and run together with data plane functions in a customized hardware platform. The last scenario is a hybrid architecture in which all functions are deployed in both the cloud and hardware-based user plane nodes. The advantages and disadvantages of each scenario are also discussed in the paper. In addition, several frameworks are proposed for matching GTP headers and handling policy and charging functions.

Similarly, Sama *et al.* [137] and Taleb *et al.* [138] have also studied the partial and full virtualization paradigms of EPC gateways (GW) and discussed the advantages and disadvantages of these two. In general, a partially virtualized (PV) paradigm has the data plane traffic running on dedicated hardware and control functions are handled as software in a data center. In a fully virtualized (FV) paradigm all GW functions are virtualized as software instances in a data center environment. These two definitions motivated us to consider them as one criterion to classify the current research work.

4) Summary: All proposals that use both SDN and NFV technologies into the current EPC architecture are summarized in Table II. As observed from this table, the number of research proposals in this group is fairly large compared to those applying only SDN or only NFV. It also can be seen that, only virtualizing control plane functions while keeping the user plane as dedicated hardware is more common than virtualizing all functions. The rationale behind this is the hardware requirements to have high-performance data processing at the user plane. For the fully virtualized paradigm, new advanced technologies need to be adopted into general-purposed servers to accelerate the data processing, for example, OpenvSwitch with the Intel data plane development kit (DPDK) library [141].

In the terms of user-traffic routing, GTP tunneling-based and non-GTP tunneling-based schemes are used in this group. For those keeping GTP tunneling as a routing mechanism, two extensions are required to modify SDN-enabled switches and the SDN API (e.g., OpenFlow): GTP-related matching fields and GTP-related action types as described in [64], [120], [127], and [136]. However, so far these extensions have not been detailed and standardized, thus further elaboration is required in future work. The header overhead caused by the nature of the GTP tunneling protocol is one of the main reasons why some research proposals avoid its use. Instead, they use other routing schemes such as Ethernet-based [128]–[130], [133] or normal flow entries [7], [89], [90], [132].

For the use of a southbound interface, most of the work in this group use OpenFlow (different versions) as the communication channel between the control and user planes. Other works use SDN API southbound interface in general or the ForCES protocol [123]. This variant of southbound interfaces results in the need of a unified and standardized communication protocol between the two planes.

VII. Functional Implementation

In the following, we will review the research work according to the functional implementation dimension in the proposed taxonomy. According to the definition in Section IV-C, there are three models to implement mobile network functions: the “1:1 migration” model refers to the implementation of a network function in one running VM or a network function reused from the conventional EPC architecture, the “splitting” model refers to the decomposition of a network function into a set of subfunctions or modularized elements, and the “merging” model refers to the merging or grouping of multiple network functions into one multi-purposed component.

A. 1:1 Migration

Research proposals purely adopting NFV into the current EPC architecture use the "1:1 migration" model to implement either all functional entities such as SoftEPC [107], Taleb *et al.* [108], [138], Baba *et al.* [110], KLEIN [109], Jeon *et al.* [114] or only gateways like Kiess *et al.* [112] or only control entities like Jeon *et al.* [114].

Research proposals purely adopting SDN into the current EPC architecture use the "1:1 migration" model to implement control functional entities (e.g., MME, HSS) as software applications on top of an SDN controller such as Nguyen and Kim [98], [99]. Other research work such as Said *et al.* [96], Sama *et al.* [97], Mahmoodi and Seetharaman [100], Shanmugalingam and Bertin [102], and Zhao *et al.* [104]. are also belong to the "1:1 migration" group since they reuse some conventional EPC entities in their proposed architectures. For example, Shanmugalingam and Bertin [102] and Zhao *et al.* [104] reuse the MME, and the PGW is kept unchanged in [96] and [97].

For those applying both SDN and NFV technologies into the current EPC architecture, the "1:1 migration" model is mostly used to implement control functional entities as VMs running in the cloud environment. For example, the architectures presented in [64], [120], [127], [128], and [131]–[133] used the "1:1 migration" model to implement the MME and HSS entities as individual VMs. Basta *et al.* [121] used this implementation option to implement SGW and PGW as individual VMs running in mobile operator data centers. All the research works which employed the "1:1 migration" model to implement mobile network functions in their architecture are summarized in Table III.

B. Splitting

Currently, the "splitting" functional implementation model is gaining a lot of attention and is being developed by an European project called 5G CONFIG (COntrol Networks in FIve G) [75] under the umbrella of European 5G Public Private Partnership (5GPP) [142]. The main objective of this project is to develop a modular functional and access-agnostic 5G control plane architecture for fixed mobile convergence networks supporting a wide variety of devices, services and applications for current and future user needs [93].

As the relation to this project, Guerzoni *et al.* [72] and Trivisonno *et al.* [73], [74] presented a detailed design of an SDN-based architecture for 5G networks, which has been described in Section VI-C in terms of technology adoption. In regard to the functional implementation, the control plane of this architecture is composed of modularized functions such as connection management, mobility management, authorization and authentication, etc., which can be seen as the functional decomposition from the conventional EPC control entities (e.g., MME, HSS). This design offers the most adaptability, flexibility and portability in the control plane. Similar approaches are presented in [86] and [94].

Lindholm *et al.* [84] presented two approaches to modularize and re-factor the mobile network functions. In their work, the analysis of space states in EPC entities (i.e., MME, SGW, PGW) with respect to four different UE events and signaling procedures including initial attachment, idle, wake-up, and mobility is carried out. As the first approach, each of these events is managed by a module responsible for that event. For example, the MME would have a module responsible for the initial attachment event, a module responsible for idle event, etc. In the second approach, the states inside each network function are first grouped into three different variable groups: location, control plane state, and user plane state, and then each variable group is managed by a module responsible for that group. For example, the MME would have a module for the location state, a module for control plane state, etc. Similar studies have been done by Basta *et al.* [136] and Sama *et al.* [137].

For those applying NFV into the current EPC architecture, (i.e., vEPC), there is a way to split an EPC VNF into multiple elements and implement them on multiple individual VMs. Such a way is illustrated in [138]. In detail, each EPC VNF (e.g., MME VNF) can be decomposed into three element types: a front-end, a worker, and a session database. The front-end entity is responsible for the communication interfaces towards other entities. The worker entity is a stateless component, which implements the logic functionalities of that specific EPC VNF. The session database is responsible for storing the user session state. This feature makes the workers stateless. Similar concepts that focus on splitting MME have been proposed in [143]–[147] and will be discussed further in Section IX-B.

Another study on splitting EPC network functions is discussed in [125]. In this paper, the authors proposed a novel PGW architecture, which adopts both SDN and NFV technologies. The main objective is how to make the user plane of PGW (i.e., PGW-U) more scalable. In this sense, the PGW-U is decomposed into multiple subcomponents: a PGW-U downlink switch, a PGW-U uplink switch, and a cluster of packet processing units (PPUs) located in a resource pool, which is controlled by an orchestrator. The design with the introduction of PPU clustering concept can definitely improve the scalability of the user plane.

1Last but not least, we consider all other research work that are "clean-slate" designs such as CellSDN [79], SoftCell [107], SoftMow [81], [82], SoftAir [7], [89], SoftNet [91], etc., as the "splitting" approach. Indeed, in these architectures, some control applications defined on top of SDN controllers such as routing, mobility management, QoS managements, etc., could be seen as sub-functions which are decomposed from EPC functions. All the research works which employed the "splitting" model to implement mobile network functions in their architecture are summarized in Table III.

C. Merging

Fig. 11 shows the architecture for the next generation network (NextGen) or 5G, which is currently being discussed within 3GPP SA2 Working Group [26]. In this figure, the conventional entities such as MME, SGW, PGW no longer exist. All the control functions are merged into a NextGen

TABLE III
COMPARISON OF CURRENT RESEARCH INITIATIVES IN TERMS OF FUNCTIONAL IMPLEMENTATION AND DEPLOYMENT STRATEGY

References	Functional Implementation			Deployment Strategy				Network Slicing	Scala-bility	Eval. Methods
				Control Plane		User/Data Plane				
	1:1 Migr.	Split	Merge	Central.	Distri.	Central.	Distri.			
CellSDN [79], SoftCell [80]	–	✓	✓	–	✓^{H}	–	✓	✓	High	Proto. & Simul.
SoftMow [81], [82]	–	✓	✓	–	✓^{H}	–	✓	✓	High	Proto. & Simul.
Yazici et al. [83]	–	✓	✓	–	✓^{H}	–	✓	✓	High	Prototype
Lindholm et al. [84]	–	✓	–	–	✓^{H}	–	✓	–	High	No
Chourasia et al. [85]	–	–	✓	✓	–	–	✓	–	Low	Analy.& Simul.
Marquezan et al. [86]–[88]	–	✓	✓	✓	–	–	✓	✓	Low	Prototype
SoftAir [7], [89]	–	✓	✓	–	✓^{H}	–	✓	✓	High	No
SoftNet [91]	–	✓	✓	–	✓^{H}	–	✓	–	High	Simulation
CleanG [92]	–	–	✓	–	✓^{H}	–	✓	–	High	Analy. Model.
Einsiedler et al. [93]	–	✓	✓	–	✓	–	✓	✓	High	No
Roozbeh [94]	–	✓	–	–	✓^{H}	✓	–	✓	High	Analy. Model.
Trivisonno et al. [72]–[74]	–	✓	–	–	✓	–	✓	✓	High	No
Yang et al. [90]	–	✓	✓	✓	–	–	✓	✓	Low	Simulation
Hampel et al. [95]	✓	–	–	✓	–	–	✓	–	Low	No
Said et al. [96]	✓	–	–	✓	–	✓	–	–	Low	No
Sama et al. [97]	✓	–	–	✓	–	✓	–	–	Low	Analy. Model.
Nguyen et al. [98], [99]	✓	–	–	✓	–	–	✓	–	Low	Analy. Model.
Mahmoodi et al. [100]	✓	–	–	✓	–	✓	–	–	Low	Analy. Model.
Page et al. [101]	✓	–	–	✓	–	✓	–	–	Low	No
Shanmugalingam et al. [102]	✓	–	–	✓	–	–	✓	–	Low	Prototype
Mueller et al. [103]	✓	–	–	✓	–	✓	–	–	Low	Prototype
Yongli et al. [104]	✓	–	–	✓	–	–	✓	–	Low	Prototype
Osmani et al. [105]	✓	–	–	✓	–	✓	–	–	High	Prototype
Jain et al. [106]	✓	–	–	✓	–	✓	–	–	Low	Prototype
SoftEPC [107]	✓	–	–	–	✓	–	✓	–	Low	Simulation
Taleb et al. [108]	✓	–	–	✓	–	–	✓	–	Low	No
KLEIN [109]	✓	–	–	–	✓^{H}	–	✓	–	High	Proto. & Simul.
Baba et al. [110]	✓	–	–	✓	–	✓	–	–	Low	Proto. & Simul.
Hawilo et al. [111]	–	–	✓	✓	–	✓	–	–	Low	Analy. Model.
Kiess et al. [112]	✓	–	–	✓	–	✓	–	–	Low	Simulation
Medhat et al. [113]	✓	–	–	✓	–	✓	–	✓	Low	Prototype
Jeon et al. [114]	✓	–	–	✓	–	✓	–	–	Low	No
FME [115], [116]	–	–	✓	–	✓	–	✓	–	Low	Simulation
Taleb et al. [117]	✓	–	–	✓	–	✓	–	–	Low	Analy. Model.
Ren et al. [118], [119]	✓	–	–	✓	–	✓	–	–	High	Analy. & Simul.
MobileFlow [120]	✓	–	✓	✓	–	–	✓	–	Low	Prototype
Hahn et al. [122]	✓	–	–	✓	–	–	✓	–	Low	No
Haleplidis et al. [123]	✓	–	–	✓	–	✓	–	–	Low	Prototype
i-Net [124]	✓	–	–	✓	–	✓	–	–	Low	No
Basta et al. [121]	✓	–	–	✓	–	✓	–	–	Low	Simulation
An et al. [125]	✓	✓	–	✓	–	✓	–	–	High	Simulation
Kempf et al. [64]	✓	–	–	✓	–	–	✓	–	Low	No
Hahn et al. [126]	✓	–	–	✓	–	–	✓	–	Low	Simulation
Sama et al. [127]	✓	–	–	✓	–	–	✓	–	Low	No
Kaippallimalil et al. [128]	✓	–	–	✓	–	–	✓	–	Low	Analy. Model.
Ameigeiras et al. [129]	✓	✓	–	–	✓^{H}	–	✓	–	High	No
Cattoni et al. [130]	✓	✓	–	–	✓^{H}	–	✓	–	High	No
MobiSDN [131]	✓	–	–	–	✓^{H}	–	✓	–	High	No
Bradai et al. [132]	✓	–	–	✓	–	–	✓	–	High	No
Costa-Requena et al. [133]	✓	–	–	✓	–	–	✓	–	Low	Prototype
Hasegawa et al. [134]	✓	–	–	✓	–	–	✓	–	Low	Analy. Model.
Heinoen et al. [135]	✓	–	–	✓	–	–	✓	–	Low	Prototype
Basta et al. [136]	–	✓	–	✓	✓	✓	✓	–	High	No
Sama et al. [137]	–	✓	–	✓	–	–	✓	✓	High	No
Taleb et al. [138]	✓	✓	✓	✓	–	✓	–	–	High	No

"✓" indicates that the attributes are provided or applicable in the research work.
"✓^{H}" indicates that the control plane is a hierarchical architecture
"–" indicates that the attributes are unspecified or non applicable in the research work

core network function while all the user plane functions are merged into a NextGen user plane function. The HSS function is replaced by a subscriber management entity while PCRF is substituted by a policy control entity. The conventional interfaces are also modified in the NextGen architecture such as S11 interface is replaced by a new named NG4, S5 interface is replaced by NG6 interface, and so on. However, the detail design of each component and corresponding interfaces are still under discussion and development. It is expected to be released in 3GPP specification Release 14 in June 2017.

Gomez *et al.* [115], [116] proposed a framework called FME, a Flexible Management Entity for virtualizing EPC as already described in Section VI-B in the aspect of technology adoption. While talking about the functional implementation, this FME entity implements the most functionalities and fundamental operations of the conventional EPC architecture and is deployed simultaneously with a physical EPC. It provides the connectivity of the UEs to external packet data networks as the conventional EPC does and it is able to perform the handover of all the user plane functions to the physical EPC when it is required. A similar work is presented in Chourasia and Sivalingam [85], where all functionalities of the EPC control plane are merged into a single element called EPC controller.

In the MobileFlow architecture [120], this type of functional implementation is also mentioned. The authors proposed a 1:m mapping model, which allows all mobile network applications (e.g., MME, SGW-C, PGW-C) on top of the MFC controller to be merged into one novel network application. Taleb *et al.* [138] also considered the "merging" model by presenting a N:1 mapping model to implement one of EPCaaS scenarios in which all the functionalities of conventional EPC are merged into one virtual component, namely merged-EPC. A state database is implemented separately to maintain the user session state.

For those purely leveraging NFV into the current EPC architecture, another option to implement the network function is to group several network functions together based on their interactions and workloads. This grouping scheme was first introduced in [111]. In this paper, the authors proposed to group EPC VNFs into four different segments. The first segment comprises the implementation of a MME entity together with a HSS front-end entity. The rationale behind this is that these entities share in common functions of authentication and authorization. The second segment comprises an implementation of a serving GPRS support node and a home location register front-end entity. These two functions are mainly used in GPRS core network to transmit IP packets from 2G and 3G mobile networks to external ones. They share in common the functions of authentication and authorization. In the third segment, the PGW is migrated with the SGW since they are both responsible for user packet processing at the data plane. The last segment comprises the implementation of all functions related to policy and charging functions such as PCRF, online/offline charging system. This design has been proven to minimize the signaling load caused by the communication between EPC elements.

Similar to the splitting model, we also classify some revolutionary approaches such as CellSDN [79], SoftCell [107], SoftMow [81], [82], SoftAir [7], [89], SoftNet [91], etc., as the merging model. This classification is possible to implement all mobile network functions as a single complex application on top of the SDN controller. All the research works which employ the "merging" model to implement mobile network functions in their architectures are summarized in Table III.

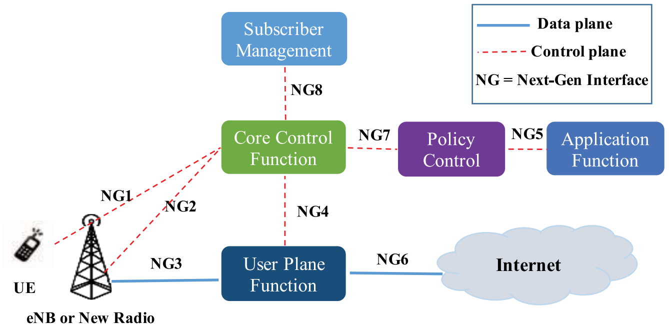

Fig. 11. NextGen (5G) core network architecture proposal by 3GPP [26].

D. Summary

As observed from Table III, all the research works in the "1:1 migration" group belong to the evolutionary approach, while those in the two other groups belong to either evolutionary or revolutionary approaches. The proposals with two checkmarks (✓) in Split and Merge columns have "clean-slate" and purely SDN-based MPC architectures. As an exception, the work in Taleb *et al.* [138] has three checkmarks (✓) because they presented several scenarios which cover all three implementation options. From the preceding survey, we can conclude that the "1:1 migration" is the simplest functional implementation model, the "splitting" model offers the most scalability and flexibility, and the "merging" model can help to reduce the communication delay between network elements by internalizing communication between them. In other words, the "1:1 migration" model and the "merging" model have low scalability while the "splitting" model results in high delay due to inter-element communication. Therefore, there is a need to understand the trade-off between these models and to decide which model we should follow depending on a certain situation and requirements.

As described above, the scalability of a network may be affected by the choice of implementation model. Indeed, the proposals following the "splitting" model have highest scalability. They are described as "High" in the last column of the table, while others are described as "Low" identifying that they have low scalability or is not addressed. However, the scalability is also affected by the choice of deployment strategy, which is covered in the following section. In addition, the "splitting" implementation model may empower the development of a network slicing paradigm. Indeed, depending on service and application requirements, each slice can be made of several modularized elements in the control plane and several other elements in the data plane. The network slicing can also be done by using the concept of a network hypervisor [148] such as FlowVisor [149] or OpenVirteX [150] in full-SDN MPC architectures. Researching on network slicing related topics has currently gained a lot of attention from both academia and industry, and will be discussed further in Section IX-E as one future research direction.

VIII. Deployment Strategy

In this section, we will review the research works in terms of deployment strategy used in their proposed architectures.

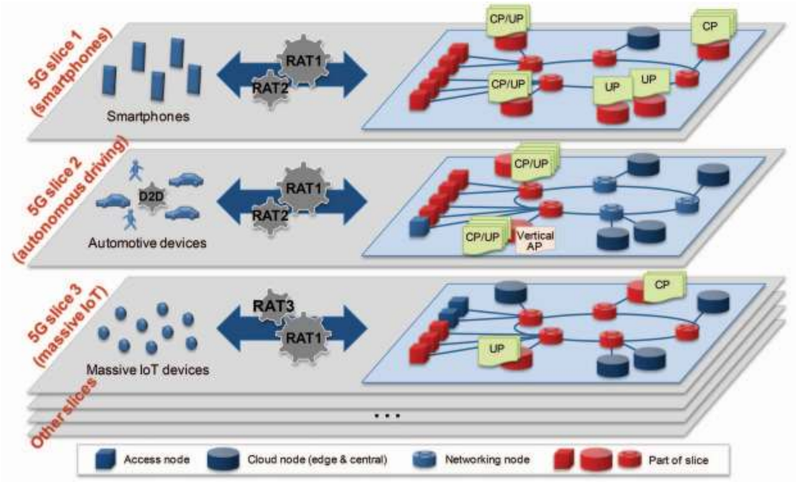

Fig. 12. 5G network slices implemented on the same infrastructure [2].

According to the definition in Section IV-D, there are four possible ways of placing the network functions across the network infrastructure in regard to the control and data planes: centralized and distributed control plane functions, and centralized and distributed data plane functions. Figure 12 illustrates an example of placing control and data/user plane functions in each service-oriented slice of the 5G network infrastructure.

A. Control Plane

In the following, we first describe the research works which centrally deploy control plane functions and then the research works deploying the control plane functions in a decentralized or distributed manner are described.

1) Centralized Control Plane: This deployment option reflects de facto nature of SDN architecture and its design principles, where all network control plane functions are separated from the data plane and are implemented as software modules of a centralized controller. For example, in [96]–[99], control functions like MME, SGW-C, PGW-C are packaged as a part of an SDN controller such as an OpenFlow controller or Mobile controller. Alternatively, Chourasia and Sivalingam [85] puts all control functions into a centralized entity called EPC controller.

Another option to centralize the control plane functions is to virtualize and place them as virtual instances in a centralized cloud data center. For example, in [64], [124], [133], and [134] the control functions such as SGW-C, PGW-C are decoupled from the data plane and run as VMs as other control functions (i.e., MME, HSS) in the cloud data center. In [110], [111], [118], and [121] the control plane of SGW and PGW are still coupled to the data plane, and they are together migrated from hardware to run as VMs in the centralized cloud data center.

In some research proposals, the control plane functions are centralized, but not as a part of a centralized controller. Instead they are separately centralized entities and communicate with the centralized controller through external interfaces. For example, in Sama *et al.* [127] and MobileFlow [120], the control plane functions are virtualized and communicate with the centralized controller (i.e., SDN controller in [127], and MobileFlow controller in [120]) via external interfaces, which can be standard interfaces reused from the conventional EPC (e.g., the S11 interface between the MME and the SDN controller).

2) Distributed Control Plane: To deal with the scalability problem caused by a single centralized controller, many research proposals have decentralized a subset of control functions to the edge of the network, thus forming a hierarchical deployment of the control plane. In CellSDN [79], and SoftCell [80], a local control agent is used to cache packet classifiers for attached UEs under the control of the central controller, to minimize interaction with the central controller. In MobiSDN [131], that entity is called edge controller, whose function is a part of the central controller. Similarly, a local classifier is introduced in SoftAir [7], [89], which collaborates with a global traffic learner at the central controller to achieve fast, fine-grained and accurate traffic classification. In the CSDN architecture [132], a program, which locally resides in the data plane, performs some simple tasks under the supervision of the central controller such as notifying the controller if the traffic exceeds a pre-defined threshold, etc.

As described in Section VII, Lindholm *et al.* [84] proposed a solution to re-factor the control plane of EPC by analyzing and classifying network states into different groups according to different UE events. Based on this state and event classification, a corresponding software module is implemented inside a mobile-core controller to serve each event. Some of the events are served by an agent of this controller called mobile core control agent. This helps to offload a subset of control modules from the central controller to the radio access network. The central controller and the agent communicate with each other according to a publish-subscribe model. For example, the agent is configured as a publisher for state changes related to the events associated with the change in UE state, while the central controller is configured as a subscriber to these state changes.

In [94], the control plane is composed of control nodes, which have all required control functions such as attachment, mobility management, and authentication. These control functions can be executed in a different physical location (e.g., the core network or the radio access network), while maintaining a parent-child relationship. For example, the child function is located closer to UE, while the parent function stays in the core network and receives the update from the child.

Another decentralized design of the control plane is presented in SoftMow [81], [82]. In this architecture, the control plane is recursively constructed as a tree with several regional levels: leaf, parent, and root. At each level, there is a controller responsible for managing all network devices in that region and abstracting network resources to the controller in the upper level. The number of levels, the number of network devices per controller, and the location of each controller are determined based on latency budgets of control functions or size of the physical topology.

In Wang *et al.* [91], the control plane of the SoftNet architecture is also decentralized. The control functions such as mobility management, and forwarding management are offloaded to an access server as a decentralized control function and a gateway control. The decentralized control function is responsible for location management and handover management of mobile terminals when they move from a radio access point to another served by the same access server. The gateway control function allows offloading user traffic locally instead of traversing the core network.

Guerzoni *et al.* [72], Trivisonno *et al.* [73], [74] and Einsiedler *et al.* [93] presented an approach to modularize the control plane of the 5G network architecture. In this work, the control plane is split into atomic elements (e.g., connectivity management, mobility management) and these elements are placed at three different controller levels: a device controller, edge controllers and an orchestration controller. A similar concept, but without a detailed procedure description, is proposed in [83].

The 5G architecture presented in [129] and [130] has also decentralized the control plane. The authors claimed to decentralize part of MME functionalities to the access network. In this sense, the MME is decomposed into a distributed MME located at a regional distributed cloud and a centralized MME placed in a centralized national cloud. In addition, the SDN control used in this architecture is also deployed in a distributed fashion with three different controller levels: a local SDN controller located at the access cloud, a regional SDN controller located at the regional cloud, and a centralized SDN controller located at the national cloud center.

B. Data/User Plane

In the following, we first describe the research works in which the data plane functions are deployed in a central manner and then the research works deploying the data plane functions in a decentralized or distributed manner.

1) Centralized Data Plane: The centralization of the data plane can be represented in two different groups: the group comprising of virtualized data plane instances in a centralized data center, and the one comprising of data plane entities reused from the conventional EPC.

As described in Section VI-C, Basta *et al.* [136] has illustrated four different scenarios for software-defining and virtualizing the EPC architecture. In the first scenario, which is a full cloud migration architecture, the data plane is centralized together with the control plane in a centralized operator's cloud. In the fourth scenario about scenario-based cloud architecture, a part of data plane functions is migrated and placed along with the control plane functions in a centralized operator cloud.

The data plane entities of the multi-vEPC architecture proposed in [110] are software-based appliances, which are placed in a virtualization infrastructure. Based on the M2M service requirements, these software-based appliances (i.e., vSGW, vPGW) are instantiated together with the control plane virtual instances (i.e., vMME) to form a dedicated vEPC architecture for that M2M service. For example, a dedicated vEPC consisting of vMME, vPGW, and vSGW is deployed to serve M2M services which require IP reachability, while a dedicated vEPC for roaming based M2M services is instantiated with only vMME and vSGW. A similar approach is presented in [117]. In this paper, depending on the underlying MTC applications, the data plane virtual instances of the LightEPC architecture called soft PGW and soft SGW are launched from an image repository in a centralized data center by a cloud controller (e.g., OpenStack). The data plane functions of the architectures proposed in [111], [112], [114], [118], [121], and [124] are also virtual instances operating either in an NFV infrastructure or a cloud data center.

The second group which have centralized data plane includes the proposals [96], [97], [100]. In these works, the data plane consists of functional entities reused from the conventional EPC architecture. The PGW is kept the same as the one in [96] and [97] while both SGW and PGW are reused in [100]. These entities are neither virtualized as instances in a distributed cloud data center nor separated control and data planes. Therefore, they still keep the nature of data plane centralization of the conventional EPC architecture.

2) Distributed Data Plane: The distributed data plane comprises of two different groups: distributing the data plane as SDN forwarding elements and distributing the data plane as virtualized instances in distributed data centers or in distributed computing nodes.

Distributing the data plane on SDN-capable forwarding elements is the most common way to distribute the data plane functions in the approaches that use the SDN concept since the data plane is completely decoupled from the control plane. In some "clean-slate" proposals such as CellSDN [79], SoftCell [80], Chourasia and Sivalingam [85], SoftAir [7], [89], Yang *et al.* [90], Shanmugalingam and Bertin [102], etc., these SDN-capable forwarding elements are OpenFlow-enabled L2 switches or L3 routers. In some proposals such as Hampel *et al.* [95], Nguyen and Kim [98], [99], Sama *et al.* [127], Heinoen *et al.* [135], etc., these OpenFlow-enabled switches are extended with GTP tunneling capability to route GTP packets. As described in Section VI-C, these switches can run either on dedicated hardware platforms or general-purpose servers.

In the second group, the data plane functions of the KLEIN [109] architecture and the carrier cloud architecture [108] are virtual or soft instances placed in different data centers. Alternatively, in the SoftEPC architecture [107], the data plane functions are virtualized in general purpose nodes (GPNs) distributed over the network. Similar to SoftEPC, the data plane functions in [115] and [116] are packaged together with the control functions in the FME entity placed distributedly at the radio access network.

C. Summary

A representative summary of the above described works in terms of deployment strategy is shown in Table III. It should be noted that the works that have the hierarchical design of the control plane is marked as ($\checkmark^H$) in the distributed control plane column of the table. As an exception, the work in [136] has two checkmarks (✓) in the control plane category because the authors presented several deployment scenarios which cover both centralized and distributed deployments of the control plane. Overall, we can observe from this table that most of the research works have logically centralized control planes and distributed data planes in their proposed architectures, according to the SDN principles. It is obvious that shifting all control plane functions into a centralized controller provides the global view of the whole network states thus easy to manage and control. However, it results in the long configuration delay which might not be suitable for some services, and especially it results in the scalability problem caused by signaling overload. This problem has been solved by some proposals with a hierarchical control-plane architecture or multiple controllers communicating to each other. These proposals are described as "High", which identifies they have higher scalability than others ("Low") in the scalability column in Table III. Although decentralizing the control plane would overcome the single point of failure, it still faces some other problems such as how to synchronize between controllers. Another way to solve this problem is proposed in [105] and [144] by introducing a signaling load balancer in the control plane. This problem will be discussed further in Section IX-B as one research direction for the future.

For the data plane, most of the proposed architectures have a distributed deployment of the data plane. As an exception, the work in [136] has two checkmarks (✓) in the data plane category because the authors presented several deployment scenarios which cover both centralized and distributed deployments of the data plane. Distributing the data plane closer to the user or the edge of the network would be beneficial to services which require traffic offloading, low latency or high mobility such as low latency video processing, surveillance, etc. However, it results in challenges of policies and charging enforcements, which can be solved by centralizing the data plane. Therefore, it is necessary to find an optimal solution to centralize and decentralize the data plane. This is considered as one future research direction and will be discussed further in Section IX-C. Besides the scalability problem of the control plane described above, the scalability in the data plane has also been considered in [125] by introducing the concept of clustering of packet processing units. We believe that this topic would also be an interesting research problem that needs to be further investigated.

Last but not least, we also compare all surveyed works in terms of their ability to inter-work with or the backward compatibility to the legacy EPC architecture. The backward compatibility mostly depends on the choices of architectural model and technology adoption. It can be seen from Table III that, most of "clean-slate" or revolutionary approaches has no capability to interoperate with the legacy EPC architecture. Other evolutionary approaches, which has a clean-slate data plane (i.e., without GTP tunneling capability) are not considered to be compatible with the legacy EPC architecture. It also means that the proposed architectures which have extended the forwarding devices and protocols to be able to route GTP packets are classified as compatibility with a checkmark (✓) in the table.

IX. Research Challenges and Future Directions

From the preceding classification of contemporary work, we can see that research in software-defining and virtualizing the MPC networks with SDN and NFV have indeed gained momentum with a large number of new architectures proposed. Through brief descriptions highlighting main contributions, advantages and disadvantages of each proposed architecture and a comprehensive comparison between them shown in two tables, we make several observations such as the trade-off between design and development choices in terms of architectural approach, technology adoption, functional implementation or the trade-off between different functional placement strategies. These choices lead to several challenges and issues that need to be solved, including the design of southbound and northbound interfaces, the scalability, reliability, and high availability of the network, the placement optimization and resource allocation, the management and orchestration, the capability of sharing and slicing the network, and the challenges in terms of network performance requirements and evaluation methodologies. The first issue is raised from the variety of southbound interfaces used in the research work and the lack of standardization. The second challenge derives from the choice of technology adoption, functional implementation as well as the deployment strategy. For example, adopting the SDN technology with the introduction of a centralized controller causes the problem of a single point of failure, thus reducing the scalability and reliability of the network. The third issue reflects the deployment strategy dimension of the survey taxonomy. It is important to find an optimal placement of network functions and to allocate resources to them in an efficient way. The fourth challenge comes from the choice of technology adoption where managing and orchestrating resources for different types of network functions (e.g., SDN forwarding devices, SDN controllers, and VNFs) are challenging tasks. The new emerged network slicing technology also imposes several challenges such as managing and orchestrating the slice, inter-slice communication, and guaranteeing the isolation between slices, etc. The last issue comes from the limitation of evaluation methodologies and benchmarking tools. In addition, the performance of a network function in the virtualized environment is also a big concern.

We believe that these challenges and open issues are critical and need to be further investigated in order to fully realize the potential benefits of SDN and NFV. In the following we discuss them in detail and offer a perspective on the future research directions in this area.

A. Southbound and Northbound Interfaces

Separating the control plane and the data plane enables the network programmability which distinguishes SDN from the traditional network. As a consequence, a communication channel which is called southbound interface is required for the controller in the control plane to communicate and program the forwarding devices in the data plane. Currently, the OpenFlow protocol and its versions are the most commonly used southbound interface in the SDN paradigm. From the previous survey section, we have seen that most of the research proposals are also supposed to use OpenFlow as the communication protocol when they adopt SDN into the MPC architecture. Among those proposals, some discuss the need of extending OpenFlow to be able to forward GTP packets and give some potential extensions, while some use standard OpenFlow. Although it has been mentioned in [127] that the extensions in the OpenFlow protocol are under development in the ONF WMWG working group, they have not yet been released. Besides the use of OpenFlow as the southbound interface, other surveyed studies either introduced a new name for that interface (e.g., Smf in [120]) or just simply used “SDN API” as a general term for the southbound interface. However, they are all conceptual designs and lack of clarify.

Another interface, which is also important in the SDN paradigm, is the northbound interface between the network application plane and the control plane. This interface helps the application developers manage and program the underlying network by using different programming languages, which have been surveyed in [154] such as JSON-RPC, Frenetic, Procera. There is a few research works that mention the use of a northbound interface. For example, Heinoen *et al.* [135] used JSON-RPC to transfer information between the S/PGW control functions and the OpenFlow controller, and REST is used in [98] and [99]. It means that this interface is still not given much attention to so far. Therefore, it is important to have a complete and standardized design of both the northbound and the southbound interfaces in order to achieve a complete SDN solution in the MPC network, to manage not only unicast services, but also multicast or broadcast [155]. In addition, the detailed design of east-west interfaces used between SDN controllers, which have been mentioned in several proposals to improve the control plane scalability, also needs to be considered.

B. Scalable, Reliable, and High Available Design

This section presents Key Performance Indicators (KPI) that we need to consider when redesigning the current MPC architecture by using SDN and NFV, especially while designing the control plane. These include scalability, reliability, and high availability.

1) Scalability: As previously described, the MME entity, which is the main control entity of the conventional EPC architecture, is responsible for handling all control signaling requests from the user devices sent through the radio access network. According to a Nokia white paper [156], the control signaling traffic is estimated to grow 50% faster than data traffic and this trend is continuing since there are a large number of new types of user devices being attached to the mobile network (e.g., M2M, IoT, etc.). As a consequence, the performance of the MME entity and the control plane overall is significantly affected. In addition, a study by Rajan *et al.* [157] has shown that, the overload at the MME can lead to bottlenecks at other entities such as the SGW because a large portion (41%) of signaling events arrived at the MME are also handled by the SGW. Therefore, it is critical to have a scalable design of MME and the whole control plane to cope with the sudden change in the control signaling load. In other words, improving the scalability of the control plane is necessary to improve the scalability of the entire network.

With the adoption of SDN and NFV in the MPC architecture, the control plane architecture has been changed. As observed from the survey section, the control plane can be: the SDN controller in the full-SDN MPC scenario; the SDN controller and other control plane VNF nodes (e.g., MME, HSS, S/PGW-C) in the SDN/NFV-based EPC scenario; or only the control plane VNF nodes (i.e., MME, HSS) in the NFV-based EPC scenario. Hence, the scalability problem of the control plane turns into the scalability of the SDN controller and the scalability of the VNF nodes.

So far, several studies have been conducted in order to improve the scalability of MME VNF node, as summarized in Table IV. In these studies, the MME VNF can be constructed as two-tier or three-tier architectures. In two-tier architectures, the MME VNF is mostly decomposed into a front-end Signaling Load Balancer (SLB in [147]) or MME Load Balancer (MLB in [144]) and a cluster of MME processing entities (MMPs). As an exception, the DMME architecture in [151] and [152] separates the MME VNF into DMME nodes, which are similar to MMPs, and a reliable object storage (ROS) subsystem. In three-tier architectures such as Takano *et al.* [153], Premsankar *et al.* [143], and vMME [145], [146], there is also a front-end load balancer, which is the same as the one in the two-tier architecture. However, all user-related session states, which are previously stored the MMP entities of the two-tier architecture are now stored in a separate database (DB) or a session database (SDB). As a result, the MMPs become stateless components such as Workers in [143] or MME Service Logics (SL) in [145] and [146]. By splitting the MME VNF into two or three functional layers, it is easy to scale in/out the resource of each component independently and effectively, even without affecting on-going sessions (e.g., in the three-tier architecture). However, we believe that more work needs to be done in this area to achieve a complete scalable design of the MME VNF node. Some suggestions for future work include be making the load balancing algorithm at the front-end nodes more intelligent instead of using simple algorithms such as round-robin, making the provisioning process of the MMP cluster more

TABLE IV
A SUMMARY OF CURRENT RESEARCH ON SCALABLE MME DESIGN

Approaches	Architecture Types	Main Components	Geographic Distribution	Load Balancing Mechanism	Reliability	High Availability	Evaluation Method
DMME [151], [152]	2-Tier	DMME and ROS	Yes	Not Mentioned	High	Low	Analy. Model.
Yusuke et al. [153]	3-Tier	MME, Proxy, and DB	No	Group Hashing	High	Low	Prototype
Gopika et al. [143]	3-Tier	FE, Worker, and DB	No	Round Robin	High	Low	Prototype
SCALE [144]	2-Tier	MLB and MMP	Yes	Consistent Hashing	Low	High	Prototype
vMME [145], [146]	3-Tier	FE, MME SL and SDB	No	Not Mentioned	High	N/A	Analy. Model.
Yousaf et al. [147]	2-Tier	SLB, MMP	No	Round Robin	Low	N/A	Analy. Model.

automatically, and making the scaling process of MMP in the cluster more elastically.

For the scalability of the SDN controller, having a centralized SDN controller in the control plane results in the most common scalability problem of the SDN paradigm. There are a large number of solutions proposed to solve this problem in wired SDN research [39], [63], [158] such as horizontally distributed SDN controllers where multiple controllers are organized in a flat control plane (e.g., DISCO [159]) and vertically distributed controllers where multiple controllers are organized in a hierarchical control plane (e.g., Orion [160]). As we saw in Section VIII, there are several proposals using the second scheme to offload subfunctions of the central controller to a classifier at the edge of the network, thus reducing the number of interactions with that controller. However, these are all conceptual designs which need to be further detailed.

Last but not least, the data plane scalability, which refers to the ability to cope with the increase of user data traffic with high performance requirements, would also need to be further studied in the future. So far, there are only a few research works such as An *et al.* [125] and Taleb *et al.* [138], which have addressed this problem by decomposing the data plane VNF nodes (e.g., SGW VNF, PGW-U VNF) into sub-components so that these components can get scaled independently and efficiently. Thus, we believe that some designs such as ScaleBricks [161] can be used as references to improve the scalability in the data plane.

2) Reliability and High Availability: From an SDN perspective, shifting the control plane of network devices into a centralized controller may have a high impact on the reliability of the control plane because it is a potential single point of failure. Since this central controller is in charge of the whole network, the whole network may collapse, if it fails. In order to address this issue, a proper hot-standby design for the SDN controllers and related recovery procedures between them need to be carefully designed. Such a redundant design of the SDN controller would also improve the availability of services in the network. Although several SDN and NFV based MPC proposals have mentioned multiple controller designs in the control plane, the main purpose is to improve the scalability, not the reliability and to obtain high availability. Therefore, this topic needs to be investigated in detail.

When it comes to NFV, dealing with the reliability and availability is to deal with the session state and user context inside VNF nodes (e.g., MME, SGW, PGW). Indeed, these VNF nodes are currently stateful nodes which are associated with many internal states and contexts of the UE. While scaling this kind of VNF nodes, especially when removing a VNF node (i.e., scaling in), the associated states and UE contexts would be lost, thus resulting in a significant impact on the session continuity. As a consequence, it affects the reliability and availability of services in the network. As shown in Table IV, some solutions have been proposed. These include storing the states and contexts in an external database (DB) such as in the three-tier MME architectures ([143], [145], [146]) or replicating them across the network such as in SCALE [144], Kaippallimalil *et al.* [170], and Cau *et al.* [171]. The former scheme has higher reliability but it results in a long delay for acquiring the states. The latter scheme has higher availability of services but it results in the synchronization challenge between these states. Therefore, it is important to decide when to separate the states from the data session, or when to replicate these states and how many replicas are needed. One suggestion for the future study could be the migration of the states among VNF nodes by adopting some state migration schemes, which have been done in the virtual middlebox networking field such as Split/Merge [172], and OpenNF [173]. Another suggestion for improving and evaluating the reliability of NFV deployments in the context of mobile broadband networks can be found in [174].

C. Placement Optimization and Resource Allocation

As described in Section VIII, by adopting SDN and NFV technologies, both the control plane and data plane functions of the MPC can be deployed in either a centralized or distributed fashion. However, the placement of these functions would significantly impact the entire network performance if the placement strategy is not carefully planned.

From an SDN perspective, this placement problem is referred to as the Controller Placement Problem (CPP), which is about finding the optimal number of SDN controllers and their proper locations in order to minimize the propagation delay between the controller and forwarding elements or to maximize fault tolerance, etc. Although the CPP problem has been discussed in many SDN research works such as [175] and [176], more work needs to be done when it comes into the mobile network, in particular the MPC network. The reason is that the MPC network may have different and specific constraints while modeling the CPP problem compared to the wired SDN network such as mobility of UEs [177]. Therefore, the CPP problem in the SDN/NFV based MPC network demands more investigations.

TABLE V
A SUMMARY OF CURRENT RESEARCH ON PLACEMENT OPTIMIZATION IN SDN/NFV BASED MPC ARCHITECTURE

References	Approach	VNF Types	Formulation	Algorithm	Main Objectives
Taleb et al. [162]	VNFP	SGW	ILP	Heuristic	Minimizing the relocation frequency and the number of SGWs needed to deploy in a carrier cloud
Bagaa et al. [163]	VNFP	PGW	NLO	Heuristic	Reducing the operator costs with the optimal number of PGWs and ensuring high QoS/QoE for users
Kiess et al. [164]	VNFP	SGW, PGW	N/A	N/A	Comparing the deployment cost of the centralized and the distributed gateway scenarios in a country-scale network
Basta et al. [165]	VNFP	SGW, PGW	MILP	Heuristic	Minimizing the total network load under a certain data plane delay budget in a SDN and NFV environment
Taleb et al. [166]	VNFP	SGW, PGW	ILP	Nash Theory	Minimizing the data path between users and their respective PGWs and optimizing their sessions' mobility
Yousaf et al. [147]	VNFP	SGW, PGW MME	N/A	Heuristic	Analyzing and comparing the deployment costs of two constraint-based heuristic approaches for deploying EPC VNFs
Marotta et al. [167]	VNFP	SGW, PGW MME	RO	Exact	Minimizing the power consumption of the servers and switches needed to deploy all the required VNFs with uncertain inputs
Baumgartner et al. [168]	VNE	SGW, PGW MME, HSS	ILP	Exact	Minimizing the cost of link and node resources in the physical substrate network when a given traffic demand is accommodated
Baumgartner et al. [169]	VNE	SGW, PGW MME, HSS	MILP	Exact	Extending the model in [169] with the consideration of user and control plane latency bounds

ILP: Integer Linear Programing; MILP: Mixed ILP; NLO: Non-Linear Optimization; RO: Robust Optimization; N/A: Not Applicable

From NFV perspective, the placement problem refers to the placement of VNF nodes over a NFV-based network infrastructure, which is so-called VNF placement (VNFP) problem. The VNFP problem in the MPC network has recently attracted some attention in the literature with the goal of finding the optimal location for a single type of VNF such as in [162] and [163] or a set of different VNFs such as in [147] and [164]–[167], while guaranteeing different performance constraints ranging from minimizing the network load, resources, and power consumption to ensuring the users' QoE and QoS. Another variant of placement optimization called Virtual Network Embedding (VNE) problem [178], in which the virtual network topologies and resource demands are mostly static [179], has also been considered by Baumgartner *et al.* [168], [169]. In these works, the authors aim at finding an optimal embedding strategy for virtual links and nodes of a requested virtual network onto a given physical substrate network.

As shown in Table V, most of the optimization problems are formulated as Integer Linear Programming (ILP) or Mixed ILP (MILP) models and then are mostly solved by using some well-known heuristic algorithms such as greedy [162], [163]. Robust Optimization (RO) is also being discussed very recently by Marotta and Kassler [167], which refers to the optimization problem with uncertain input parameters. This RO approach is considered as one of our suggestions for future study on the topic of placement optimization. Since the presented optimization solutions have considered some set of constraints such as mobility-aware, QoE-aware, or latency-aware, it could be interesting to see if we could combine these constraints and to formulate a multi-objective optimization model. In addition, the placement of an SDN controller and VNF nodes are currently being solved separately, thus jointly considering these two placement problems could be a future research topic. Moreover, we believe that the advent of the network slicing technology, where each slice is created with a set of required network functions according to different service requirements, will create more possible ways of placing network functions. Last but not least, placing the VNFs over the physical infrastructure would result in a situation where multiple VNFs reside on the same physical machine and share the same resources. Therefore, having efficient scheduling techniques to schedule the resources among VNFs become crucial, and we believe that more work needs to be done in the future to achieve this goal.

D. Management and Orchestration

The transformation of the legacy network functions from hardware-centric to software-centric, driven by SDN and NFV technologies, demand changes in the current network management systems. For example, decoupling the network functions from the dedicated hardware to run as VMs in a general-purpose server results in a new set of management functions focused on the VNF lifecycle management such as instantiation, modification, or termination of the VNF. Understanding the need of this demand, ETSI NFV working group has proposed a MANO framework [45] (see Section II-C), which covers the orchestration and lifecyle management of all infrastructure resources, the lifecycle management of VNFs and Network Services (NS). Since then, several open source platforms have been released to provide the practical implementation of the NFV MANO framework (Table VI). For example, Open Source MANO (OSM) [185] is recently launched by ETSI NFV working group with the objective of providing a reference framework that implements MANO functionalities by integrating three other open source platforms (OpenMANO [181], RIFT.ware [187], and JUJU [182]) into a single platform. It should be noted that, although most of the MANO projects listed in Table VI do not have their own VIM implementations, their MANO platforms are still able to support integration with different

TABLE VI
A SUMMARY OF CURRENT MANO OPEN SOURCE PLATFORMS AND PROJECTS

Name	Institute/Company	MANO Scope				Programming Language	First Release	Current Version
		VIM	VNFM	NFVO	SO			
OPNFV [180]	Linux Foundation	✓	–	–	–	Python	2015	Colorado v3.0
OpenMANO [181]	Telefonica	✓	–	✓	–	Python	2015	Release 0.4
JUJU [182]	Canonical	–	✓	–	–	Go, Python	2015	Release 2.0
OpenBaton [55]	FOKUS	–	✓	✓	–	Java, Python	2015	Release 3.0
OpenStack Tacker [54]	OpenStack	–	✓	✓	–	Python	2015	OpenStack Ocata
Cloudify [183]	GigaSpace	–	✓	✓	–	Python	2014	Release 3.4
Open-O [184]	Linux Foundation	–	✓	✓	✓	Java	2016	Release 1.0
OSM [185]	ETSI NFV	–	✓	✓	✓	Python	2016	Release 1.0
ECOMP [186]	AT&T	–	✓	✓	✓	N/A	2016	N/A

VIM = Virtualized Infrastructure Manager, VNFM = VNF Manager, NFVO = NFV Orchestrator, SO = Service Orchestrator

kinds of VIM (e.g., OpenStack [53], VMWare, etc.). However, OpenStack [53] is recognized as the most common one. As shown in Table VI some of the projects including the Open-O project [184], the OSM project [185], and the AT&T's ECOMP project [186], introduce a component called Service Orchestrator (SO), which is responsible for service orchestration implemented on top of the NFVO in the ETSI NFV framework [45]. Currently, the OPNFV project [180] is expanding its original scope (i.e., only VIM) to include MANO. It should be noted that not long ago Open-O and ECOMP has been merged to create a new platform called Open Network Automation Platform (ONAP).

However, the current focus of the MANO framework is on aspects of management and orchestration that are specific to NFV [45] with less consideration of the management and orchestration of SDN resources such as SDN controllers, SDN network infrastructure (i.e., links and forwarding elements) to interconnect VNFs. Although SDN and NFV are two different technologies, they are complementary. For example, a command from the MANO to destroy or create a VNF would trigger changes in the number of links and forwarding elements in the network infrastructure. Although there are some efforts being developed by the Open-O project [184], the 5G NORMA project [188], [189], and Verizon [190], there is a clear need for more research and implementation efforts in this topic to have a complete management solution to manage and orchestrate both SDN and NFV resources. In addition, as discussed in Section IX-C, optimal placement of the network functions over the network infrastructure can be solved by several optimization algorithms in a finite amount of time. Implementing these algorithms in a placement engine as a part of one of the listed MANO open source platforms, along side with the implementation of innovative algorithms for migrating VNFs across the infrastructure, could be an interesting future research challenge. Furthermore, improving the decision making of MANO based on policy-aware [191] or QoE-aware [192] is also a future research consideration.

Other challenges of MANO may come from hybrid deployment of network functions where not only VNFs but also physical network functions (PNFs) are managed and orchestrated as discussed in [193]. Moreover, since the VNFs can be deployed in distributed data centers spreading multiple domains, it requires the consistency of configuration or synchronization between domains [194], [195]. Also, the new emerging network slicing concept also imposes challenges to the management and orchestration. It is referred to as an end-to-end network slice orchestrator or manager [23] which covers all aspects of network slicing such as lifecyle management of a slice, assigning nodes in a slice, service provisioning, etc. These hybrid, inter-domain, and slicing MANO issues need to be further investigated. Other MANO-related issues are also considered as important research directions such as programmability and interoperability [196], interworking with the existing OSS/BSS systems [197], and automatic and real-time orchestration [198].

E. Network Sharing and Slicing

In the past, scenarios to share the network infrastructure between the Mobile Network Operators (MNOs) such as active or passive sharing are typically used to reduce their OPEX and CAPEX [202], [203], thus resulting in the development of Mobile Virtual Network Operators (MVNOs), which rely on the infrastructure and most other things provided by the MNO. The network infrastructure sharing has now changed when the mobile network moves into the software-based platforms driven by SDN and NFV technologies. Indeed, implementing the mobile network functions as software brings up the notion of multi-tenancy in which multiple VNFs can be configured on the same NFV infrastructure [113] and each MVNO can be a tenant owning an isolated set of interconnected VNFs. Recently, the network sharing paradigm has evolved into a "network slicing" [20], [22], where the network infrastructure is shared to support particular communication services not only phone-to-phone communications but other emerging communication services such as autonomous cars, massive IoT, etc. In this situation, the network functions and applications can be provided through the notion of function store or application store [204] according to different use cases. The term "network slice" has been generalized as the main construct of the 5G network services and several efforts have been made to illustrate the use of the network slicing and its benefits [205], [206]. Regarding SDN, the network slicing concept has been discussed in several proposals in the context of network hypervisors [148], such as FlowVisor [149],

TABLE VII
A SUMMARY OF CURRENT EPC OPEN SOURCE PLATFORMS

Name	Institute/Author	Software Type	Main Features	Supporting		Deployment Capability			Programming Language	Current Version
				SDN	NFV	Real	Emul.	Simul.		
OpenEPC [139]	CND	License	EPC Rel. 12	Yes	Yes	✓	✓	–	C	Rel.6
Open5GCore [199]	FOKUS	License	EPC Rel. 12	Yes	Yes	✓	✓	–	C	Rel.2
OpenAirInterface [140]	EURECOM	Open source	EPC Rel. 10	Soon	Yes	✓	✓	✓	C	v0.4.0
nwEPC [200]	Amit Chawre	Open source	EPC S/PGW	No	Yes	✓	✓	–	C	N/A
NS-3 LENA [201]	NS3	Open source	N/A	No	No	–	–	✓	C++	N/A
SDN-EPC [106]	IIT Bombay	Open source	N/A	Yes	Yes	–	–	✓	C++	1.0
NFV-EPC [106]	IIT Bombay	Open source	N/A	No	Yes	–	–	✓	C++	1.1

OpenVirteX [150]. Most of these network hypervisors focus on the slicing in fixed and wired SDN network. Inspired from the FlowVisor, Nguyen and Kim [207] proposed MobileVisor, which slices the MPC network infrastructure into virtual networks owned by different MVNOs.

However, many challenges and issues still need to be addressed in future work such as slice formation, dynamic slicing, end-to-end slice provisioning. For example, although several types of slices have been outlined in [2], it still remains an issue how to classify these types. One suggestion could be to use the outputs of service classification frameworks based on a machine learning approach, which is being investigated by the FANTASTIC-5G project [208] as inputs of a slice creator or slice orchestrator. In addition, the problems related to placement of network functions within a slice, slicing orchestration, or inter-domain services slicing also need to be further studied to achieve the effectiveness of network slicing. Another possible research direction in this topic could be related to slice resource allocation such as inter-slice resource optimization, resource movement from slice to slice, or slice resource scaling, etc. Last but not least, guaranteeing the isolation between slices and security are also important factors when realizing the network sharing and slicing in the practical implementation.

F. Network Performance, Evaluation and Benchmarks

In this section, we discuss some challenges coming from the high performance demands of the network functions available in the mobile environment and the lack of performance evaluation methodology and benchmarking tool.

Moving from the network functions running on customized and specialized hardware platforms into VMs running in the general-purpose servers, will impact the network performance. Indeed, currently the VNFs mainly execute inside VMs on top of some hypervisor such as KVM or XEN so a data packet has to traverse many layers before reaching the application on the user space, thus resulting in a high latency. This latency overhead contributes to the degradation of the network performance and would not meet the carrier-class, high-performance requirements of mobile and telecommunication networks. One possible solution is to leverage new emerging virtualization technologies such as Docker [209]. In contrast to the hypervisor technologies, Docker allows containers, which run applications or network functions to share the same host operating system (OS) kernel instead of running their own OS. By using this lightweight virtualization technology, Docker containers provide better performance than equivalent hypervisor-based VMs running the same software or network functions [210], [211]. Fontenla-González *et al.* [212] have illustrated the benefit of using containers over VMs in virtualizing EPC network functions in terms of memory efficiency. Another possible solution to improve the network performance is to adopt highly efficient packet processing frameworks into the data plane such as OpenFlow-enabled OpenvSwitch [213] with Intel DPDK libraries and drivers [141], Linux New API (NAPI) [214], NetVM and OpenNetVM [215], [216].

While talking about the performance evaluation aspect, few efforts have been made to evaluate and assess the performance of the SDN- and NFV-based EPC network and its components. Lange *et al.* [217] provided a performance comparison between NAPI-based SGW VNF and DPDK-based SGW VNF implementations. A comparison of SDN-based and NFV-based EPC gateways is conducted in [218]. Kurtz *et al.* [219] have conducted an experiment to examine the performance between bare-metal and virtualized deployment of an SDN-based 5G. An experiment to assert the performance of the entire EPC network on general-purpose servers has been done by Hirschman *et al.* [220]. Through the survey classification section, many of the proposed designs have not been evaluated in sufficient detail. Some have very simple modeling analysis of signaling evaluations such as Chourasia and Sivalingam [85], Sama *et al.* [97], Nguyen and Kim [98], [99], and Hasegawa and Murata [134] while several proposals have run some simple experiments on a license-based platform called OpenEPC [139], such as Mueller *et al.* [103], Medhat *et al.* [113], Fontenla-González *et al.* [212]. We realize that the root of the problem is the shortage of benchmarking tools. As listed in Table VII, there are currently only a few platforms for evaluating and benchmarking and not all of them are fully open source such as OpenEPC [139] and Open5GCore [199]. We believe that developing experimentation tools and simulation models for this topic can substantially improve our understanding of these solutions and accelerate the rate of innovation in this area. In addition, evaluating and testing the performance of a system cannot be achieved without having a detailed evaluation methodology, which identifies testing scenarios and performance metrics. As regards, the NFV performance and portability best practices, and NFV pre-deployment testing specifications created by ETSI NFV in [221] and [222], respectively, are good starting points to look at.

X. Conclusion

In the mobile network, the MPC plays a crucial role to bridge the gap between the mobile users and devices which are managed by the mobile network operators, and the packet data network. The MPC is currently under pressure of accommodating the rapidly increasing number of mobile devices, new types of services and applications, and meeting the new requirements introduced in the 5G era. Due to many drawbacks of the current MPC network in terms of the control and data coupling, and costly hardware-based network functions, it is economically infeasible to accommodate these new devices and services, and to meet 5G requirements without radically changing the architecture. To this end, SDN and NFV technologies are considered as key drivers in the development of the MPC, paving the way towards the 5G era.

In this paper, we present a comprehensive architectural survey of state-of-the art works leveraging SDN and NFV into the current MPC network architecture. We first describe the benefits these two technologies offer to mobile network operators as well as typical ways how they can transform their MPC network architectures into a new softwarized and virtualized ecosystem. Besides, we also include some research activities from standardization groups, industry-related efforts, and on-going global projects on this topic during surveying. To provide readers a complete view on this topic, we approach the existing research proposals from different angles, which serve as dimensions of our survey taxonomy, including the architectural approach they follow, the technology they adopt, the functional implementation they select, and the deployment strategy they prescribe. For each group, in each survey dimension, we review the most relevant research works by highlighting their main contributions and characteristics. We then compare all these works according to the criteria defined in the four-dimensional taxonomy. In addition, we extend the comparison tables with some new comparison attributes including the type of mechanism to route the user data packets, the type of southbound interfaces used, the backward compatibility, the capability to support network slicing, and the scalability of the proposed architectures.

Based on the exhaustive comparison and classification of contemporary work in both text description and table representation, we can draw many lessons, which can then pose many new challenges and open issues that need to be further addressed in order to achieve a complete solution on a SDN and NFV based MPC network architecture. Firstly, there is no-one-solution-fits-all in using SDN and NFV to redesign the MPC network architecture. The combined SDN and NFV based MPC approaches are currently becoming a mainstream. Main challenge of these approaches is the management and orchestration of the heterogeneity of network resources (i.e., SDN resources such as SDN controllers, and NFV resources such as VNFs). Thus, it is necessary to have a tool that can efficiently manage and orchestrate network resources in such heterogeneous environment. Finding optimal placement and efficient resource allocation of network functions in such environment would also be an emerging research field. Secondly, most of the surveyed works rely on the use of OpenFlow protocol and its extensions as the southbound interface between the control and user planes without detailed design. Therefore, the complete and standardized design of southbound as well as northbound interfaces are needed. Thirdly, the scalability of the control plane has been addressed in some proposals by using hierarchical design or splitting up the network function into multiple subfunctions. However, the scalability of the user plane have not been considered. In addition, the performance of user-plane network functions when moved to the virtualized environment needs to be carefully examined. Fourthly, the network slicing becomes a more and more important research topic since it is one of the key features of 5G. However, there are many challenges that need to be addressed such as management and orchestration of slices, resource allocation for a slice, isolation between slices, etc. Last but not least, the lack of evaluation tools to evaluate and assess the performance of the SDN- and NFV-based MPC network and its components is also challenging. We believe that these six challenges are important to be further studied in the future work.

References

[1] Cisco. (2016). *Mobile VNI Forecast Highlights*. Accessed on Jul. 6, 2016. [Online]. Available: http://www.cisco.com/assets/sol/sp/vni/forecast_highlights_mobile/index.html

[2] NGMN. (2015). *NGMN 5G White Paper*. Accessed on Jul. 6, 2016. [Online]. Available: https://www.ngmn.org/uploads/media/NGMN_5G_White_Paper_V1_0.pdf

[3] 3GPP. (2011). *3GPP TS 23.401 Technical Specification Group Services and System Aspects; General Packet Radio Service (GPRS) Enhancements for Evolved Universal Terrestrial Radio Access Network (E-UTRAN) Access, (Release 10)*. Accessed on Jul. 6, 2016. [Online]. Available: http://www.3gpp.org/dynareport/23401.htm

[4] J. Qadir, N. Ahmed, and N. Ahad, "Building programmable wireless networks: An architectural survey," *EURASIP J. Wireless Commun. Netw.*, vol. 2014, no. 1, p. 172, 2014.

[5] M. Yang *et al.*, "Software-defined and virtualized future mobile and wireless networks: A survey," *Mobile Netw. Appl.*, vol. 20, no. 1, pp. 4–18, 2015.

[6] N. A. Jagadeesan and B. Krishnamachari, "Software-defined networking paradigms in wireless networks: A survey," *ACM Comput. Surveys (CSUR)*, vol. 47, no. 2, pp. 1–11, 2015.

[7] I. F. Akyildiz, S.-C. Lin, and P. Wang, "Wireless software-defined networks (W-SDNs) and network function virtualization (NFV) for 5G cellular systems: An overview and qualitative evaluation," *Comput. Netw.*, vol. 93, pp. 66–79, Dec. 2015.

[8] N. Bizanis and F. A. Kuipers, "SDN and virtualization solutions for the Internet of Things: A survey," *IEEE Access*, vol. 4, pp. 5591–5606, 2016.

[9] I. T. Haque and N. Abu-Ghazaleh, "Wireless software defined networking: A survey and taxonomy," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 4, pp. 2713–2737, 4th Quart., 2016.

[10] S. Tomovic, M. Pejanovic-Djurisic, and I. Radusinovic, "SDN based mobile networks: Concepts and benefits," *Wireless Pers. Commun.*, vol. 78, no. 3, pp. 1629–1644, 2014.

[11] T. Chen, M. Mantinmikko, X. Chen, X. Zhou, and P. Ahokangas, "Software defined mobile networks: Concept, survey, and research directions," *IEEE Commun. Mag.*, vol. 53, no. 11, pp. 126–133, Nov. 2015.

[12] V.-G. Nguyen, T.-X. Do, and Y. Kim, "SDN and virtualization-based LTE mobile network architectures: A comprehensive survey," *Wireless Pers. Commun.*, vol. 86, no. 3, pp. 1401–1438, 2016.

[13] Open Networking Foundation. (2012). *Software-Defined Networking: The New Norm for Networks*. Accessed on Jul. 6, 2016. [Online]. Available: https://www.opennetworking.org/images/stories/downloads/sdn-resources/white-papers/wp-sdn-newnorm.pdf

[14] ETSI NFV ISG. (2012). *Network Functions Virtualization, White Paper*. Accessed on Jul. 6, 2016. [Online]. Available: https://portal.etsi.org/ nfv/nfv_white_paper.pdf

[15] B. Naudts, M. Kind, S. Verbrugge, D. Colle, and M. Pickavet, "How can a mobile service provider reduce costs with software-defined networking?" *Int. J. Netw. Manag.*, vol. 26, no. 1, pp. 56–72, 2016.

[16] E. Hernandez-Valencia, S. Izzo, and B. Polonsky, "How will NFV/SDN transform service provider opex?" *IEEE Netw.*, vol. 29, no. 3, pp. 60–67, May/Jun. 2015.

[17] P. K. Agyapong, M. Iwamura, D. Staehle, W. Kiess, and A. Benjebbour, "Design considerations for a 5G network architecture," *IEEE Commun. Mag.*, vol. 52, no. 11, pp. 65–75, Nov. 2014.

[18] I. F. Akyildiz, S. Nie, S.-C. Lin, and M. Chandrasekaran, "5G roadmap: 10 key enabling technologies," *Comput. Netw.*, vol. 106, pp. 17–48, Sep. 2016.

[19] A. Manzalini *et al.*, "Towards 5G software-defined ecosystems: Technical challenges, business sustainability and policy issues," White Paper, 2016. [Online]. Available: http://sdn.ieee.org/images/files/pdf/towards-5g-software-defined-ecosystems.pdf

[20] IMT-2020 Promotion Group. (2016). *5G Network Architecture Design*. Accessed on Jul. 6, 2016. [Online]. Available: http://www.catr.cn/kxyj/qwfb/bps/201606/P020160624527039498728.pdf

[21] S. Abdelwahab, B. Hamdaoui, M. Guizani, and T. Znati, "Network function virtualization in 5G," *IEEE Commun. Mag.*, vol. 54, no. 4, pp. 84–91, Apr. 2016.

[22] Open Networking Foundation. (2016). *Tr-526: Applying SDN Architecture to 5G Slicing*. Accessed on Jul. 6, 2016. [Online]. Available: https://www.opennetworking.org/images/stories/downloads/sdn-resources/technical-reports/Applying_SDN_Architecture_to_5G_Slicing_TR-526.pdf

[23] X. Zhou, R. Li, T. Chen, and H. Zhang, "Network slicing as a service: Enabling enterprises' own software-defined cellular networks," *IEEE Commun. Mag.*, vol. 54, no. 7, pp. 146–153, Jul. 2016.

[24] *Open Networking Foundation*. Accessed on Jul. 6, 2016. [Online]. Available: https://www.opennetworking.org

[25] *ETSI NFV ISG*. Accessed on Jul. 6, 2016. [Online]. Available: http://www.etsi.org/technologies-clusters/technologies/nfv

[26] 3GPP. (2016). *3GPP TR 23.799, Study on Architecture for Next Generation System, Release 14, Version 7.0*. Accessed on Jul. 6, 2016. [Online]. Available: http://www.3gpp.org/DynaReport/23799.htm

[27] 3GPP. (2016). *3GPP TR 23.714, Study on Control and User Plane Separation of EPC Nodes, Version 14.0.0.* Accessed on Jul. 6, 2016. [Online]. Available: http://www.3gpp.org/DynaReport/23714.htm

[28] 3GPP. (2015). *Study on Network Management of Virtualized Networks*. Accessed on Jul. 6, 2016. [Online]. Available: http://www.3gpp.org/DynaReport/23714.htm

[29] M. Olsson, S. Rommer, C. Mulligan, S. Sultana, and L. Frid, *SAE and the Evolved Packet Core: Driving the Mobile Broadband Revolution*. Boston, MA, USA: Academic Press, 2009.

[30] S. Sesia, I. Toufik, and M. Baker, *LTE-The UMTS Long Term Evolution*. Chichester, U.K.: Wiley, 2015.

[31] Open Networking Foundation. (2013). *SDN Architecture Overview*. Accessed on Jul. 6, 2016. [Online]. Available: https://www.opennetworking.org/images/stories/downloads/sdn-resources/technical-reports/SDN-architecture-overview-1.0.pdf

[32] N. McKeown *et al.*, "OpenFlow: Enabling innovation in campus networks," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 2, pp. 69–74, 2008.

[33] A. Doria *et al.*, "Forwarding and control element separation (ForCES) protocol specification," Internet Eng. Task Force, Fremont, CA, USA, RFC 5810, 2010.

[34] *ON.Lab*. Accessed on Jul. 6, 2016. [Online]. Available: http://onosproject.org/

[35] *Linux Foundatation*. Accessed on Jul. 6, 2016. [Online]. Available: https://www.opendaylight.org/

[36] E. Haleplidis *et al.*, "Software-defined networking (SDN): Layers and architecture terminology," Internet Eng. Task Force, Fremont, CA, USA, RFC 7426, 2015.

[37] B. A. A. Nunes, M. Mendonca, X.-N. Nguyen, K. Obraczka, and T. Turletti, "A survey of software-defined networking: Past, present, and future of programmable networks," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 3, pp. 1617–1634, 3rd Quart., 2014.

[38] D. Kreutz *et al.*, "Software-defined networking: A comprehensive survey," *Proc. IEEE*, vol. 103, no. 1, pp. 14–76, Jan. 2015.

[39] Y. Jarraya, T. Madi, and M. Debbabi, "A survey and a layered taxonomy of software-defined networking," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 4, pp. 1955–1980, 4th Quart., 2014.

[40] W. Xia, Y. Wen, C. H. Foh, D. Niyato, and H. Xie, "A survey on software-defined networking," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 1, pp. 27–51, 1st Quart., 2015.

[41] H. Farhady, H. Lee, and A. Nakao, "Software-defined networking: A survey," *Comput. Netw.*, vol. 81, pp. 79–95, Apr. 2015.

[42] *Open Networking Foundation WMWG*. Accessed on Jul. 6, 2016. [Online]. Available: https://www.opennetworking.org/technical-communities/areas/specification

[43] R. Mijumbi *et al.*, "Network function virtualization: State-of-the-art and research challenges," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 1, pp. 236–262, 1st Quart., 2016.

[44] ETS NFV ISG. (2014). *Network Function Virtualization; Virtual Network Function Architecture*. Accessed on Jul. 6, 2016. [Online]. Available: http://www.etsi.org/deliver/etsi_gs/NFV-SWA/001_099/001/01.01.01_60/gs_nfv-swa001v010101p.pdf

[45] ETS NFV ISG. (2014). *Network Function Virtualization; Management and Orchestration*. Accessed on Jul. 6, 2016. [Online]. Available: http://www.etsi.org/deliver/etsi_gs/NFV-MAN/001_099/001/01.01.01_60/gs_nfv-man001v010101p.pdf

[46] ETS NFV ISG. (2013). *Network Function Virtualization; Use Cases*. Accessed on Jul. 6, 2016. [Online]. Available: http://www.etsi.org/deliver/etsi_gs/nfv/001_099/001/01.01.01_60/gs_nfv001v010101p.pdf

[47] ETS NFV ISG. (2015). *Network Function Virtualization; Ecosystem; Report on SDN Usage in NFV Architectural Framework*. Accessed on Jul. 6, 2016. [Online]. Available: http://www.etsi.org/deliver/etsi_gs/NFV-EVE/001_099/005/01.01.01_60/gs_nfv-eve005v010101p.pdf

[48] Open Networking Foundation. (2015). *Tr-518 Relationship of SDN and NFV*. Accessed on Jul. 6, 2016. [Online]. Available: https://www.opennetworking.org/images/stories/downloads/sdn-resources/technical-reports/onf2015.310_Architectural_comparison.08-2.pdf

[49] Y. Li and M. Chen, "Software-defined network function virtualization: A survey," *IEEE Access*, vol. 3, pp. 2542–2553, 2015.

[50] J. Halpern and C. Pignataro, "Service function chaining (SFC) architecture," Internet Eng. Task Force, Fremont, CA, USA, RFC 7665, 2015.

[51] A. M. Medhat *et al.*, "Service function chaining in next generation networks: State of the art and research challenges," *IEEE Commun. Mag.*, vol. 55, no. 2, pp. 216–223, Feb. 2017.

[52] R. Bifulco, A. Matsiuk, and A. Silvestro, "Ready-to-deploy service function chaining for mobile networks," in *Proc. 2nd IEEE Conf. Netw. Softwarization (NetSoft)*, Seoul, South Korea, 2016, pp. 175–183.

[53] *OpenStack*. Accessed on Jul. 6, 2016. [Online]. Available: https://openstack.org/

[54] *OpenStack*. Accessed on Jul. 6, 2016. [Online]. Available: https://wiki.openstack.org/wiki/Tacker

[55] *Fraunhofer FOKUS Research Institute*. Accessed on Jul. 6, 2016. [Online]. Available: http://openbaton.github.io/

[56] *NEC Corporation*. Accessed on Jul. 6, 2016. [Online]. Available: http://www.nec.com/en/press/201310/global_20131022_03.html

[57] *Ericsson*. Accessed on Jul. 6, 2016. [Online]. Available: https://www.ericsson.com/news/1761217

[58] Cisco. (2016). *Cisco Virtualized Packet Core*. Accessed on Jul. 6, 2016. [Online]. Available: http://www.cisco.com/c/en/us/solutions/collateral/service-provider/lte-epc/datasheet-c78-730493.html?cachemode=refresh

[59] Nokia. (2016). *Nokia Virtualized EPC: Delivering on the Promise of NFV and SDN*. Accessed on Jul. 6, 2016. [Online]. Available: http://resources.alcatel-lucent.com/asset/182576

[60] *NTT DOCOMO*. Accessed on Jul. 6, 2016. [Online]. Available: https://www.nttdocomo.co.jp/english/info/media_center/pr/2016/0219_00.html

[61] ETS MEC ISG. (2014). *Mobile Edge Computing White Paper*. Accessed on Jul. 6, 2016. [Online]. Available: https://portal.etsi.org/portals/0/tbpages/mec/docs/mobile-edge_computing_-_introductory_technical_white_paper_v1%2018-09-14.pdf

[62] B. J. van Asten, N. L. van Adrichem, and F. A. Kuipers, "Scalability and resilience of software-defined networking: An overview," arXiv, Preprint arXiv: 1408.6760, 2014.

[63] O. Blial, M. B. Mamoun, and R. Benaini, "An overview on SDN architectures with multiple controllers," *J. Comput. Netw. Commun.*, vol. 2016, Apr. 2016, Art. no. 9396525, doi: 10.1155/2016/9396525.

[64] J. Kempf, B. Johansson, S. Pettersson, H. Lüning, and T. Nilsson, "Moving the mobile evolved packet core to the cloud," in *Proc. IEEE 8th Int. Conf. Wireless Mobile Comput. Netw. Commun. (WiMob)*, Barcelona, Spain, 2012, pp. 784–791.

[65] J. Kempf, B. E. Johansson, S. Pettersson, T. K. Nilsson, and H. Lüning, "Implementing EPC in a cloud computer with openflow data plane," U.S. Patent 8 867 361, Oct. 2014.

[66] S. Gebert *et al.*, "Demonstrating the optimal placement of virtualized cellular network functions in case of large crowd events," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 4, pp. 359–360, 2014.

[67] *ETSI NFV ISG*. Accessed on Jul. 6, 2016. [Online]. Available: http://nfvwiki.etsi.org/index.php?title=PoCs_Overview

[68] NFV ISG PoC. (2015). *SDN Enabled Virtual EPC Gateway*. Accessed on Jul. 6, 2016. [Online]. Available: http://nfvwiki.etsi.org/images/NFVTST(15)000006_NFV_ISG_PoC_Proposal_SDN_Enabled_EPC_Gwy_r2_was_PER114.pdf

[69] Hewlett Packard Enterprise. (2016). *SDN-Enabled Mobile Packet Core*. Accessed on Jul. 6, 2016. [Online]. Available: https://www.hpe.com/h20195/v2/GetPDF.aspx/4AA6-3419ENN.pdf

[70] T. Quanjun, "Introducing SDN technology into a mobile core network," *ZTE Technol.*, vol. 17, no. 2, pp. 22–24, 2015.

[71] K.-W. Lee. (2016). *Building A Next Generation Mobile Network Using Open Technologies*. Accessed on Jul. 6, 2016. [Online]. Available: http://events.linuxfoundation.org/sites/events/files/slides/ons_keynote_wed_kwonlee.pdf

[72] R. Guerzoni, R. Trivisonno, and D. Soldani, "SDN-based architecture and procedures for 5G networks," in *Proc. 1st Int. Conf. 5G Ubiquitous Connectivity (5GU)*, 2014, pp. 209–214.

[73] R. Trivisonno, R. Guerzoni, I. Vaishnavi, and D. Soldani, "SDN-based 5G mobile networks: Architecture, functions, procedures and backward compatibility," *Trans. Emerg. Telecommun. Technol.*, vol. 26, no. 1, pp. 82–92, 2015.

[74] R. Trivisonno, R. Guerzoni, I. Vaishnavi, and D. Soldani, "Towards zero latency software defined 5G networks," in *Proc. IEEE Int. Conf. Commun. Workshop (ICCW)*, London, U.K., 2015, pp. 2566–2571.

[75] R. Trivisonno. (2016). *CONFIG: Convergent Core Architecture for Next Generation Networks*. Accessed on Jul. 6, 2016. [Online]. Available: http://www.5g-control-plane.eu/wp-content/uploads/2016/06/CONFIG-Brussels-Presentation-April-6th-v3.pdf

[76] I. Vaishnavi *et al.*, "A prelude to the 5G core network architecture," in *Proc. IEEE Glob. Commun. Conf. Interact. Demo (GLOBECOM)*, San Diego, CA, USA, 2015, pp. 1–5.

[77] Cisco. (2010). *Top 10 Considerations for a Successful 4G LTE Evolved Packet Core (EPC) Deployment*. Accessed on Jul. 6, 2016. [Online]. Available: http://www.cisco.com/c/en/us/solutions/collateral/service-provider/lte-epc/white-paper-c11-730105.pdf

[78] Z. Savic. (2011). *LTE Design and Deployment Strategies*. Accessed on Jul. 6, 2016. [Online]. Available: http://www.cisco.com/web/ME/expo2011/saudiarabia/pdfs/LTE_Design_and_Deployment_Strategies-Zeljko_Savic.pdf

[79] L. E. Li, Z. M. Mao, and J. Rexford, "Toward software-defined cellular networks," in *Proc. IEEE Eur. Workshop Softw. Defined Netw.*, Darmstadt, Germany, 2012, pp. 7–12.

[80] X. Jin, L. E. Li, L. Vanbever, and J. Rexford, "SoftCell: Scalable and flexible cellular core network architecture," in *Proc. 9th ACM Conf. Emerg. Netw. Exp. Technol. (CoNEXT)*, Santa Barbara, CA, USA, 2013, pp. 163–174.

[81] M. Moradi, W. Wu, L. E. Li, and Z. M. Mao, "SoftMoW: Recursive and reconfigurable cellular WAN architecture," in *Proc. 10th ACM Int. Conf. Emerg. Netw. Exp. Technol. (CoNEXT)*, Sydney, NSW, Australia, 2014, pp. 377–390.

[82] M. Moradi, L. E. Li, and Z. M. Mao, "SoftMoW: A dynamic and scalable software defined architecture for cellular WANs," in *Proc. 3rd ACM Workshop Hot Topics Softw. Defined Netw. (HotSDN)*, Santa Clara, CA, USA, 2014, pp. 201–202.

[83] V. Yazıcı, U. C. Kozat, and M. O. Sunay, "A new control plane for 5G network architecture with a case study on unified handoff, mobility, and routing management," *IEEE Commun. Mag.*, vol. 52, no. 11, pp. 76–85, Nov. 2014.

[84] H. Lindholm, L. Osmani, H. Flinck, S. Tarkoma, and A. Rao, "State space analysis to refactor the mobile core," in *Proc. 5th Workshop All Things Cellular Oper. Appl. Challenges*, London, U.K., 2015, pp. 31–36.

[85] S. Chourasia and K. M. Sivalingam, "SDN based evolved packet core architecture for efficient user mobility support," in *Proc. 1st IEEE Conf. Netw. Softwarization (NetSoft)*, London, U.K., 2015, pp. 1–5.

[86] C. C. Marquezan, X. An, Z. Despotovic, R. Khalili, and A. Hecker, "Dispatching PACKET_INs to the right SDN control application via context interpretation in mobile core networks," in *Proc. IEEE/IFIP Netw. Oper. Manag. Symp. (NOMS)*, Istanbul, Turkey, 2016, pp. 686–690.

[87] C. C. Marquezan, Z. Despotovic, R. Khalili, D. Perez-Caparros, and A. Hecker, "Understanding processing latency of SDN based mobility management in mobile core networks," in *Proc. 27th IEEE Annu. Int. Symp. Pers. Indoor Mobile Radio Commun. (PIMRC)*, Valencia, Spain, 2016, pp. 2324–2330.

[88] C. C. Marquezan *et al.*, "Context awareness in next generation of mobile core networks," arXiv Preprint arXiv:1611.05353, 2016.

[89] I. F. Akyildiz, P. Wang, and S.-C. Lin, "SoftAir: A software defined networking architecture for 5G wireless systems," *Comput. Netw.*, vol. 85, pp. 1–18, Jul. 2015.

[90] M. Yang, Y. Li, B. Li, D. Jin, and S. Chen, "Service-oriented 5G network architecture: An end-to-end software defining approach," *Int. J. Commun. Syst.*, vol. 29, no. 10, pp. 1645–1657, 2016.

[91] H. Wang, S. Chen, H. Xu, M. Ai, and Y. Shi, "SoftNet: A software defined decentralized mobile network architecture toward 5G," *IEEE Netw.*, vol. 29, no. 2, pp. 16–22, Mar./Apr. 2015.

[92] A. Mohammadkhan, K. K. Ramakrishnan, A. S. Rajan, and C. Maciocco, "CleanG: A clean-slate EPC architecture and control plane protocol for next generation cellular networks," in *Proc. ACM CoNEXT Workshop Cloud Assist. Netw. (CAN)*, 2016, pp. 31–36.

[93] H.-J. Einsiedler *et al.*, "System design for 5G converged networks," in *Proc. IEEE Eur. Conf. Netw. Commun. (EuCNC)*, Paris, France, 2015, pp. 391–396.

[94] A. Roozbeh, "Distributed cloud and de-centralized control plane: A proposal for scalable control plane for 5G," in *Proc. IEEE/ACM 8th Int. Conf. Utility Cloud Comput. (UCC)*, Limassol, Cyprus, 2015, pp. 348–353.

[95] G. Hampel, M. Steiner, and T. Bu, "Applying software-defined networking to the telecom domain," in *Proc. IEEE Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, Turin, Italy, 2013, pp. 133–138.

[96] S. B. H. Said *et al.*, "New control plane in 3GPP LTE/EPC architecture for on-demand connectivity service," in *Proc. IEEE 2nd Int. Conf. Cloud Netw. (CloudNet)*, San Francisco, CA, USA, 2013, pp. 205–209.

[97] M. R. Sama, S. B. H. Said, K. Guillouard, and L. Suciu, "Enabling network programmability in LTE/EPC architecture using OpenFlow," in *Proc. 12th Int. Symp. Model. Optim. Mobile Ad Hoc Wireless Netw. (WiOpt)*, Hammamet, Tunisia, 2014, pp. 389–396.

[98] V.-G. Nguyen and Y. Kim, "Signaling load analysis in OpenFlow-enabled LTE/EPC architecture," in *Proc. Int. Conf. Inf. Commun. Technol. Convergence (ICTC)*, Busan, South Korea, 2014, pp. 734–735.

[99] V.-G. Nguyen and Y. Kim, "Proposal and evaluation of SDN-based mobile packet core networks," *EURASIP J. Wireless Commun. Netw.*, vol. 2015, no. 1, p. 172, 2015.

[100] T. Mahmoodi and S. Seetharaman, "Traffic jam: Handling the increasing volume of mobile data traffic," *IEEE Veh. Technol. Mag.*, vol. 9, no. 3, pp. 56–62, Sep. 2014.

[101] J. Pagé and J.-M. Dricot, "Software-defined networking for low-latency 5G core network," in *Proc. Int. Conf. Mil. Commun. Inf. Syst. (ICMCIS)*, Brussels, Belgium, 2016, pp. 1–7.

[102] S. Shanmugalingam and P. Bertin, "Programmable mobile core network," in *Proc. IEEE Symp. Comput. Commun. (ISCC)*, Funchal, Portugal, 2014, pp. 1–7.

[103] J. Mueller, Y. Chen, B. Reichel, V. Vlad, and T. Magedanz, "Design and implementation of a carrier grade software defined telecommunication switch and controller," in *Proc. IEEE Netw. Oper. Manag. Symp. (NOMS)*, Kraków, Poland, 2014, pp. 1–7.

[104] Y. Zhao, Z. Chen, J. Zhang, and X. Wang, "Dynamic optical resource allocation for mobile core networks with software defined elastic optical networking," *Opt. Express*, vol. 24, no. 15, pp. 16659–16673, 2016.

[105] L. Osmani *et al.*, "Building blocks for an elastic mobile core," in *Proc. 10th ACM Int. Conf. Emerg. Netw. Exp. Technol. Student Workshop (CoNEXT)*, Sydney, NSW, Australia, 2014, pp. 43–45.

[106] A. Jain, N. Sadagopan, S. K. Lohani, and M. Vutukuru, "A comparison of SDN and NFV for re-designing the LTE packet core," in *Proc. 2nd IEEE Int. Conf. Netw. Function Virtualization Softw. Defined Netw. (NFV SDN)*, Palo Alto, CA, USA, 2016, pp. 1–7.

[107] F. Z. Yousaf, J. Lessmann, P. Loureiro, and S. Schmid, "SoftEPC: Dynamic instantiation of mobile core network entities for efficient resource utilization," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Budapest, Hungary, 2013, pp. 3602–3606.

[108] T. Taleb, "Toward carrier cloud: Potential, challenges, and solutions," *IEEE Wireless Commun.*, vol. 21, no. 3, pp. 80–91, Jun. 2014.

[109] Z. A. Qazi *et al.*, "KLEIN: A minimally disruptive design for an elastic cellular core," in *Proc. Symp. SDN Res. (SOSR)*, Santa Clara, CA, USA, 2016, pp. 1–12.

[110] H. Baba, M. Matsumoto, and K. Noritake, "Lightweight virtualized evolved packet core architecture for future mobile communication," in *Proc. 2nd IEEE Wireless Commun. Netw. Conf. (WCNC)*, New Orleans, LA, USA, 2015, pp. 1811–1816.

[111] H. Hawilo, A. Shami, M. Mirahmadi, and R. Asal, "NFV: State of the art, challenges, and implementation in next generation mobile networks (vEPC)," *IEEE Netw.*, vol. 28, no. 6, pp. 18–26, Nov./Dec. 2014.

[112] W. Kiess, X. An, and S. Beker, "Software-as-a-service for the virtualization of mobile network gateways," in *Proc. IEEE Glob. Commun. Conf. (GLOBECOM)*, San Diego, CA, USA, 2015, pp. 1–6.

[113] A. M. Medhat, G. Carella, J. Mwangama, and N. Ventura, "Multi-tenancy for virtualized network functions," in *Proc. 1st IEEE Conf. Netw. Softwarization (NetSoft)*, London, U.K., 2015, pp. 1–6.

[114] S. Jeon, D. Corujo, and R. L. Aguiar, "Virtualised EPC for on-demand mobile traffic offloading in 5G environments," in *Proc. IEEE Conf. Stand. Commun. Netw. (CSCN)*, Tokyo, Japan, 2015, pp. 275–281.

[115] K. Gomez, T. Rasheed, L. Reynaud, and L. Goratti, "FME: A flexible management entity for virtualizing LTE evolved packet core," in *Proc. IEEE Netw. Oper. Manag. Symp. (NOMS)*, Kraków, Poland, 2014, pp. 1–4.

[116] K. Gomez, L. Goratti, T. Rasheed, and L. Reynaud, "Enabling disaster-resilient 4g mobile communication networks," *IEEE Commun. Mag.*, vol. 52, no. 12, pp. 66–73, Dec. 2014.

[117] T. Taleb, A. Ksentini, and A. Kobbane, "Lightweight mobile core networks for machine type communications," *IEEE Access*, vol. 2, pp. 1128–1137, 2014.

[118] Y. Ren, T. Phung-Duc, J.-C. Chen, and Z.-W. Yu, "Dynamic auto scaling algorithm (DASA) for 5G mobile networks," in *Proc. IEEE Conf. Glob. Telecommun. (GLOBECOM)*, Washington, DC, USA, 2016, pp. 1–6.

[119] T. Phung-Duc, Y. Ren, J.-C. Chen, and Z.-W. Yu, "Design and analysis of deadline and budget constrained autoscaling (DBCA) algorithm for 5G mobile networks," in *Proc. IEEE Conf. Cloud Comput. (CloudCom)*, 2016, pp. 94–101.

[120] K. Pentikousis, Y. Wang, and W. Hu, "MobileFlow: Toward software-defined mobile networks," *IEEE Commun. Mag.*, vol. 51, no. 7, pp. 44–53, Jul. 2013.

[121] A. Basta *et al.*, "SDN and NFV dynamic operation of LTE EPC gateways for time-varying traffic patterns," in *Proc. 6th Int. Conf. Mobile Netw. Manag. (MONAMI)*, 2015, pp. 63–76.

[122] W. Hahn and B. Gajic, "GW elasticity in data centers: Options to adapt to changing traffic profiles in control and user plane," in *Proc. 18th IEEE Int. Conf. Intell. Next Gener. Netw. (ICIN)*, Paris, France, 2015, pp. 16–22.

[123] E. Haleplidis *et al.*, "Building softwarized mobile infrastructures with ForCES," in *Proc. 23rd IEEE Int. Conf. Telecommun. (ICT)*, Thessaloniki, Greece, 2016, pp. 1–5.

[124] J. Wang, Z. Lv, Z. Ma, L. Sun, and Y. Sheng, "i-Net: New network architecture for 5G networks," *IEEE Commun. Mag.*, vol. 53, no. 6, pp. 44–51, Jun. 2015.

[125] X. An, W. Kiess, and D. Perez-Caparros, "Virtualization of cellular network EPC gateways based on a scalable SDN architecture," in *Proc. IEEE Glob. Commun. Conf. (GLOBECOM)*, Austin, TX, USA, 2014, pp. 2295–2301.

[126] W. Hahn, "Mobile network architecture evolution options: GW decomposition and software defined networks," in *Proc. Int. Conf. Mobile Netw. Manag.*, Abu Dhabi, UAE, 2015, pp. 118–131.

[127] M. R. Sama *et al.*, "Software-defined control of the virtualized mobile packet core," *IEEE Commun. Mag.*, vol. 53, no. 2, pp. 107–115, Feb. 2015.

[128] J. Kaippallimalil and H. A. Chan, "Network virtualization and direct Ethernet transport for packet data network connections in 5G wireless," in *Proc. IEEE Glob. Commun. Conf. (GLOBECOM)*, Austin, TX, USA, 2014, pp. 1836–1841.

[129] P. Ameigeiras *et al.*, "Link-level access cloud architecture design based on SDN for 5G networks," *IEEE Netw.*, vol. 29, no. 2, pp. 24–31, Mar./Apr. 2015.

[130] A. F. Cattoni *et al.*, "Ethernet-based mobility architecture for 5G," in *Proc. IEEE 3rd Int. Conf. Cloud Netw. (CloudNet)*, 2014, pp. 449–454.

[131] Y. Li *et al.*, "MobiSDN: Vision for mobile software defined networking for future cellular networks," in *Proc. IEEE Glob. Commun. Conf. Ind. Forum (GLOBECOM)*, 2014, pp. 1–6.

[132] A. Bradai, K. Singh, T. Ahmed, and T. Rasheed, "Cellular software defined networking: A framework," *IEEE Commun. Mag.*, vol. 53, no. 6, pp. 36–43, Jun. 2015.

[133] J. Costa-Requena *et al.*, "SDN and NFV integration in generalized mobile network architecture," in *Proc. Eur. Conf. Netw. Commun. (EuCNC)*, Paris, France, 2015, pp. 154–158.

[134] G. Hasegawa and M. Murata, "Joint bearer aggregation and control-data plane separation in LTE EPC for increasing M2M communication capacity," in *Proc. IEEE Glob. Commun. Conf. (GLOBECOM)*, San Diego, CA, USA, 2015, pp. 1–6.

[135] J. Heinonen *et al.*, "Dynamic tunnel switching for SDN-based cellular core networks," in *Proc. 4th Workshop All Things Cellular Oper. Appl. Challenges*, Chicago, IL, USA, 2014, pp. 27–32.

[136] A. Basta, W. Kellerer, M. Hoffmann, K. Hoffmann, and E.-D. Schmidt, "A virtual SDN-enabled LTE EPC architecture: A case study for S-/P-gateways functions," in *Proc. IEEE SDN Future Netw. Services (SDN4FNS)*, Trento, Italy, 2013, pp. 1–7.

[137] M. R. Sama, X. An, Q. Wei, and S. Beker, "Reshaping the mobile core network via function decomposition and network slicing for the 5G era," in *Proc. 3rd IEEE Conf. Wireless Commun. Netw. (WCNC)*, Doha, Qatar, 2016, pp. 1–7.

[138] T. Taleb *et al.*, "EASE: EPC as a service to ease mobile core network deployment over cloud," *IEEE Netw.*, vol. 29, no. 2, pp. 78–88, Mar./Apr. 2015.

[139] *Core Network Dynamic*. Accessed on Jul. 6, 2016. [Online]. Available: http://www.openepc.com/

[140] *EURECOM Research Institute*. Accessed on Jul. 6, 2016. [Online]. Available: http://www.openairinterface.org/

[141] *Intel DPDK*. Accessed on Jul. 6, 2016. [Online]. Available: http://dpdk.org/

[142] *5GPPP*. Accessed on Jul. 6, 2016. [Online]. Available: https://5g-ppp.eu/

[143] G. Premsankar, K. Ahokas, and S. Luukkainen, "Design and implementation of a distributed mobility management entity on OpenStack," in *Proc. IEEE 7th Int. Conf. Cloud Comput. Technol. Sci. (CloudCom)*, Vancouver, BC, Canada, 2015, pp. 487–490.

[144] A. Banerjee *et al.*, "Scaling the LTE control-plane for future mobile access," in *Proc. ACM 11th Int. Conf. Emerg. Netw. Exp. Technol. (CoNEXT)*, Heidelberg, Germany, 2015, Art. no. 19.

[145] J. Prados-Garzon, J. J. Ramos-Munoz, P. Ameigeiras, P. Andres-Maldonado, and J. M. Lopez-Soler, "Latency evaluation of a virtualized MME," in *Proc. Wireless Days (WD)*, Toulouse, France, 2016, pp. 1–3.

[146] P. Andres-Maldonado, P. Ameigeiras, J. Prados-Garzon, J. J. Ramos-Munoz, and J. M. Lopez-Soler, "MME support for M2M communications using network function virtualization," in *Proc. IEEE 12th Adv. Int. Conf. Telecommun. (AICT)*, 2016, pp. 106–111.

[147] F. Z. Yousaf, P. Loureiro, F. Zdarsky, T. Taleb, and M. Liebsch, "Cost analysis of initial deployment strategies for virtualized mobile core network functions," *IEEE Commun. Mag.*, vol. 53, no. 12, pp. 60–66, Dec. 2015.

[148] A. Blenk, A. Basta, M. Reisslein, and W. Kellerer, "Survey on network virtualization hypervisors for software defined networking," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 1, pp. 655–685, 1st Quart., 2016.

[149] R. Sherwood *et al.*, "Flowvisor: A network virtualization layer," OpenFlow Switch Consortium, Tech. Rep. OPENFLOW-TR-2009-1, pp. 1–13, 2009.

[150] A. Al-Shabibi *et al.*, "OpenVirteX: Make your virtual SDNs programmable," in *Proc. 3rd Workshop Hot Topics Softw. Defined Netw. (HotSDN)*, Chicago, IL, USA, 2014, pp. 25–30.

[151] X. An, F. Pianese, I. Widjaja, and U. G. Acer, "dMME: Virtualizing LTE mobility management," in *Proc. IEEE 36th Conf. Local Comput. Netw. (LCN)*, Bonn, Germany, 2011, pp. 528–536.

[152] X. An, F. Pianese, I. Widjaja, and U. G. Acer, "DMME: A distributed LTE mobility management entity," *Bell Labs Tech. J.*, vol. 17, no. 2, pp. 97–120, 2012.

[153] Y. Takano, A. Khan, M. Tamura, S. Iwashina, and T. Shimizu, "Virtualization-based scaling methods for stateful cellular network nodes using elastic core architecture," in *Proc. IEEE 6th Int. Conf. Cloud Comput. Technol. Sci. (CloudCom)*, Singapore, 2014, pp. 204–209.

[154] C. Trois, M. D. D. D. Fabro, L. C. E. de Bona, and M. Martinello, "A survey on SDN programming languages: Toward a taxonomy," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 4, pp. 2687–2712, 4th Quart., 2016.

[155] T.-X. Do, V.-G. Nguyen, and Y. H. Kim, "SDN-based mobile packet core for multicast and broadcast services," *Wireless Netw.*, pp. 1–14, Dec. 2016, doi: 10.1007/s11276-016-1433-6.

[156] Nokia Siemens Networks. (2012). *Signaling Is Growing 50% Faster Than Data Traffic*. Accessed on Jul. 6, 2016. [Online]. Available: http://docplayer.net/6278117-Signaling-is-growing-50-faster-than-data-traffic.html

[157] A. S. Rajan *et al.*, "Understanding the bottlenecks in virtualizing cellular core network functions," in *Proc. 21st IEEE Int. Workshop Local Metropolitan Area Netw. (LANMAN)*, Beijing, China, 2015, pp. 1–6.

[158] J. Xie, D. Guo, Z. Hu, T. Qu, and P. Lv, "Control plane of software defined networks: A survey," *Comput. Commun.*, vol. 67, pp. 1–10, Aug. 2015.

[159] K. Phemius, M. Bouet, and J. Leguay, "DISCO: Distributed multi-domain SDN controllers," in *Proc. IEEE Netw. Oper. Manag. Symp. (NOMS)*, Kraków, Poland, 2014, pp. 1–4.

[160] Y. Fu *et al.*, "A hybrid hierarchical control plane for flow-based large-scale software-defined networks," *IEEE Trans. Netw. Service Manag.*, vol. 12, no. 2, pp. 117–131, Jun. 2015.

[161] D. Zhou *et al.*, "Scaling up clustered network appliances with ScaleBricks," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 45, no. 4, pp. 241–254, 2015.

[162] T. Taleb and A. Ksentini, "Gateway relocation avoidance-aware network function placement in carrier cloud," in *Proc. 16th ACM Int. Conf. Model. Anal. Simulat. Wireless Mobile Syst. (MSWiM)*, Barcelona, Spain, 2013, pp. 341–346.

[163] M. Bagaa, T. Taleb, and A. Ksentini, "Service-aware network function placement for efficient traffic handling in carrier cloud," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Istanbul, Turkey, 2014, pp. 2402–2407.

[164] W. Kiess and A. Khan, "Centralized vs. distributed: On the placement of gateway functionality in 5G cellular networks," in *Proc. IEEE Glob. Commun. Conf. (GLOBECOM)*, Austin, TX, USA, 2014, pp. 4788–4793.

[165] A. Basta, W. Kellerer, M. Hoffmann, H. J. Morper, and K. Hoffmann, "Applying NFV and SDN to LTE mobile core gateways, the functions placement problem," in *Proc. 4th Workshop All Things Cellular Oper. Appl. Challenges*, Chicago, IL, USA, 2014, pp. 33–38.

[166] T. Taleb, M. Bagaa, and A. Ksentini, "User mobility-aware virtual network function placement for virtual 5G network infrastructure," in *Proc. IEEE Int. Conf. Commun. (ICC)*, London, U.K., 2015, pp. 3879–3884.

[167] A. Marotta and A. Kassler, "A power efficient and robust virtual network functions placement problem," in *Proc. 28th IEEE Int. Conf. Teletraffic Congr. (ITC)*, Würzburg, Germany, 2016, pp. 331–339.

[168] A. Baumgartner, V. S. Reddy, and T. Bauschert, "Mobile core network virtualization: A model for combined virtual core network function placement and topology optimization," in *Proc. 1st IEEE Conf. Netw. Softwarization (NetSoft)*, London, U.K., 2015, pp. 1–9.

[169] A. Baumgartner, V. S. Reddy, and T. Bauschert, "Combined virtual mobile core network function placement and topology optimization with latency bounds," in *Proc. 4th Eur. Workshop Softw. Defined Netw. (EWSDN)*, Bilbao, Spain, 2015, pp. 97–102.

[170] J. Kaippallimalil *et al.*, "Data distribution and synchronization in next generation mobile core network," in *Proc. IEEE Conf. Standards Commun. Netw. (CSCN)*, Tokyo, Japan, 2015, pp. 288–293.

[171] E. Cau *et al.*, "Efficient exploitation of mobile edge computing for virtualized 5G in EPC architectures," in *Proc. 4th IEEE Int. Conf. Mobile Cloud Comput. Services Eng. (MobileCloud)*, Oxford, U.K., 2016, pp. 100–109.

[172] S. Rajagopalan, D. Williams, H. Jamjoom, and A. Warfield, "Split/merge: System support for elastic execution in virtual middleboxes," in *Proc. 10th USENIX Symp. Netw. Syst. Design Implement. (NSDI)*, Lombard, IL, USA, 2013, pp. 227–240.

[173] A. Gember-Jacobson *et al.*, "OpenNF: Enabling innovation in network function control," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 4, pp. 163–174, 2015.

[174] J. Liu, Z. Jiang, N. Kato, O. Akashi, and A. Takahara, "Reliability evaluation for NFV deployment of future mobile broadband networks," *IEEE Wireless Commun.*, vol. 23, no. 3, pp. 90–96, Jun. 2016.

[175] B. Heller, R. Sherwood, and N. McKeown, "The controller placement problem," in *Proc. 1st Workshop Hot Topics Softw. Defined Netw. (HotSDN)*, Helsinki, Finland, 2012, pp. 7–12.

[176] Y. Hu, W. Wendong, X. Gong, X. Que, and C. Shiduan, "Reliability-aware controller placement for software-defined networks," in *Proc. IFIP/IEEE Int. Symp. Integr. Netw. Manag. (IM)*, Ghent, Belgium, 2013, pp. 672–675.

[177] H. Selvi, S. Güner, G. Gür, and F. Alagöz, "The controller placement problem in software defined mobile networks (SDMN)," in *Software Defined Mobile Networks (SDMN): Beyond LTE Network Architecture*. West Sussex, U.K.: Wiley, 2015, pp. 129–147.

[178] A. Fischer, J. F. Botero, M. T. Beck, H. De Meer, and X. Hesselbach, "Virtual network embedding: A survey," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 4, pp. 1888–1906, 4th Quart., 2013.

[179] J. G. Herrera and J. F. Botero, "Resource allocation in NFV: A comprehensive survey," *IEEE Trans. Netw. Service Manag.*, vol. 13, no. 3, pp. 518–532, Sep. 2016.

[180] *Linux Foundation*. Accessed on Jul. 6, 2016. [Online]. Available: https://www.opnfv.org/about

[181] *Telefonica Research Institute*. Accessed on Jul. 6, 2016. [Online]. Available: https://github.com/nfvlabs/openmano

[182] *Canonical*. Accessed on Jul. 6, 2016. [Online]. Available: http://www.ubuntu.com/cloud/juju

[183] *GigaSpace Technologies*. Accessed on Jul. 6, 2016. [Online]. Available: http://getcloudify.org/

[184] *Linux Foundation*. Accessed on Jul. 6, 2016. [Online]. Available: https://www.open-o.org/

[185] *ESTI NFV*. Accessed on Jul. 6, 2016. [Online]. Available: https://osm.etsi.org/

[186] AT&T. *Enhanced Control, Orchestration, Management, and Policy (ECOMP) Architecture*. Accessed on Jul. 6, 2016. [Online]. Available: about.att.com/content/dam/snrdocs/ecomp.pdf

[187] *RIFT.io*. Accessed on Jul. 6, 2016. [Online]. Available: https://riftio.com/tag/rift-ware/

[188] P. Rost *et al.*, "Mobile network architecture evolution toward 5G," *IEEE Commun. Mag.*, vol. 54, no. 5, pp. 84–91, May 2016.

[189] B. Sayadi *et al.*, "SDN for 5G mobile networks: Norma perspective," in *Proc. Int. Conf. Cogn. Radio Orient. Wireless Netw.*, Grenoble, France, 2016, pp. 741–753.

[190] Verizon. *SDN-NFV Reference Architecture v1.0*. Accessed on Jul. 6, 2016. [Online]. Available: http://innovation.verizon.com/content/dam/vic/PDF/Verizon_SDN-NFV_Reference_Architecture.pdf

[191] C. Makaya, D. Freimuth, D. Wood, and S. Calo, "Policy-based NFV management and orchestration," in *Proc. IEEE Conf. Netw. Function Virtualization Softw. Defined Netw. (NFV-SDN)*, San Francisco, CA, USA, 2015, pp. 128–134.

[192] S. Dutta, T. Taleb, and A. Ksentini, "QoE-aware elasticity support in cloud-native 5G systems," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Kuala Lumpur, Malaysia, 2016, pp. 1–6.

[193] J. Keeney, S. van der Meer, and L. Fallon, "Towards real-time management of virtualized telecommunication networks," in *Proc. 10th IEEE Int. Conf. Netw. Service Manag. (CNSM)*, Rio de Janeiro, Brazil, 2014, pp. 388–393.

[194] R. Casellas, R. Muñoz, R. Vilalta, R. Martínez, "Orchestration of IT/cloud and networks: From inter-DC interconnection to SDN/NFV 5G services," in *Proc. Int. Conf. Opt. Netw. Design Model. (ONDM)*, Cartagena, Spain, 2016, pp. 1–6.

[195] M. Gharbaoui, I. Cerutti, B. Martini, and P. Castoldi, "An orchestrator of network and cloud resources for dynamic provisioning of mobile virtual network functions," in *Proc. 2nd IEEE Conf. Netw. Softwarization (NetSoft)*, Seoul, South Korea, 2016, pp. 98–101.

[196] R. Mijumbi *et al.*, "Management and orchestration challenges in network functions virtualization," *IEEE Commun. Mag.*, vol. 54, no. 1, pp. 98–105, Jan. 2016.

[197] I. Nenadić, D. Kobal, and D. Palata, "About the telco cloud management architectures," in *Proc. 39th Int. Conv. Inf. Commun. Technol. Electron. Microelectron. (MIPRO)*, Opatija, Croatia, 2016, pp. 545–552.

[198] R. Inam *et al.*, "Towards automated service-oriented lifecycle management for 5G networks," in *Proc. IEEE 20th Conf. Emerg. Technol. Factory Autom. (ETFA)*, 2015, pp. 1–8.

[199] *Fraunhofer FOKUS Research Institute*. Accessed on Jul. 6, 2016. [Online]. Available: http://www.open5gcore.org/

[200] A. Chawre. *nwEPC - EPC SAE Gateway*. Accessed on Jul. 6, 2016. [Online]. Available: https://sourceforge.net/p/nwepc/wiki/Home/

[201] *CTTC Research Institute*. Accessed on Jul. 6, 2016. [Online]. Available: http://networks.cttc.es/mobile-networks/software-tools/lena/

[202] M. R. Sama, Y. Gourhant, and L. Suciu, "Cloud based mobile network sharing: A new model," *EAI Endorsed Trans. Ind. Netw. Intell. Syst.*, vol. 15, no. 3, pp. 1–10, 2015.

[203] K. Samdanis, X. Costa-Perez, and V. Sciancalepore, "From network sharing to multi-tenancy: The 5G network slice broker," *IEEE Commun. Mag.*, vol. 54, no. 7, pp. 32–39, Jul. 2016.

[204] N. Nikaein *et al.*, "Network store: Exploring slicing in future 5G networks," in *Proc. 10th Int. Workshop Mobility Evolving Internet Archit. (MobiArch)*, Paris, France, 2015, pp. 8–13.

[205] X. An *et al.*, "On end to end network slicing for 5G communication systems," *Trans. Emerg. Telecommun. Technol.*, vol. 28, no. 4, 2017, p. e3058–n/a.

[206] Q. Li, G. Wu, A. Papathanassiou, and U. Mukherjee, "An end-to-end network slicing framework for 5G wireless communication systems," *arXiv preprint arXiv:1608.00572*, 2016.

[207] V.-G. Nguyen and Y. H. Kim, "Slicing the next mobile packet core network," in *Proc. 11th IEEE Int. Symp. Wireless Commun. Syst. (ISWCS)*, Barcelona, Spain, 2014, pp. 901–904.

[208] *FANTASTIC-5G Project*. Accessed on Jul. 6, 2016. [Online]. Available: http://fantastic5g.eu

[209] *Docker*. [Online]. Available: https://www.docker.com/

[210] R. Bonafiglia, I. Cerrato, F. Ciaccia, M. Nemirovsky, and F. Risso, "Assessing the performance of virtualization technologies for NFV: A preliminary benchmarking," in *Proc. 4th Eur. Workshop Softw. Defined Netw. (EWSDN)*, Bilbao, Spain, 2015, pp. 67–72.

[211] J. Anderson *et al.*, "Performance considerations of network functions virtualization using containers," in *Proc. Int. Conf. Comput. Netw. Commun. (ICNC)*, 2016, pp. 1–7.

[212] J. Fontenla-González, C. Pérez-Garrido, F. Gil-Castiñeira, F. J. González-Castaño, and C. Giraldo-Rodriguez, "Lightweight container-based openEPC deployment and its evaluation," in *Proc. 2nd IEEE Conf. Netw. Softwarization (NetSoft)*, Seoul, South Korea, 2016, pp. 435–440.

[213] B. Pfaff *et al.*, "The design and implementation of open vSwitch," in *Proc. 12th USENIX Symp. Netw. Syst. Design Implement. (NSDI)*, Oakland, CA, USA, 2015, pp. 117–130.

[214] *Linux Foundation*. Accessed on Jul. 6, 2016. [Online]. Available: http://www.linuxfoundation.org/collaborate/workgroups/networking/napi

[215] J. Hwang, K. K. Ramakrishnan, and T. Wood, "NetVM: High performance and flexible networking using virtualization on commodity platforms," *IEEE Trans. Netw. Service Manag.*, vol. 12, no. 1, pp. 34–47, Mar. 2015.

[216] T. Wood, K. K. Ramakrishnan, J. Hwang, G. Liu, and W. Zhang, "Toward a software-based network: Integrating software defined networking and network function virtualization," *IEEE Netw.*, vol. 29, no. 3, pp. 36–41, May/Jun. 2015.

[217] S. Lange *et al.*, "Performance benchmarking of a software-based LTE SGW," in *Proc. 11th IEEE Int. Conf. Netw. Service Manag. (CNSM)*, Barcelona, Spain, 2015, pp. 378–383.

[218] X. An *et al.*, "SDN-based vs. software-only EPC gateways: A cost analysis," in *Proc. 2nd IEEE Conf. Netw. Softwarization (NetSoft)*, Seoul, South Korea, 2016, pp. 146–150.

[219] F. Kurtz, N. Dorsch, and C. Wietfeld, "Empirical comparison of virtualized and bare-metal switching for SDN-based 5G communication in critical infrastructures," in *Proc. 2nd IEEE Conf. Netw. Softwarization (NetSoft)*, Seoul, South Korea, 2016, pp. 453–458.

[220] B. Hirschman *et al.*, "High-performance evolved packet core signaling and bearer processing on general-purpose processors," *IEEE Netw.*, vol. 29, no. 3, pp. 6–14, May/Jun. 2015.

[221] ETSI NFV ISG. (2014). *Network Functions Virtualization; NFV Performance and Portability Best Practises*. Accessed on Jul. 6, 2016. [Online]. Available: http://www.etsi.org/deliver/etsi_gs/NFV-PER/001_099/001/01.01.01_60/gs_nfv-per001v010101p.pdf

[222] ETSI NFV ISG. (2014). *Network Functions Virtualization; Pre-Deployment Testing; Report on Validation of NFV Environments and Services*. Accessed on Jul. 6, 2016. [Online]. Available: http://www.etsi.org/deliver/etsi_gs/NFV-TST/001_099/001/01.01.01_60/gs_NFV-TST001v010101p.pdf

Van-Giang Nguyen (S'15) received the bachelor's degree in electronics and telecommunication engineering from the Hanoi University of Science and Technology, Vietnam, in 2012, and the master's degree in information and telecommunication engineering from Soongsil University, South Korea, in 2015. He is currently working toward the Ph.D. degree in computer networks and telecommunications at the Department of Computer Science and working as a Research Assistant at Distributed System and Communications (DISCO) Research Group, Karlstad University, Sweden. From 2013 to 2015, he worked as a Research Assistant at the Distributed Computing Network (DCN) Laboratory, Soongsil, University. His current research interests include software defined networking, network function virtualization, future mobile packet core networks, open source networking, and 5G networking.

Anna Brunstrom (M'00) received the B.Sc. degree in computer science and mathematics from Pepperdine University, CA, USA, in 1991, and the M.Sc. and Ph.D. degrees in computer science from the College of William & Mary, VA, USA, in 1993 and 1996, respectively. She joined the Department of Computer Science, Karlstad University, Sweden, in 1996, where she is currently a Full Professor and Research Manager for the Distributed Systems and Communications Research Group. Her research interests include transport protocol design, techniques for low latency Internet communication, multi-path communication and performance evaluation of mobile broadband systems. She has lead several externally funded research projects within these areas and served as the Principal Investigator and Coordinator from Karlstad University (KaU) in additional national and international projects. She is currently the KaU Principal Investigator within two EU H2020 projects, the NEAT project aiming to design a new, evolutive API and transport-layer architecture for the Internet, and the MONROE project proposing to design and operate a European transnational open platform for independent, multi-homed, large-scale monitoring and assessment of mobile broadband performance. She is a Co-Chair of the RTP Media Congestion Avoidance Techniques (rmcat) Working Group within the IETF. She has authored/coauthored 10 book chapters and over 150 international journal and conference papers.

Karl-Johan Grinnemo (S'00–A'05–M'06–SM'11) received the M.Sc. degree in computer science and engineering from the Linköping Institute of Technology, Sweden, in 1994, and the Ph.D. degree in computer science from Karlstad University, Sweden. He has worked for almost 15 years as an Engineer in the telecom industry, first at Ericsson and then as a Consultant at Tieto. A large part of his work has been related to Ericsson's signaling system in the mobile core and radio access network. From the Fall of 2009 until the Fall of 2010, he was on leave from Tieto and worked as acting Associate Professor at the School of Information and and Communication Technology, KTH Royal Institute of Technology. Between the Fall of 2010 and the Fall of 2014, he was an Associate Senior Lecturer at Karlstad University, and became a Senior Lecturer in the Fall of 2014. His research primarily targets application- and transport-level service quality. He has participated in the development of two partially reliable transport protocols, PRTP and PRTP-ECN; two protocols that particularly address the requirements of services with soft real-time requirements, e.g., video conferencing. Furthermore, he has suggested ways to improve the performance of SCTP: a transport protocol heavily used for the transportation of signaling traffic in the LTE core network. More recently, his research has focused on the use of multi-path transport protocols such as Multipath TCP and CMT-SCTP to increase reliability and throughput and decrease latency in IP networks. He has authored and co-authored around 30 conference and journal papers.

Javid Taheri (M'16) received the B.Sc. and M.Sc. degrees in electrical engineering from the Sharif University of Technology, Tehran, Iran, in 1998 and 2000, respectively. He received the Ph.D. degree in mobile computing from the School of Information Technologies, University of Sydney, Australia. Since 2006, he has been actively working in several fields, including: networking, optimization, parallel/distributed computing, and cloud computing. He also holds several cloud/networking related industrial certification from VMware (VCP-DCV, VCP-DT, and VCP-Cloud), Cisco (CCNA), Microsoft, etc. He is currently working as an Associate Professor at the Department of Computer Science, Karlstad University, Sweden. His major areas of interest are profiling, modelling and optimization techniques for cloud infrastructures, software defined networking, and network function virtualization.