Hindawi
Wireless Communications and Mobile Computing
Volume 2017, Article ID 8618364, 21 pages
https://doi.org/10.1155/2017/8618364

Review Article

Softwarization of Mobile Network Functions towards Agile and Energy Efficient 5G Architectures: A Survey

Dlamini Thembelihle,[1,2] Michele Rossi,[1] and Daniele Munaretto[2]

[1] *Department of Information Engineering, University of Padova, Padova, Italy*
[2] *Athonet, Bolzano Vicentino, Vicenza, Italy*

Correspondence should be addressed to Dlamini Thembelihle; dlamini@dei.unipd.it

Received 4 August 2017; Revised 19 October 2017; Accepted 23 October 2017; Published 20 November 2017

Academic Editor: Fabrizio Granelli

Future mobile networks (MNs) are required to be flexible with minimal infrastructure complexity, unlike current ones that rely on proprietary network elements to offer their services. Moreover, they are expected to make use of renewable energy to decrease their carbon footprint and of virtualization technologies for improved adaptability and flexibility, thus resulting in green and self-organized systems. In this article, we discuss the application of software defined networking (SDN) and network function virtualization (NFV) technologies towards softwarization of the mobile network functions, taking into account different architectural proposals. In addition, we elaborate on whether mobile edge computing (MEC), a new architectural concept that uses NFV techniques, can enhance communication in 5G cellular networks, reducing latency due to its proximity deployment. Besides discussing existing techniques, expounding their pros and cons and comparing state-of-the-art architectural proposals, we examine the role of machine learning and data mining tools, analyzing their use within fully SDN- and NFV-enabled mobile systems. Finally, we outline the challenges and the open issues related to evolved packet core (EPC) and MEC architectures.

1. Introduction

The evolution towards a softwarized evolved packet core (EPC) is expected to solve current mobile networks (MNs) challenges and set the way for high data rate and low latency 5G networks. Such changes should make it possible to effectively cope with the anticipated mobile data traffic explosion that will be mostly generated by smartphones, portable devices, and new traffic types, such as machine-to-machine (M2M) applications. Traditionally, as new services are introduced into the mobile space, operators upgrade the network infrastructure for a better management, while at the same time still guaranteeing a target quality of service (QoS) to their users. This introduces network complexity, as new specific hardware is usually deployed into the network for these purposes. The possibility of running network functions (NFs) in software, instead of hardware devices, is seen as a promising solution towards network complexity reduction. This technique will permit dynamically scaling the network resources for a more efficient network management.

Current research focusing on virtualizing the EPC functions is surveyed in this article. Here, we review existing EPC architectural proposals, paying attention to the different strategies involved in virtualizing the EPC functions, the adopted virtualization technology, open issues, and related challenges. Also, we discuss the current technological trends, their advantages, and drawbacks and, when possible, analyze their differences. Softwarization and virtualization of resources and services are undoubtedly among the main drivers of 5G and beyond 5G networks, as they will provide flexibility and adaptability and will facilitate network maintenance and the update of all network functions. However, to reap the full benefits of a virtualized architecture, this technology must be combined with intelligent mechanisms for handling network resources. For this reason, in this paper we also address several optimization means, including machine learning and data mining tools, and discuss how these can be employed within a virtualized mobile network.

Vendors and researchers are targeting different approaches to use virtualization within EPC architectures, such as *grouping the NFs*, running the virtualized functions on *clouds*, partitioning the NFs into *slices*, and redesigning the network to only use network function virtualization (NFV) technology. All these techniques are here discussed, emphasizing the offered advantages, the existing similarities among them, and the road ahead. In addition, we elaborate on using energy harvesting (EH) hardware to make future 5G networks as much as possible *energy neutral* and discuss how EH technology can be integrated into future softwarized mobile systems.

Software defined networking (SDN) is an emerging virtualization technology, which supports programmable interfaces to provide flexibility and agility on the network control management [1–3]. It basically consists of a number of network nodes such as switches, virtual switches, routers, and firewalls, which are automated, controlled, and reprogrammed through software commands. Open source software such as Open Flow [4] can be used to dynamically reconfigure network elements, through an *SDN controller* which can handle multiple network switches at a time. An SDN-based architecture allows dynamic and flexible network operations by decoupling the *network control plane* from the *data plane*, leveraging standard protocols which enable remote management and operation. The SDN *controller* can run *on a commodity hardware* and gives logically centralized control towards multiple switches. This enables accurate monitoring and control of traffic load within the network and is also expected to minimize operational cost, while improving load balancing and data traffic handling at the edge, through the use of generic hardware [5].

Another virtualization technology is NFV, which has recently emerged to virtualize the EPC network functions and move them from proprietary to commodity hardware platforms, as the use of specialized hardware devices has been one of the limiting factors towards mobile evolution and the fast deployment of new services within the mobile space [6]. Network functions may be firewalls, domain name servers (DNS), network address translation (NAT) services, intrusion detection systems, caching services, and so forth. These functions, which are of prime importance for the accurate operation of any network, are migrated into software and ran on top of general purpose servers. NFV is complementary to SDN but the two technologies can coexist within the same network: SDN tries to achieve a centralized control approach on switching and routing elements, thus allowing programmability of the network, while NFV moves NFs out of dedicated hardware into software that is imported into general purpose hardware. While these technologies do not possess any intrinsic cognition, they will give rise to more flexible networks, where resources could be controlled and combined in a flexible manner. This is expected to facilitate network management and to make it more efficient, moving the intelligence that is required to manage network resources, such as load balancing, intrusion detection algorithms, and firewalls into NFs. Since SDN and NFV are drivers for 5G networks, it is crucial to look at their applications within these systems.

As dictated by mobile edge computing (MEC), an emerging architectural paradigm for the design and implementation of communication networks through NFV, the virtualized network functions (VNFs) can then be deployed *at the BS*, that is, at a MEC platform colocated with the BS, or at an aggregation point (a central point that manages a set of BSs located close to each other). MEC effectively moves the network intelligence towards the network edge: the VNFs are instantiated and executed over a hosting environment, and the combination of NFV and MEC can help achieve dynamic resource/service management and configuration. These new network technologies, SDN, NFV, and MEC, are expected to improve the quality of experience (QoE) of users, while at the same time making legacy and current MNs more flexible, quickly reprogrammable, and energy efficient [7].

As a last consideration, 5G technology is currently adopting a so-called *network densification* approach, which involves the deployment of a large number of base stations (BSs), to increase the network coverage and provide higher throughput to the users. This however results in higher energy consumption, which is expected to considerably contribute to carbon emissions into the atmosphere [8, 9]. In order to minimize the carbon footprint of 5G MNs, we advocate for the integration of energy harvesting (EH) into future base stations (especially small cells). This brings the notion of energy harvesting BS (EHBS) and energy harvesting-powered MEC (EH-MEC) systems to reduce the dependence of MNs on the electricity grid. Besides helping to minimize the operational expenses (OPEX), in terms of annual electricity bills, the use of renewable energy will help extend network coverage to areas where there is insufficient electricity, or to assist during the case of a natural disaster scenario, where the conventional electricity grid may become unavailable. At the same time, the deployment of EH technology (batteries, solar cells, etc.) entails a certain capital expenditure (CAPEX) and whether or not this is convenient depends on the return of investment time. Nevertheless, current trends in battery and solar module costs are promising and suggest that in the future this equipment will be cheap enough. Further discussion and results on these aspects can be found in [10].

We stress that energy efficiency is a key consideration in future networks and can be addressed as follows. First, the network procedures have to be streamlined and carefully orchestrated, and here is where virtualization technology (also entailing new architectural designs) will play a crucial role. This will allow for a more energy efficient network operation. Second, a modern and flexible management can be combined with EH technology to reduce the carbon footprint of communication networks. In this paper, current softwarization technologies, architectures, and trends are reviewed with a special focus on energy efficiency.

The rest of the paper is structured as follows. In Section 2 we discuss the existing EPC architectural proposals, analyzing the following virtualization techniques: (A) grouping EPC functional entities (Section 2.1), (B) NFV-enabled network clouds (Section 2.2), (C) network slicing (Section 2.3), and (D) mobile edge computing (Section 2.4). In Section 3, we discuss the use of machine learning, data mining, and context-awareness within softwarized 5G networks. In Section 4, we outline some challenges and open issues related to the EPC and MEC proposals in the state of the art and, lastly, in Section 5 we provide some final considerations.

2. State-of-the-Art EPC Architectural Proposals

The EPC network consists of a number of NFs, all interconnected through an Internet protocol (IP) infrastructure to provide packet data services to the access networks. The EPC carries traffic between E-UTRAN Node Bs (eNBs for short) and the Internet on behalf of the user equipment (UE) using specialized hardware. This includes the packet data network gateway (PGW), which is responsible for IP address allocation for the UEs, as well as for QoS enforcement and flow-based charging, according to rules from the policy control and charging rules function (PCRF). It is also responsible for the filtering of downlink user IP packets into different QoS-based bearers. The serving gateway (SGW) serves as the local mobility anchor for the data bearers when the UE moves across BSs. It also retains the information about the bearers when the UE is in the idle state (known as EPS connection management) and temporarily buffers downlink data while the mobility management entity (MME) initiates paging of the UE to reestablish the bearers. It also serves as the mobility anchor for interworking with other 3GPP technologies such as general packet radio service (GPRS) and UMTS. The MME is the control node that processes the signaling between the UE and the EPC, while at the same time authenticating the UE with the home subscriber server (HSS). It is involved in the bearer activation/deactivation process and is also responsible for choosing the SGW for a UE at the initial attach time and at time of intra-LTE handover involving EPC node relocation. The non-access stratum (NAS) signaling terminates at the MME, which is also responsible for the generation and allocation of temporary identities to the UEs. It checks the authorization of the UE to camp on the service provider's public land mobile network (PLMN) and enforces UE roaming restrictions. The BS, SGW, and PGW communicate over GPRS tunneling protocols *(GTP tunnels)*, traversing a network of switches and routers. The PCRF is responsible for policy control decision making, as well as for controlling the flow-based charging functionalities in the policy control enforcement function (PCEF), which resides in the PGW. The PCRF provides the QoS authorization (QoS class identifier, QCI, and bit rates) that dictates how a certain data flow is handled by the PCEF and ensures that this is done in accordance with the UE's subscription profile [11].

Traditional EPC networks are complex and rather inflexible, use proprietary (costly) equipment, and incur high signaling overhead. To overcome these limitations, an architectural evolution that will permit dynamically scaling the EPC network functions while adapting to real world needs is in order. This can be achieved through the use of *softwarization techniques*, and such potential can be observed in the mobile network evolution trends [12]. These are illustrated in Figure 1, where the changes in the access network and EPC are shown. The evolution in the access network involved the change from the use of base transceiver station (BTS) into Node Bs. The management entity evolved from base station controller (BSC) into radio network controllers (RNC), which then became the MME. The serving GPRS support node (SGSN), which acts as a gateway to the services within the network, evolved into the SGW, and the gateway GPRS support node (GGSN), which acts as a gateway to the outside world, evolved into the PGW. In the last subfigure on the right, the *data and control planes* are decoupled and the control plane interfaces are handled by SDN controllers (acting on the data plane, indicated by gray boxes). In addition, the controllers handle *network slices*, which consist of a logical instantiation of a network, and enforce network management rules. Also, the BS in this last subfigure possesses energy harvesting capabilities, which is expected to reduce carbon emissions from mobile networks.

The EPC architectural evolution has resulted in different approaches being proposed by industry and academia towards a unified goal of having an energy efficient EPC architecture for 5G networks, thus resulting in fragmented inputs from the research/technical community. Currently available architectural designs/proposals somehow overlap in terms of the functions being softwarized, technologies used, and so forth. Therefore, it is difficult, if not impossible, to come up with a coherent system design that can act as a benchmark for future EPC designs. In this paper, we try to shed some light on the main architectural approaches, emphasizing their differences, pros, and cons.

The state-of-the-art EPC architectural proposals for next-generation networks can be categorized into the following strategies towards 5G network evolution:

(a) Grouping EPC functional entities (Section 2.1).

(b) Using NFV-enabled network clouds (Section 2.2).

(c) Network slicing (Section 2.3).

(d) Mobile edge computing (Section 2.4).

These strategies are discussed in greater detail in the following subsections.

2.1. Grouping EPC Functions. Virtualization in the EPC can be enabled by grouping the EPC network functional entities into different segments to attain less control, signaling traffic, and less congestion in the data plane [16, 17]. This can be achieved by integrating the PGW with the SGW and place them into a controller. In [16, 17] the MME is migrated with the home subscriber server (HSS) front-end (HSS-FE). The HSS-FE is an application that implements all the logical functionalities of the HSS but does not contain the user's information database. It requests the user information from the user-data repository (UDR, the central user information database) and stores these data temporarily in its cache memory. In this way, authentication and authorization are processed internally, without performing any data transmission through the network. The PGW and SGW are migrated into one virtual machine (VM) or into one virtual network function (VNF), to minimize the number of nodes involved in the data plane chain. Furthermore, the UDR, the online charging system (OCS), and the offline charging system (OFCS) are migrated into the PCRF. The idea behind this migration is that the PCRF requests user information in order to generate the required policies for each established bearer, and thus information exchange is minimized resulting in low latency for policy function generation.

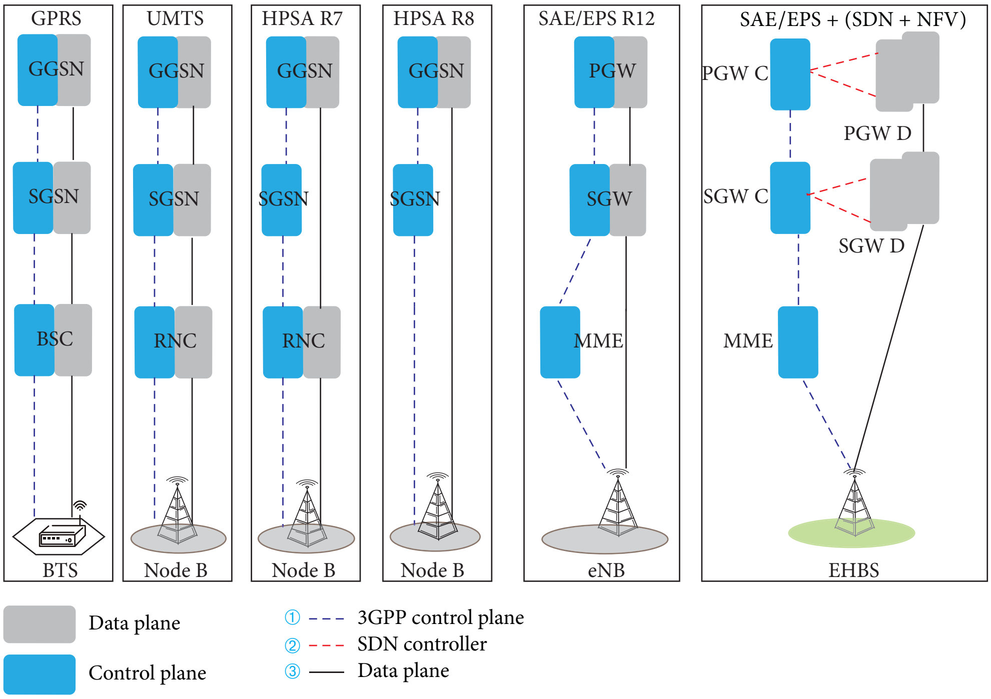


Figure 1: Illustration of the mobile network evolution trends enabling network softwarization [12].

The two approaches that we describe next [17, 18] both group some of the EPC functionalities, although in different ways. The SoftCell architecture in [18] involves disaggregation of the PGW, SGW, and MME functions and then partially implementing them in a distributed manner, as VNFs in the *controller* and *switches*. The PCRF and PCEF, usually implemented in the PGW, are grouped and implemented in the *controller*, while the packet classification is performed by the *switches*. The *controller* also performs the MME functions, in its traffic management layer. The data bearers assignment, usually performed by the SGW, is implemented by the *controller* in advance, as soon as the UE moves near a new BS. The architecture introduced in [17] adopts a "one plane grouping" strategy; that is, the EPC and Radio Access Network (RAN) entities are grouped into one common network plane, in the presence of *controllers* deployed in an upper plane. The network entities are virtualized and deployed in one plane to achieve efficient interworking, and such allows independent networks to be reconfigured in a flexible manner and automatically on the same physical infrastructure. These two architectures are discussed in greater detail next.

The SoftCell architecture [18] consists of softwarized access switches that perform fine-grained packet classification on traffic from UEs located at the EPC network edge. Then, the controller computes and installs switching rules that realize a high level service policy, specified based on the subscriber's profile. In [18], researchers try to provide flexible policies in the EPC without compromising scalability. To come up with an efficient EPC design, the factors affecting the EPC scalability were considered and publicly available network statistics were utilized. The EPC design consists of commodity middleboxes (e.g., transcoders, firewalls, and routers) and switches managed by a controller that supports flexible high level service policies. It computes and installs rules in the switches to direct traffic in both directions of a connection, thus minimizing the use of specialized network devices. The data traffic is then directed through a sequence of middleboxes optimized to the network conditions and UE locations, using the controller. In the data plane, hierarchical addressing (grouped by BS) and policy tags (identifying paths through middleboxes) are used in the EPC switches to forward traffic and the packet classification is pushed to the access switches, which are located at the EPC edge. There, fine-grained rules are specified by the controller and applied to map UE traffic into policy tags and hierarchical addresses. The SoftCell architecture leverages some properties of the EPC, that is, by considering that traffic begins at the network edge. In this way, each BS has a serving access switch (e.g., an open vSwitch [19]), located at the edge of the EPC, that performs fine-grained packet classification, whereas the radio hardware in the BS is not modified: in fact, SoftCell only changes how the BS communicates with the EPC, providing softwarized EPC functional elements.

The EPC architecture proposed in [17] uses SDN, splitting the network into three planes; (i) *application* plane, (ii) *control* plane, and (iii) *forwarding* plane. In the forwarding plane, the EPC coexists with the access network as all the control functions are moved into the control plane. By doing so, the forwarding plane consists of virtualized network devices that perform switching and packet forwarding, according to the SDN paradigm. To relieve the EPC from traffic load, NFV is exploited to instantiate network functionalities such as PGW in the access network, while SDN triggers path reconfiguration of data traffic. In addition, *data caching* strategies are exploited to minimize the traffic that goes through the EPC; that is, NFV techniques are used for caching popular content and store it on the EPC and access network when the network is idle, thus reducing the pressure in the data links and SDN controllers. In this way, latency can be minimized as content is cached locally, and better network management can be achieved through the use of virtualized EPC NFs.

Discussion. The proposed architectures [17, 18] differ from one another. In [17], the networks, RAN and EPC, and the VNFs are in one plane whereas, in [18], the distinction between RAN and EPC is maintained, with only the EPC entities being virtualized. However, both of them use an *SDN controller* for network management and policy enforcement, under different user mobility and traffic load variations. That is, the infrastructures are deployed with simplified and virtualized network devices, whose software is decoupled from the hardware and centralized to the control entity. Entities such as switches solely take the strategy developed by the *controller* and forward traffic to access networks. The architecture in [17] avails the potential of reducing latency through traffic offloading and caching in the RAN and EPC network. An advantage of [18], is the ability to "mix and match" network devices from different vendors, which is also possible in [17]. The challenge in [18] is that the centralized controller may become a source of bottleneck for the network operation when the network scales up, due to the fact that all the control functionalities are pushed towards it. To reduce this burden, a wireless side of the network introducing the concept of Cloud-RAN (C-RAN) is proposed in [17], where the base band unit (BBU) pool has both control and data forwarding functions, in addition to the *controllers* deployed in the control plane. The logically centralized BBU pool has the network wide view of the RAN and the EPC, yielding a seamless integration of the wired and wireless parts of the network. Through collaborative control, contents can be optimally distributed and stored in diverse devices of the EPC and the RAN via caching and broadcasting, thus overcoming the bottleneck problem of [18].

Energy Efficiency. The emergence of a network paradigm supporting social requirements is one of the aims of 5G. With softwarization, information centric networking (ICN), as one of the candidates, can be enabled where in-network caching can be provided. The content cache server caches contents passing through the node and then autonomously selects which contents to cache based on the need of the mobile users accessing the node, that is, based on the content request frequency. This approach can reduce the overall energy consumption within the network, since contents are cached and stored in close proximity to mobile users. In addition, in-network caching reduces the traffic (content transmission) within the network and also facilitates in-network data processing [14], whereby each network node carries out some data processing and service provisioning. This leaves some of the nodes within the network inactive, thus enabling energy saving procedures (e.g., switching-off unused nodes). The proposed architecture in [17] can improve energy efficiency as it allows content caching in the EPC and the RAN, while [18] does not employ any energy efficient EPC procedures, except for implementing policies for data traffic directions, which can also make some of the middleboxes inactive.

2.2. NFV-Enabled Network Cloud for EPC. The virtualized EPC on VMware vCloud NFV platform is introduced in [20] to enable the degree of flexibility that will make it possible to deploy services closer to the edge, while managing, monitoring, and automatically scaling the heavier workload. Such flexibility and diversity of the virtual EPC combinations (availed by the VNF deployment approach) can be delivered over the virtual Cloud (vCloud) NFV platform with lowered operational costs. It mainly abstracts the EPC network functions, decomposing and allowing them to run as software instances (virtual machines), on standard servers. This allows service providers to customize services and policies to design networks in new ways, to reduce costs and simplify operations.

Another cloud-based approach that provides all network and access functionalities is proposed in [21], where the network cloud utilizes NFV for dynamic deployment and scaling of the NFs. The key elements in this architecture are (1) a data-driven network intelligence for optimizing network resources usage and planning and (2) relaying and nesting techniques: to support multiple devices, group mobility, and nomadic hotspots. The EPC is virtualized into three parts, namely, (i) *control* plane entity (CPE), which is responsible for authentication, mobility management, radio resource control and non-access stratum (NAS), and access stratum (AS) integration, (ii) the *user* plane entity (UPE), acting as a gateway, mobility anchor, and over-the-air (OTA) security provisioner, and, lastly, (iii) the *network intelligence* (NI) plane is for the extraction of actionable insights from big data, orchestration, or required services and functionalities (e.g., traffic optimization, caching). The realization of the network cloud can be achieved by enabling virtual function instances to be hosted in data centers when needed. The use of virtualization techniques will enable quick deployment and scalability of CPE and UPE functions. For example, in case of a natural disaster, with this technology the local data center is maybe unable to cope with the traffic upsurge; therefore, additional capacity can be sourced quickly from other data centers.

Discussion. The proposed architectures [20, 21] both use a *cloud-based approach* with NFV platform that enables a dynamic deployment of edge networks, the scaling of NFs, network monitoring, and load management. Also, they can pool capacity of resources when required, intelligently. The difference between them is that, in [20], only the NFV platform is available for enabling network services provision while, in [21], the combination of SDN and NFV is utilized to provide network control and to host the network intelligence. The architecture proposed in [20] is commercially available. The driving force behind such architectures is the use of virtualization tools for instantiating each service when required, and these tools are discussed in the following.

Virtualization Tools. Virtualization was introduced to primarily optimize hardware utilization by overlaying one operating system on top of another. Each of the systems consequently shares hardware resources to support underlying processes. The tools that enable virtualization can be categorized into (i) *hypervisors* and (ii) *docker engine* and are discussed next.

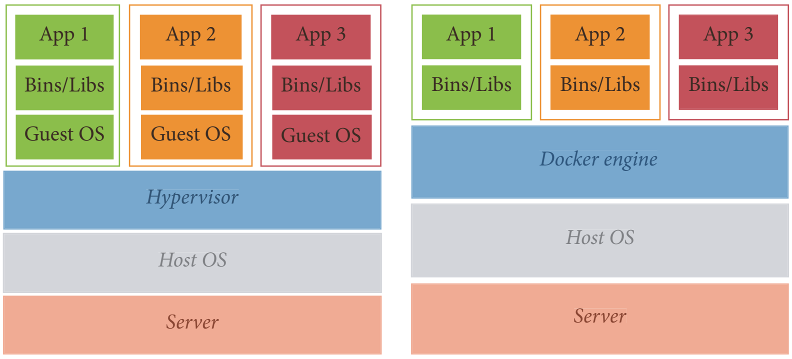
(i) Hypervisors. These are functions which abstract or isolated operating systems and applications from the underlying computer hardware. This abstraction allows the underlying "host machine" hardware to independently operate one or more virtual machines as "guest machines" (also referred to as "guest VMs"), allowing them to share the system's physical computing resources, such as processing time, memory space, and network bandwidth. A new agnostic OS is generated to manage the underlying resources. For example, with a Windows system based hypervisor running on underlying physical hardware, another system running on virtual resources can be generated and Linux can be installed on it. This second OS will be the guest OS. The base OS (Windows in this example) simply adapts the underlying physical hardware resources to accommodate the processing requirements of the guest OS. Since hypervisors sit between the actual physical hardware and the guest operating system (OS), they are also referred to as virtual machine monitors (VMMs). They are usually implemented as a software layer, for example, VMware vSphere or Microsoft Hyper-V, but they can also be implemented as code embedded in a system's firmware. Other existing hypervisors include Citrix XenServer (Xen) and Kernel Virtual Machine (open source KVM). Xen is based on the open source Xen Project [22]. This hypervisor is a bare metal virtualization platform that has been included in the Linux kernel. It is used for a number of different commercial and open source applications, such as server virtualization, infrastructure as a service (IaaS), desktop virtualization, security applications, and embedded and hardware appliances. KVM is another hypervisor built into the Linux kernel, that is, a special operating mode of QEMU (which is a generic and open source machine emulator and virtualizer) that uses CPU extensions for virtualization via a kernel module. The kernel module provides the core virtualization infrastructure and a processor specific module. Using KVM, one can run multiple virtual machines running unmodified Linux or Windows images. Each virtual machine has private virtualized hardware: a network card, disk, graphics adapter, and so forth. The kernel component of KVM is included in mainline Linux, as of 2.6.20.

Another fundamental concept is that of a virtual machine (VM), which is an operating system (OS) or application environment that is installed on software, which imitates dedicated hardware. Each VM includes a full copy of an operating system, one or more apps, necessary binaries (`Bins`) and libraries (`Libs`) taking up tens of GBs. The hypervisor allows multiple VMs to run on a single machine. In Figure 2(a), we observe that each VM has a virtual OS of its own and the hypervisor provides the VMs with a platform to manage and execute multiple guest OS and allows host computers to share their resources among them. A drawback of VMs is that they can be slow to boot.

(ii) Docker Engine. Docker is an open platform, or a software technology written in the `Go` programming language, and takes advantage of several features of the Linux kernel to deliver its functionality, for developing and running applications. It runs natively on Linux systems, where it uses Linux kernel features like `namespaces`, to provide isolated workspace, and `control groups` (cgroups), a technology that limits an application to a specific set of resources, to create a loosely isolated environment called a *container*, thus avoiding the overhead of starting and maintaining VMs. Mainly, it provides tooling, that is, software packaging tools that can package an application and its dependence in a virtual container that can run on any Linux server, and a platform to manage the *containers* lifecycle.

Containers are abstraction units for isolating applications and their dependence, that can run in any environment. They can run on the same machine, on top of the docker engine, sharing the OS kernel with other containers. They occupy less memory space than VMs, and this allows them to have a shorter start-up time. Mainly, they enable OS level virtualization whereas VMs provide hardware virtualization. However, they are similar as they also have a private space for processing, executing commands as root, and making use of private network interface and IP addresses. The main difference between containers and VMs is that containers share the host system's kernel with other containers, and also they do not bundle a full OS, instead only libraries and settings required to make the software work are needed. Figure 2(b) shows that containers only package up the user space and not the kernel or virtual hardware, like a VM does. Each container gets its own isolated user space to allow multiple containers to run on a single host machine. We observe that the entire OS level architecture is being shared across them. The only parts that are created from scratch are the `Bins` and `Libs`. This makes containers lightweight: by isolating application environments, they achieve better resource utilization than hypervisors. Each application uses its own set of resources, without affecting the overall performance of the server. They are therefore ideal for enterprises which concurrently run multiple processes on the same server. Despite the process isolation and lightweight character, containers are less secure and more vulnerable compared to hypervisors. By only accessing a couple of namespaces through `libcontainers`, the default container format, and leaving out the rest of the kernel subsystems, it is possible to crack and hack through the OS. In [14, 15], a dockerized EPC is presented as one architecture utilizing containers as a virtualization tool.

Discussion. NFV is envisioned to be a key virtualization technology in 5G. One of the challenges to be faced by developers is the selection of the appropriate virtualization tool to use when developing the virtualized NFs platforms, that is, either hypervisors or the docker engine, or to simply let VMs and docker containers coexist in the same platform; see [13]. Currently, the most commonly used architecture is the one shown in Figure 2(a) (key platforms include VMware vSphere, Microsoft Hyper-V, Citrix XenServer, or KVM). However, the future looks different; docker will probably coexist with hypervisors as the use of containers, running



(a)

(b)

Figure 2: An illustration showing the structure of virtual machines on top of the hypervisor (a) and containers on top of the docker engine (b) [13].

on top of the docker engine, speeds up innovation, requires less space and can be deployed across different platforms and hypervisors, with VMs on top, and allows running multiple applications on multiple VMs. The combination of the virtualization tools can be beneficial as operators cannot be restricted to one infrastructure; instead they can simply develop applications once and then run them on any infrastructure [13].

Energy Efficiency. The proposed architectures [20, 21] both make use of NFV and cloud computing platforms. These technologies avail the possibility of scaling down resources when the demand is low and schedule resources based on demand; that is, resources can be outsourced through infrastructure as a service (IaaS) business models during peak hours. The dynamic scaling of resources avails opportunities for improving energy efficiency (EE) within the cloud platform, as presented in [21]. Also, it allows the network cloud to collect user-centric, network-centric, and context-centric data. Through this information centric approach, intelligent algorithms, mainly network optimization tools, can be applied to the aggregated data in order to provide useful input for network planning and resource management, thus improving EE.

The virtualization tools used can also play a role towards EE improvement within the network. For example, the hypervisor can report resource usage to the orchestrator in order to trigger system automated sleep mode states and also to implement policies provided by management and orchestration, which includes power management and power stepping [23]. Since VNFs provide on-demand access to a pool of shared resources, where the locus of energy consumption for components is the VM instance where the VNF is instantiated. Therefore, the NFV framework can exploit the potential possessed by the virtualization technologies in order to reduce the energy consumption in future networks.

2.3. Network Slicing. Supporting the separation of the control and user plane functions is one of the most significant principles of the 5G EPC architecture. With the advent of virtualization (NFV, SDN, and cloud technology), it is now possible to build networks in a more scalable, flexible, and dynamic way. The concept of flexibility applies not only to the hardware and software parts of the network, but also to its management. For example, setting up a network instance that uses different network functions optimized to deliver a specific service needs to be automated. With virtualization technology, resources can be isolated resulting in a so-called *network slice*, which refers to an isolated set of (programmable) resources to enable network functions and services. With network slicing, one physical network is sliced into multiple virtual ones, each architected and optimized for a specific service or application.

The dockerized EPC architecture using the FLARE node (an open deeply programmable network node architecture) is introduced as a network slicing architecture example in [14, 15, 24]. The EPC is decomposed into *network slices*, each implementing a network service as illustrated in Figure 3. In this figure, there are a number of FLARE control slices consisting of the virtualized MME, linked with the HSS and SP-GW (integrating SGW and PGW) hosted in a docker platform (a software container platform). In the data plane, only the SP-GW is present for data management for each FLARE slice. This architecture provides the EPC NFs in each network slice. In [24], the FLARE architecture was used to resolve technical challenges that include ease of programming, reasonable and predictable performance, and the isolation among multiple concurrent logics for a faster and modular programming of the SDN data plane. To facilitate programming, the Toy-Block networking programming model [25] has been introduced and furthermore, the combination of computational resources was introduced to obtain a reasonable and predictable performance. To improve performance, a lightweight resource virtualization technique, called *resource container for isolation of multiple logics*, was proposed. The cores were partitioned into groups, each with a resource container. The isolation used virtualization techniques.

Another architecture based on network slicing, called mobile-central office rearchitected as a datacenter (M-CORD), has been proposed [26]. M-CORD is expected to deliver the agility of a cloud provider, also featuring software

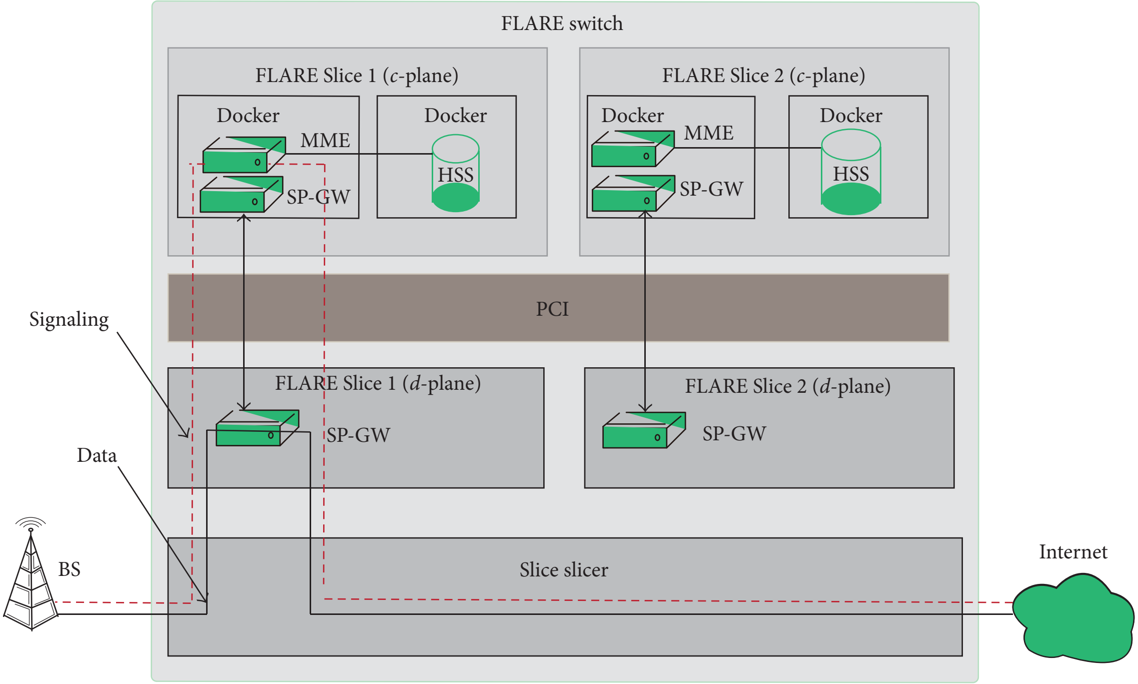


Figure 3: Dockerized EPC on FLARE node [14, 15]. The slice "slicer" works as a controller to distribute packets to an individual slice according to the input/output port the packets use or to the tag information that they may contain. PCI means "peripheral component interconnect" and acts as an interface between the data and the control slices.

platforms that enable rapid creation of new services. Its objectives are to enhance resource utilization, especially in terms of radio spectrum, to provide customized services and better QoE to customers and to offer an agile and cost-efficient deployment through the virtualization of the EPC. The EPC consists of sliced SGW for the user plane, and the control functions of the EPC are hosted in an SDN controller. The data plane of the EPC is expected to be connectionless (no GTP), and the signaling side S11 (MME to SGW) will be extended to an SDN controller. However, the S1-MME interface (eNB to MME) will be largely intact. It is expected that this architecture will involve the integration of vendor solutions within the CORD service framework. Such include the use of ONOS (for "open network operating system") as an SDN controller, through the development of an SGW/PGW application in ONOS. The role of such application will be to map restful APIs from the MME and the application of Open Flow rules towards a Radisys SGW/PGW data plane component.

Discussion. End-to-end (E2E) network quality is an important consideration when using SDN technology. E2E QoS depends on the radio access, the EPC, and the wired part of networks, and 5G systems should have the capability to tailor it by organizing functions and connectivity so as to satisfy the system requirements, for example, mission critical applications. Mobile users are supposed to be satisfied with the quality, when using any applications anytime, anywhere. Considering the wide variety of application domains to be supported by 5G, it is necessary to extend the concept of slicing to cover a wider range of use cases than those targeted by the current SDN/NFV technologies and to also address a number of issues on how to utilize slices created on top of programmable software defined infrastructures. There exists a gap between current SDN technology developments, as noted in current technology reports [14], and the functionalities that are required by 5G networks for E2E quality. For example, current focus is towards coming up with robust SDN controllers for network control and rules enforcement in networks, that is, reducing bottlenecks, while the use of SDN controllers to manage network slices is overlooked, yet management of network slices, under latency constraints, is one of 5G requirements whereby each *slice* needs to be controlled in an efficient way for service provision, based on the quantity of data and quality requirements. Apart from slice(s) management, even the radio side of the network needs to be managed for resource reservation, especially in cases of disasters, thus guaranteeing end-to-end quality. In [14], the slice "slicer" acts as a controller for packet distribution not for slice(s) management; thus SDN technology advancement must consider slice(s) management.

The observed similarities in the proposed architectures [14, 26] are that both of them reorganize the network into *network slices*, respectively, consisting of data and control planes. Also, they have controllers which perform different functions; the slice "slicer" works as a controller to distribute packets to an individual slice in [14], whereas the SDN controller in [26] performs the control functions. The difference between them is that the MME and the integrated gateway, SP-GW, are implemented in the docker platform for control purposes [14], whereas, in [26], the controller manages the network as it performs control functionalities.

Energy Efficiency. In future mobile networks, network slicing is considered as a key in realizing network flexibility [14]; therefore it is imperative that the RAN and EPC network works jointly in executing the "extreme flexibility." Achieving efficient network flexibility will help to serve devices with different quality of service, through slice isolation and resources provision. However, network slicing still poses a challenge as there are many dimensions and technologies included in this paradigm [27]. The challenges include RAN and EPC reconstruction to support end-to-end network slicing, slice management, and cooperation with other 5G technologies [27]. In addition, focus towards energy efficiency in network slicing is still lacking, as observed in [15, 26], yet resources for the network slices can be set up based on various service characteristics, for example, bandwidth demand and latency demand, over the same or shared infrastructure. The slice manager, if present, can be able to allocate resources per slice and also trigger energy savings strategies in unused resources. Since each slice provides customized connectivity and also runs on the same, shared infrastructure, by employing "soft resource scaling" (allocating reduced time for each resource usage, by each VM) [28] resource usage can be minimized thus improving energy efficiency within the EPC.

2.4. Mobile Edge Computing. The evolution towards 5G is expected to bring about several new ways of designing networks, so that the promise of always on, high-bandwidth, low latency, massive networks will become a reality. The concept of MEC is one such evolution and it is based on NFV. MEC makes use of the large amount of power and storage space distributed at the network edge, which can yield sufficient capacities to perform computation-intensive and latency-critical tasks on mobile devices. Mainly, it aims at enabling cloud computing capabilities and information technology (IT) services in close proximity to end users, by pushing computation and storage resources towards the network edge (i.e., placing computing and storage resources in the access networks to improve delivery of content/applications to end users). The direct interaction between mobile devices and *edge servers* through wireless communications brings the possibility of supporting applications with ultra-low latency requirements, prolonging device battery life and facilitating highly efficient network operations. This technology is expected to enable operators to better adapt traffic to the prevailing radio conditions, optimize service quality, and improve network efficiency [29].

MEC uses a virtualization platform to run applications at the mobile network edge and this can turn a cell/BS into a *computation hub*. Some of the computing functions that formerly only existed in the EPC are now moved out to the network edge. By disaggregating network services and functions out of the EPC, significant savings in cost, latency, round trip time (RTT), traffic download time, physical security (no need for security provision to facilities as the network consists of virtualized network devices), and caching efficiency [29] can be attained. Energy efficiency is a major concern in the design of 5G systems and, as such, is also a prime concern for the design of MEC architectures [30].

In the following subsections, we provide an overview of (1) the reference ETSI MEC architecture, (2) the integration of renewable energy into MEC systems, (3) the optimization of MEC systems, and (4) we provide a discussion of relevant use cases.

2.4.1. ETSI MEC Reference Architecture. MEC is a foundational network architecture concept which is expected to help 5G networks deliver the significant capability gains that are required by IoT, enhanced mobile broadband, virtual reality, self-driving vehicles, and many other applications. It will also provide a set of services that can be consumed by applications running on the MEC platform. These services will offer real time network information such as radio conditions, network statistics, and the location of connected devices to the running applications. Different architectures are being proposed for future 5G MEC networks. In [31], a MEC NFV-based architecture is proposed and new APIs are opened, availing hosting environments for both mobile operators and external players, which can make use of the access network related information for their services. This architecture consists of the *infrastructure* plane, the *control application* plane, and the *management* plane. In addition, there is an *orchestration and management* plane hosting MEC management activities. The hosting environment consists of hardware resources, a virtualization infrastructure (virtual computation, storage, and network resources), and a set of associated management services for MEC applications. The major components of the MEC reference architecture [32, 33] are the MEC application platform, providing infrastructure services, radio network information services (RNIS, which provides radio network information systems features), and the user location services (LOC, which provides UE location features). The services are hosted on the MEC server deployed in proximity to the BSs or colocated with it. Through the RNIS function, the radio network data and other real time context information can be exposed to authorized MEC network management applications. The MEC platform functions and applications are linked with the traffic offloading function (TOF), which is located at the user's data plane. The TOF is responsible for service prioritization and routes selection, policy-based, and user-data stream to and from applications. The overall view of the deployed MEC servers is maintained by the mobile edge orchestrator, which determines the optimum location(s) for instantiating a MEC application, and the virtualized infrastructure manager (VIM), which is responsible for resource management of the virtualized infrastructure. In the orchestration and management plane, an additional manager is introduced, which is dedicated to managing the MEC platform, including its services and the respective APIs.

An optimal deployment of MEC servers is key [34]. There are several options where the MEC server can be deployed within the network edge, and the ETSI ISG specifies that the MEC platform can either be part of eNB or be run on an external server that can be deployed between the eNBs and the EPC. Such approach allows different vendors to develop applications and deploy them within the access network.

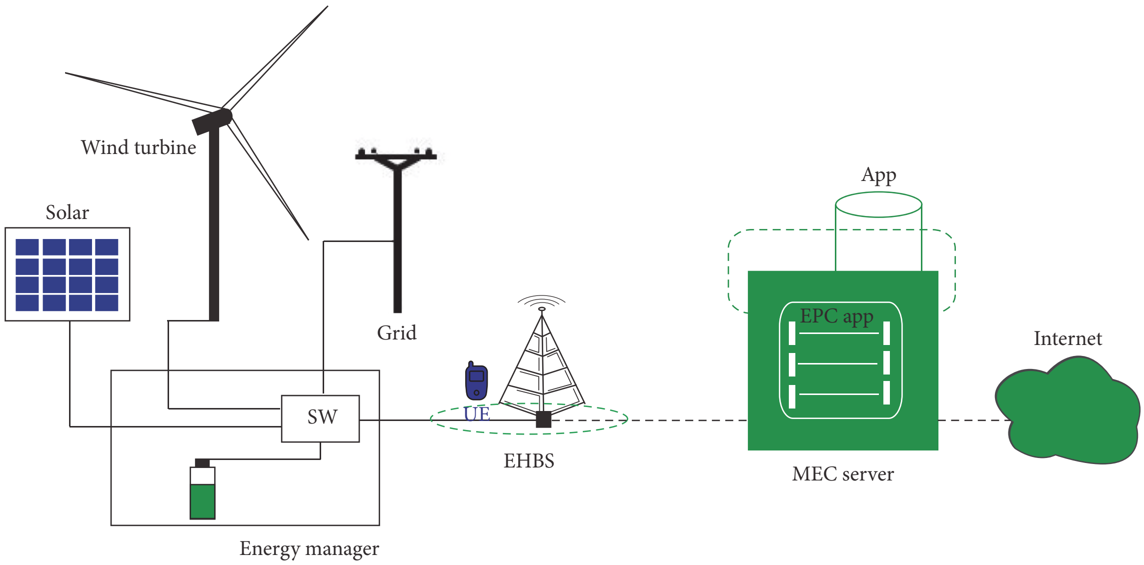


Figure 4: MEC-based network design with energy harvesting capabilities. The electromechanical switch (SW) is responsible for selecting the appropriate source of energy for powering the base station (BS) *and* the MEC server, if they are colocated, or only powering the BS, if they are not colocated.

To minimize latency, the MEC platform can be placed inside the BS as it is the first connection point for the mobile user. When considering colocated BSs, from one mobile operator, it may be beneficial to place the MEC platform at an aggregation point, a point within range to a set of BSs, as this can centralize resources and avails BS management without incurring significant amount of latency. The most viable MEC-based network design, in the mobile operator's network, is shown in Figure 4. There, the MEC server is placed between the base station (EHBS in the figure) and the EPC. The MEC platform should be transparent to the GTP protocol for IP packets encapsulation and decapsulation, after MEC services have been provided to the user via the locally hosted MEC applications, that is, the applications hosted by the MEC platform located in proximity to the BS and mobile user. The control plane is SCTP-based, S1-MME, and the user plane is GTP-based, S1-U. Mainly, the MEC server(s) host the applications (App) and MEC enabled-services, which are accessible via the application programmable interfaces (APIs). The access to the Internet (and operators services) is via the SGi interface located between the EPC (present in the MEC server as an EPC application, "EPC App") and the Internet, and the radio access is via the S1 interface. In Figure 4, an energy harvesting enabled-BS is shown, emphasizing the role of renewable energy for the energetic self-sustainability of future mobile networks (see also the next subsection "energy harvesting").

A new functional architecture that is worth mentioning is proposed within the COMBO project [35], where a new element called the universal access gateway (UAG) is introduced. The EPC gateways (SGW/PGW) are moved into the UAG. They are located in a central office closer to end users so that they can access the national IP network and reach the Internet sooner, thus enhancing latency and saving transport resources. By placing the mobile gateways closer to end users, all traffic that does not need specific treatment is delivered locally to the operator IP EPC network. A proper functional integration is of great importance to offer virtual resources at the mobile edge, while effectively adapting to the actual network load.

2.4.2. Energy Harvesting. EH is orthogonal to what we have discussed so far. In fact, up to now we have elaborated on increasing the efficiency in the network management, whereas the aim of energy harvesting is to supply network apparatuses, reducing their carbon footprint. Current mobile systems are powered using grid energy, which inevitably emits large amounts of carbon into the atmosphere. Recently, off-grid renewable energy sources such as solar radiation and wind energy have emerged as viable and promising sources for various IT systems due to the advancement of energy harvesting techniques [36, 37]; see Figure 4. where the BS is powered by either solar or wind energy. The authors in [38] observed that solar energy is more suitable for workloads with high peak-to-mean ratio (PMR), while wind energy fits better for workloads with small PMR. This avails the development of proper strategies for renewable energy provisioning for edge servers with the objective of eliminating any chance of energy shortage. This can be achieved by selecting the appropriate renewable energy source at each time instance taking into account current and forecast traffic loads. Since MEC servers are small-scale data centers, each of which consumes less energy than conventional cloud data centers, it is expected that powering the MEC infrastructure with renewable energy sources will reduce the overall network energy consumption. This is important, especially in light of the dense deployment pattern that is foreseen in 5G systems. A challenge to be addressed for renewable energy powered MEC systems is the green-aware resource allocation and computation offloading, which should take the renewable energy constraints into account (e.g., energy availability, battery charge/discharge cycles). Also, with renewable energy sources, the energy side information (ESI), which indicates the amount of available renewable energy, will play a key role in decision making for storage and computing applications. MEC devices may also be energized through wireless power transfer (WPT) [39, 40], when the renewable energy is insufficient. WPT may be exploited for computational offloading in mobile devices [41] or data offloading for MEC in future networks [42, 43]. We stress, however, that the energy transfer efficiency of current WPT techniques is still very low, and that new methods are required to increase it and make WPT appealing in practice; see [44].

2.4.3. MEC Optimization. MEC servers can allow their resources to be jointly managed for serving a large number of mobile devices. However, as the network size increases the resource management becomes a large scale optimization problem with respect to offloading decisions, radio, and computational resource allocation variables. For energy efficiency reasons, it is desirable that MEC systems make use of low complexity optimization algorithms with moderate signaling overhead. Despite recent advancements in large scale optimization algorithms for radio resource management, these may be difficult to be verbatim-applied due to the combinatorial and nonconvex nature of computation offloading problems, which thus require ad hoc solutions [45], able to handle huge traffic volumes. In the following paragraphs, some optimization examples are discussed.

(1) Optimization in MEC Systems Using Renewable Energy. The use of renewable energy in MEC systems was investigated in [46, 47], focusing on EH powered MEC servers and EH powered mobile devices. In [46], a reinforcement learning-based online algorithm was used to enhance decision making for EH powered MEC systems in determining the amount of workload to be offloaded from the edge servers to the central cloud, as well as the processing speed of the edge server, taking into account the congestion status in the EPC, the computation workload, and the ESI. Furthermore, a Lyapunov optimization technique based on channel state indicator (CSI) and ESI was used to obtain dynamic offloading policies for EH powered mobile devices [47]. Both optimization techniques, reinforcement learning (online) and Lyapunov optimization based, were used to study small-scale MEC deployments, without taking into account large scale networks. Further work shall be carried out to scale up and this still poses a challenge to researchers as data demand increases. Robust optimization algorithms for handling large scale deployments of MEC servers have to be developed under the ESI constraint. In [48], a new energy efficient design principle for the BS (colocated with the MEC server) to minimize its energy consumption, while ensuring self-sustainable computation at the mobile devices (through WPT), is investigated using the Lagrangian duality method. A multiuser MEC system consisting of a multiantenna access point and multiple users is assumed. Each mobile device is equipped with two antennas, one for WPT and the other for computational offloading. The antennas operate over different frequency bands such that WPT and computational offloading can be performed simultaneously, without mutual interference. Users rely on their harvested wireless energy to execute the latency-sensitive computational tasks either via local computing or (possibly partial) offloading to the MEC server. The optimal policy under energy harvesting constraints is obtained leveraging the Lagrange duality [49] and the ellipsoid methods [50].

(2) Optimization for MEC Offloading. During the MEC computation offloading process, the energy consumption for processing the task involves the energy spent by the mobile device to transmit the data to the MEC server and that involved in the computation at the server side. To minimize the system energy consumption under latency constraints, a three-stage energy optimization scheme, that is, (i) mobile device classification, (ii) priority determination, and (iii) radio resource allocation, is proposed in [43]. This algorithm involves priority assignment and classification type for mobile devices to reduce the problem complexity. The problem is formulated as a special maximum cardinality bin packing program [51], where mobile devices choose their task allocation mode through binary strategies, taking into account the transmission interference with other terminals and the limited radio resources. The obtained numerical results demonstrate that the approach increases the energy efficiency of the MEC system. A centralized two-stage resource allocation optimization scheme is proposed in [41] with the objective of minimizing the mobile energy consumption in a MEC offloading system using TDMA and OFDMA. The MEC server is assumed to have knowledge of the local computation energy consumption, of channel gains and fairness indices for all users. The resource allocation is formulated as a convex optimization problem, whose solution is obtained through dual-decomposition coupled with a relaxation constraint on the cloud capacity.

(3) Optimization for Mobility Management in MEC Systems. Conventionally, mobile services are provided to users through the operators EPC network; that is, the traffic goes through the EPC from the Web server(s) hosted within the Internet, while mobile users can move across radio cells. However, mobility management in MEC systems poses a significant challenge due to the fact that the systems will be implemented in the heterogeneous networks (HetNets) architecture consisting of multiple macro, small cell, and eNBs. Thus, user's movement will result in frequent handovers among these radio access technologies (RAT). This challenge implies that the way in which mobility is currently handled may no longer be appropriate in MEC systems, and some rethinking about how it can be best handled at the network edge is in order. In [52], a mobility aware offloading decision maker named *Mob-aware* is proposed. This scheme estimates future network changes based on typical user's mobility patterns. In this paper, it is observed that user mobility possesses some regularity [53], and this can provide relevant information on what kind of changes can occur within the network in the near future. For example, it is possible to predict a sequence of networks to which users

will be connected and thus apply some sort of predictive control policy in making offload decisions. The authors of [53] developed a mobility pattern model and then used trace-based simulation with real log data traces from 14 Android users to validate the mobility regularity. Along the same lines, in [54] user mobility patterns and cloudlets admission control policies were investigated, and the minimization problem for computation and offloading costs was modeled using Markov decision processes (MDP). In [55], the contact patterns regulated by mobile devices mobility were exploited and opportunistic computation offloading strategies were derived using convex optimization techniques. To maximize the edge computation performance for a user, while keeping the user's communication energy consumption below a certain threshold, a user-centric mobility management scheme exploiting Lyapunov optimization [56] and multi-armed bandits [57] has been proposed in [58]. Based on these optimization techniques, an online algorithm was developed by considering user-side state information (user location, candidate BS, and workload rate) and BS-side state information (background workload, maximum service rate, and uplink channel conditions). The overall scheme is online and was compared against an oracle-based look-ahead algorithm, which has full knowledge of the system state information. The performance of the online algorithm is satisfactory, providing optimal delay performance when the energy constraint is large and achieving a tradeoff between optimal delay and optimal energy consumption otherwise, while still satisfying the energy budget constraint.

2.4.4. MEC Use Cases Scenarios. There are several use cases that will greatly benefit from pushing MEC towards the edges of the network, into small cells, Wi-Fi access points, media gateways, and even extending edge computing to user devices themselves. The benefits of *edge storage* or *computation* include low latency, traffic optimization, agility and adaptability, and context-awareness. In combination with visibility into prevailing radio conditions via the MEC's RNIS function, applications can adapt content delivery in real time to ensure an adequate QoE to the end users [32]. Some of the leading use cases are discussed next.

(1) Radio Access Network- (RAN-) Aware Content Optimization. Video on demand is one of the services that constitutes half of the mobile network traffic and expected to be 82% of all consumer Internet traffic by 2021, up from 73% in 2016 [59], while at the same time the available network capacity varies by an order of magnitude within seconds, as a result of fluctuations in the radio channel condition. In the case of rapidly varying channel conditions, the transmission control protocol (TCP) may not be able to adapt fast enough, leading to underutilization of radio resources. To overcome this problem, researchers in [29, 32] proposed to use MEC technology to inform the video server of the optimal bit rate to use given the channel state conditions. The idea behind this strategy is to make use of analytics applications to estimate the throughput at the radio downlink interface for a user and then use packet headers to convey that information to the video server, so that it can adapt the stream quality accordingly, at runtime.

(2) Edge Video Orchestration. From the traffic characteristics observed in [60], it is observed that, during the time of a big event, some of the BSs are underutilized. Thus, large public venues are good candidates for MEC, especially where localized venue services are important in exchange for BS energy savings within the network. In this use case, the video from the big event, for example, a soccer or baseball match, can be served to on-site consumers from a MEC server running an appropriate application located, for example, within the stadium premises. This type of service can be linked with a dedicated RAN deployment at the venue such that MEC servers can be colocated with RAN controllers and the backhaul equipment. Edge orchestration requires that the recorded video be locally stored, processed, and directly delivered to the users at the event without backhaul connection to a centralized EPC. This enables the virtualization of the EPC functions in the edge servers for fast data delivery.

(3) Geolocation. The availability of geolocation applications in mobile devices can enhance device tracking for service delivery as Geoanalytics application hosted in the MEC server can make use of the real time information provided by the network about the direction and location of the mobile devices. Such MEC applications will enable location based services for enterprises and consumers, for example, in venues, retail locations, and coverage areas where GPS coverage is not available. Moreover, geolocation can assist in monitoring the health of individuals while on transit and in case of emergency the device can send a report signal to the nearest MEC platform for emergency assistance.

Energy Efficiency. To address power management challenges in the context of virtualized data centers, either large or micro, the authors in [28] proposed a new power management technique called "soft scaling," in addition to "hard scaling" where the processor frequency is either scaled down/up depending on the workload [61]. The idea behind soft scaling is to mimic hardware scaling by allocating reduced time periods for resource usage by each VM, using a VM scheduler. A new architecture is presented by the authors together with a two-tier policy approach, local and global policies, to be applied on the available resources within the virtualized data center. At global level, the system is responsible for coordinating VM migration (e.g., live migration), where VM migration refers to a mechanism for migrating one VM from its local server to another target server, within the data center. The process at local level involves performing actions corresponding to resource management procedures enforced by the controller. From the obtained results in [28], it is shown that the combination of hard and soft scaling may yield better energy savings. Nonetheless, it has to be noted that the VMs are unaware of the power quantity they consume, due to lack of power metering capabilities in virtualized platforms. To provide visibility into VM power consumption, the authors in [62] propose a mechanism for VM power metering. They infer energy consumed from resource usage by each VM and then develop individual system component power models. From the obtained results, it is observed that hard scaling is not suitable for virtualized environments since it affects the performance of all running VMs in the server. The authors advocate for VMs power capping in order to eliminate undesirable "noisy neighbors effect," which is prevalent in cases where sufficient isolation is lacking. In [63], the authors propose an online algorithm, based on Lyapunov optimization, for reducing the operational cost and satisfying carbon neutrality, within a data center, without future workload information. In this work, the server is allowed to autonomously tune its processing speed and to decide on the amount of workload it can compute per time slot. Despite the efforts made towards reducing power consumption in data centers, as presented in [28, 61, 62], new management approaches are required for handling energy consumption in computing platforms as virtualization technology evolves, in addition to the advent of MEC and ultra dense networking (UDN).

With MEC being part of 5G network plans, energy efficiency is also important in such computing environments, as the microplatforms host virtualized mobile functions, and the edge devices are empowered with computing and storage capabilities to serve user's requests locally. This requires edge devices tuning and soft scaling the VMs running on top of the hypervisors in order to improve energy efficiency in virtualized platforms. In addition, the dense deployment of MEC servers requires redesigning energy efficient procedures as with edge computing BSs will provide computing services apart from radio access services. In [64], the authors try to address the issue of incorporating MEC into dense cellular networks, where each small cell BS is considered to have a computing platform colocated with it, and then propose an online algorithm for jointly managing offloading and BS sleep modes decision making, while keeping energy consumption low. Furthermore, they developed a decentralized algorithm for BS-BS cooperation in order to optimize sleep modes and offloading decisions. To optimize the average delay cost, the proposed online algorithm makes use of the Lyapunov optimization technique.

However, to address 5G use cases in a more energy efficient way, visibility into power usage is required for developing power management policies in virtualized computing platforms. The computing platforms can either utilize on-site renewable energy, off-site renewable energy, grid power, or a combination of the sources depending on one's demands. Therefore, quantifying power usage can yield better power management policies in data centers, resulting in energy efficient 5G networks. The reader is referred to [65] for a comprehensive review regarding VM power metering, including server models, sampling, VM power metering methods, and the accuracy of the methods.

2.5. Summary. Different strategies have been employed in the design of the afore-discussed softwarized EPC architectural proposals, as we summarize in Table 1. The main driving force behind softwarization resides in the creation of more flexible, fully reprogrammable networks, which are expected to better handle the diverse and high volume mobile traffic that is foreseen in future networks. The architectural proposals that were investigated so far involve grouping the virtualized EPC network functions [16–18] and using NFV techniques for the EPC cloud [20, 21]. Furthermore, it is shown that network management can be facilitated by the logical instantiation of a network, referred to as *network slice*. This allows a complete separation of network portions, which can be specialized to different purposes. Typical network slices consist of integrated SGW and PGW and the MME as proposed in [14, 24, 26]. With the emergence of MEC, virtualized EPC NFs can be deployed in close proximity to the BSs and this makes it possible to deliver services within a short space of time. Recently, many delay sensitive applications are emerging, and this has resulted in an increase in computation demand, which exceeds what mobile devices can deliver. Therefore, application-aware cell performance optimization for each device in real time is desirable, as this can improve the customer's QoE. MEC has emerged to enable data processing locally, at the network edge, and edge devices are empowered with computing and storage capabilities to serve user requests by significantly reducing the transmission latency, as they are placed in proximity of the end users. While many algorithms are being proposed, a considerable amount of research still has to be carried out and MEC optimization strategies need to be further investigated, compared, and experimented in real systems. In Table 2, we provide a summary of the strategies proposed for handling MEC optimization problems. In [41, 43, 46–48, 52, 54, 58] different strategies have been proposed towards offloading decisions by taking into account different constraints. The numerical results in these papers all demonstrate that MEC can help optimize the energy efficiency of future mobile networks.

Table 1: Summary of EPC architecture proposals.

Softwarized EPC architectures utilizing SDN and NFV technologies
(1) Grouping EPC functions [16–18]
(2) NFV enabled network cloud for EPC [20, 21]
(3) Networking slicing [14, 24, 26]
(4) Mobile edge computing [31, 34, 35]

The proposed EPC architectural designs exhibit some similarities regarding energy efficiency (EE) improvements for future networks. The shared similarities relate to enabling sleep modes at low traffic periods [17, 64] and dynamic scaling of resources within the virtualized domain [21, 28, 61, 63]. In the architecture of Section 2.1, through content caching [17], at RAN and EPC, EE can be improved as content transmission and delay, bandwidth usage can be reduced, and in the architecture of Section 2.2 dynamic scaling of resources can yield improved energy savings [21]. The lack of slice manager in the network slicing architecture (Section 2.3) as observed in [15, 26] results in less efficient resource utilization, and if present, efficient resource management can be achieved over the shared infrastructure as unused resources can be disabled, thus improving EE in virtualized platforms. Lastly, in Section 2.4, the combination of soft [28] and hard [61] scaling can improve the EE, while at the same time using sleeping modes and cooperation can

Table 2: Summary of MEC optimization strategies.

Optimization problem	Optimization technique
EE computation offloading under tasks delay constraints [43]	Maximum cardinality bin packing problem
EE multiuser resource allocation [41]	Dual decomposition, partial Lagrange
Joint computation and communication cooperation [48]	Lagrange duality, ellipsoid method
Optimal offloading and autoscaling decision making in EH powered MEC systems [46]	Markov decision processes and reinforcement learning
Computation and offloading execution cost minimization for MEC systems [47]	Lyapunov optimization
Mobility aware offloading decision [52]	Mobility patterns and heuristics
Minimizing computation and offloading costs under intermittent connectivity [54]	Markov decision processes
Maximizing edge computation performance under energy constraint [58]	Lyapunov optimization and multi-armed bandits

also yield EE improvements [64]. EH can also be integrated with EE policies, reducing the carbon footprint of network deployments. Since future EPC architectural designs will consist of a virtualized infrastructure, understanding VM power metering is key towards reducing the power consumption in virtualized data centers. This enables the design and implementation of new power management algorithms that improve EE within the network.

3. Machine Learning and Data Mining for 5G

The specific benefits of softwarization, network densification, and energy harvesting must be combined in a timely and efficient way, according to the system requirements, in order to achieve high gains. For these reasons, application of network optimization tools such as *machine learning* and *data mining*, in combination with *context-aware techniques*, is of importance, as they are expected to lower the network management costs and enable network wide intelligence and automation, resulting in self-organized networks. These issues are discussed next.

3.1. Machine Learning. Future MNs are expected to learn the diverse characteristics of users behavior, as well as renewable energy source(s) variations, in order to autonomously and dynamically determine good system configurations. Network elements are expected to rely on sophisticated learning and decision making procedures, for an efficient network management. Machine learning (ML) techniques constitute a promising solution for network management and energy savings in cellular networks. According to [66, 67], they can be categorized as *supervised*, *unsupervised*, or *reinforcement learning*-based. Supervised and unsupervised learning, respectively, indicate whether the samples from the dataset are labelled or not. Reinforcement learning instead considers an agent surrounded by a generally unknown environment, whose actions are either rewarded or punished according to a reward/cost model. The final objective is to teach the agent to solve a certain task. *Semisupervised* learning is also possible [68], although less explored: with it, the learner has access to a small amount of labelled data and to a large number of unlabeled examples. This is pretty common in practical cases and, as such, it may eventually become the preferred learning technology for many application domains. The basic concept of ML algorithms and the corresponding applications according to the category of supervised, unsupervised, and reinforcement learning is presented in [66].

ML can be used in modeling various technical problems for next-generation systems. For example, the authors in [46] use reinforcement learning-based resource management algorithm to incorporate renewables into a mobile edge computing platform. The algorithm learns on the fly the optimal policy for dynamic workload offloading and edge server provisioning to minimize the long term cost, that is, service delay and operational cost. A significant learning rate was achieved, from the use of the online learning algorithm with a decomposition of (offline) value iteration and reinforcement learning. Also, an increase in runtime performance was observed when compared with the Q-learning algorithm. An ML based routing preplan solution for an SDN environment is presented in [69], considering (i) flow feature extraction, (ii) user requirement prediction, and (iii) route selection. Under the SDN route planning context, the core idea is to predict the user's business requirements and then plan ahead and set up routing policies, with a view of reducing delay effects. To improve the effectiveness of SDN routing, relevant features were extracted from the user's historical data and then utilized within a semisupervised clustering algorithm for data classification. Through the extraction of user's data, flow, and data plane load features, the flow service demand forecasts were then predicted using supervised classification. The network structure was constantly updated by optimizing the extreme learning method (ELM), and old data was discarded upon completion of the training. In addition, based on the flow feature extraction and flow demand forecasts, a personalized route selection mechanism based on policy making was introduced (the path computation involves the use of linear programming and a set of cost functions).

Another learning technique inspired by a behavioral psychological concept, in the context of machine learning, is called metacognitive scaffolding [70]. The concept of scaffolding theory [70], a prominent tutoring theory for a student to learn a complex task, is based on the fact that the learning process of human beings is metacognitive in nature, since it involves *what-to-learn*, *how-to-learn,* and *when-to-learn*. Such metacognitive learning approach can be extended to cellular networks, more especially to remote access networks, where the network can learn the environment behavior and then adjust its network configuration with respect to the observed and forecast changes. The authors in [71] use a metacognitive scaffolding learning approach for identifying (predicting) tool failures before they occur, in the context of manufacturing processes. The conducted experimental studies, using real-world data, show that the prediction accuracy can be improved using a low complexity algorithm. Such technique can be also used in remote 5G sites (off-grid) to manage the BSs utilizing only renewable energy sources, solar or wind, under traffic variation and battery constraints. This can yield better edge network management, in the case of MEC.

In conclusion, ML is concerned with the design and development of means that allow network devices to *learn*. ML applications in 5G networks involve channel estimation or detection, spectrum sensing, cell/user clustering, handover among HetNets, energy modeling and prediction, user behavior analysis (including mobility and traffic profiling/forecasting), intrusion (fault/anomaly) detection, channel selection association, and so forth. The family of supervised learning techniques relies on known labels, it can then be applied to higher-layer applications such as discovering the mobile user's location and behavior. This can assist in improving the QoE being offered. Unsupervised learning makes use of the input data to automatically discover patterns, and it can be utilized for load balancing in HetNets. Lastly, reinforcement learning relies on a dynamic iterative learning and decision making process. Its recent evolution consists of combining it with deep neural networks, leading to the so-called *deep reinforcement learning* (DRL), which is becoming a standard technique in modern learning systems [72]. RL and DRL can be used for inferring mobile users policies under unknown network conditions, for example, BS association under unknown ESI of the BS, in EH networks. The reader is referred to [66, 67] for a comprehensive review of ML techniques applied to cellular networks.

3.2. Data Mining. The process of turning raw data into useful information, such as discovering patterns in large datasets, is referred to as data mining (DM). Due to the large amounts of high quality data available in the mobile industry, mobile operators can learn about the behavior of their customers and develop effective marketing strategies. The datasets that can be used to this end include *call detail records*, which contain detailed information for each call made, and mobility traces, that is, the sequence of serving cells.

In mobile communication systems, due to the convergence of user behavior, there usually exists some typical scenarios which exhibit different traffic patterns, for example, stadium, campus, special (or big) events, and central business district (CBD). Therefore, accurate traffic scenario recognition and analysis are expected to lead to more efficient resource management and better QoE provision. The authors in [73] used the Louvain method [74] (a widely used method for detecting communities in large networks) for recognizing and analyzing the typical scenarios in wireless cellular networks. In this paper, a modularity optimization based method is used to discover the community structure, where researchers utilize previously measured spatial-temporal wireless traffic datasets. The obtained experimental results show that the proposed method can acquire satisfactory performance in traffic scenario recognition and analysis, which can lead to the development of efficient resource allocation schemes.

3.3. Context-Awareness. A system is said to be context-aware if it uses context to provide relevant information and/or services to its customers, where relevancy depends on the user's task; see [75]. Context information can be divided into two groups: (i) network context and (ii) user context. The former describes the status of the network, for example, type and position of devices, activity status, energy consumption, capacity, and current load. Such information is operator-related and it can be obtained via the backhauling/EPC network. Then, the user context information is the information on the user profile in terms of mobility and service requirements. This information is used for resource allocation, for example, the position and quality of the available channels to be allocated to the user [76].

The authors in [76] offer an overview of context management in future wireless networks, and they focus on developing energy efficient resource management policies and using user position information for channel quality prediction. The considered strategies are based on radio signal strength intensity (RSSI) estimation obtained through propagation models and measurements provided by mobile terminals. To produce accurate RSSI estimates, a fingerprinting approach [77] using power maps is proposed, whereby shadowing, fading, and non-line-of-sight (NLOS) effects are captured. The power maps are constructed by storing the RSSI measurements for a terminal located at a certain position during an offline phase. To increase the estimation accuracy, the fingerprints were collected under various environmental conditions (time of day, weather conditions, etc.). Although this approach provides accurate databases, it is time consuming and expensive. An improvement to it involves mobile users in the creation and maintenance of the power map; that is, users with positioning capabilities (e.g., GPS) report their position and RSSI observations to a database.

The collection of data from a limited number of physical (hardware) and virtual (software) sensors, towards the development of network solutions, is of great importance. However, as the number of devices being introduced into the mobile space increases, it becomes difficult to process all the data that is collected. Therefore, *context-awareness computing* can play a crucial role in deciding what data needs to be collected and processed, acting as a filtering block. Such allows for the storage of context information linked to any piece of data, so that its interpretation can be done easily and be effective. From the context information, context reasoning is applied to deduce new knowledge and better understanding based on the available context [78]. The need for reasoning is due to the imperfection and uncertainty characteristics of raw context. In addition, context reasoning techniques can also be classified, similar to ML, into supervised and unsupervised. If supervised, training data is first collected and then labelled according to the expected results. A function that can generate the expected results using the training data is derived. Such learning techniques can be used for activity recognition, for example, public commuters and big events. Unsupervised techniques find hidden structures in unlabelled data. Since training data is not used, there is no error or reward signal involved in the evaluation of a solution. This technique can be used for situations where possible outcomes are unknown, such as the detection of unusual user behavior.

Discussion. The network elements of future mobile networks should be appropriately configured with the aid of learning techniques. Operators can employ ML to exploit user, network, and mobile traffic datasets to better understand their subscriber base and to analyze network traffic for network management procedures. They may also apply these approaches to boost services or to identify why users do not adopt them.

The applications of ML and DM techniques, coupled with context-awareness, are expected to be a point of strength of MEC platforms for edge network management. A possible model for information retrieval and processing is as follows. Through the RNIS function, the radio network data and other real time context information can be exposed to authorized MEC applications for network management purposes. In addition, the user location services (LOC, which provides UE location features) can obtain user mobility patterns at runtime and then share them with authorized applications for handover management procedures. Having the network visibility, combined with observed mobility patterns, proper resource planning can be achieved resulting in an efficient network. If the network service charge can be effected at the MEC platform, through a charging function (or application) linked with the TOF, traffic load statistics can be obtained and then, ML tools can be utilized to understand the traffic patterns within the network edge, thus availing adaptive resource allocation, for example, adaptive BS power transmission for conserving energy and MEC servers provisioning.

We finally stress that while it is clear that ML, DM, and context-awareness will be among the most crucial elements of 5G and beyond-5G mobile networks, the current unavailability of mobile traffic datasets obtained from operators limits the progress towards research aimed at building effective management algorithms based on them.

Energy Efficiency. Different machine learning frameworks have been proposed for imitating human intelligence, model complex relationships between inputs and outputs, extract statistical structure, and then identify patterns in observed data [79]. In the context of mobile networks, mainly in the EPC, network and user behavior prediction is required for designing efficient management strategies for computational resource allocation, content caching, improving quality of service and reliability. Application of network optimization tools is important towards EE improvement. The context and social information can be used to enable mobile edge caching and computing, thus reducing delay and in-network data traffic (content transmission), as well as improving EE in virtualized environment through VM consolidation and switching-off idle servers [46]. To realize optimal strategies for EE, ML tools can be used to predict users content request distributions, mobility patterns, and content request frequency. In addition, they can be used to learn about computing center workloads variation, and then allocate computing resources/tasks based on the learned information to minimize computation durations. This avails energy savings opportunities, as user behavior can be learned in advance, and then allocates the required computational resources to serve the predicted/inferred quantity of mobile users. The authors in [80] propose a new architecture where content popularity is estimated by applying machine learning tools in the harnessed big data. They also show how this architecture can enable caching at the edge and yield backhaul offloading gains, thus leaving some network nodes inactive. For example, deep spiking neural networks (SNNs) as part of ML optimization tools are shown to possess a potential for improving latency and EE through event-based computation [81]. The authors in [81] introduce a novel technique for differentiating spike events using error backpropagation mechanism. In this work, the membrane potentials of spiking neurons are treated as differentiable signals and discontinuities as noise at spike time. Using this strategy, it is shown that the proposed method surpassed all previously reported results. Despite the promising strategies towards network learning procedures and usage, the access to big data, mainly *call detail records*, is still a limiting factor towards energy efficient procedures utilizing ML tools.

4. Challenges and Open Issues

Next, we discuss some challenges and open issues that are key to the development of efficient, self-adaptable, and fully softwarized networks.

4.1. EPC Related

4.1.1. EPC Control Strategies for BS Energy Management and Big Data Analytics. The power consumption in BSs can be minimized by considering strategies that allow tuning their energy consumption to the network load. For example, by adapting the transmission power in relation to the offered traffic demand [82, 83]. This can overcome the radio access network architecture limitations which is provisioned to handle maximum expected network load, yet for each BS the load fluctuates over time due to different usage patterns and mobility. To enable power transmission adaptation and load balancing across BSs, traffic load information (operator mobile *traffic datasets*) obtained from the EPC can be utilized to extract relevant demand patterns to design dynamic BS management mechanisms [84–86]. Optimization based on mobile traffic datasets from the operator, obtained from multiple network elements, will make it possible to minimize the amount of time it takes to steer traffic on a real time basis. These datasets hold a very promising potential when it comes to mobile network system analysis, as they can capture the traffic dynamics over time and space, at unmatched scales. Their use is expected to pave the way towards very efficient and self-adaptable network solutions.

4.1.2. Deployment and Nature of the EPC Control Logic. The lack of a centralized controller for eNBs is discussed in [87], and it is noted that the absence of such may result into an inefficient radio resource control. Virtualization has introduced the use of a centralized controller exploiting SDN in the EPC; however, this approach still lacks efficient power consumption management strategies, as it does not take into account daily traffic variations. Another management strategy that has to be investigated entails the use of a *distributed* approach [88], where some decisions may be made locally (e.g., at the BSs) to configure instructions remotely within a centralized controller. For example, in the case where the MEC platform is colocated with the BS, it can collect traffic statistics, make local decisions (thus ensuring fast reaction to data traffic and mobility variations), and then share them with the central controller when required, for example, when a BS has to be deactivated. The traditional SDN architecture is centralized [89], with the controller assuming the management responsibility for processing the entire network load. However, as the number of devices receiving network services increases, the processing load also increases. Thus, a centralized controller can potentially be a point of failure, despite its large computation and communication capacity. The placement of SDN controllers has been investigated in [90], where authors provide an analysis about the determining factors that have to be taken into account. This analysis provides useful guidance for SDN operators and application designers, although there are currently no placement rules that apply to every network. Also, the EPC is aware of the aggregate traffic load per BS, but it does not have any knowledge about the energy side information (ESI) in the BSs (energy consumed, harvested, etc.). The lack of such information motivates the need for mobility and *energy-aware* procedures that can be deployed in the EPC, either in a centralized SDN controller or within subcontrollers (SC) as proposed in [88]. These shall perform mobility management tasks, while at the same time being aware of the available energy in the (energy harvesting) BSs. This will ease the energy management tasks in the BSs, by taking into account the current and forecast traffic load.

4.1.3. EPC Procedures for Disaster Management. During natural disaster, it is important to provide support for highly mobile field communication and to possibly deploy mobile networks on the spot in a short time (as the standard telecom infrastructure may be down), for example, to provide coverage and network access for rescue teams. Towards this, an isolated E-UTRAN [91] operation (IOPS) BS may be a good option. This requires agile strategies to manage the IOPS enabled-BS along with colocated virtualized EPC functions. Achieving energy saving in such situation is of utmost interest when renewable energy sources are also utilized, and the network has to be operated off-grid.

4.2. MEC Related

4.2.1. Resource Reservation for Scheduling in MEC Systems. MEC server scheduling requires the user offloading priority order, channel gains, and latency requirements and these depend on the user's location and mobility pattern. Usually, the user's statistical profile consists of varying information as the user traverses the mobile network. This (time varying) information can enhance the design of adaptive servers that regenerate the scheduling order from time to time. To efficiently schedule mobile devices tasks, we can accurately predict users mobility profiles and channel state information, while at the same time using the GPS location trajectories to identify the nearest MEC server(s). Another alternative is to reserve resources that can enhance the server scheduling performance taking into account forecast computation offloading requests.

4.2.2. Green Large Scale MEC Systems. MEC servers are expected to be deployed close to end users, along with small cells, and this will result in dense deployments, which still raise the concern of energy consumption in wireless systems. Since MEC servers are dimensioned for high speed computation, they still consume energy even during their idle state, similar to the always on approach of eNBs, and this calls for the development of means that will allow the servers to consume energy in a way that is proportional to their computational load [92]. Another option to save energy in idle MEC servers is to adopt sleep modes (which have also been proposed as an energy saving strategy for BSs). However, such approach might degrade the user's QoE; therefore proper workload variation prediction strategies have to be considered. In this way, it can be determined which MEC servers are to be deactivated and which ones are to remain active to handle mobility with minimal impact for the users. Moreover, by exploiting the spatial diversity in the workload patterns, MEC servers can coordinate to serve mobile users according to their location [38]. This load balancing approach can help improve the energy efficiency of the lightly loaded edge servers. Finally, we observe that, despite the presented benefits, in renewable energy powered MEC systems, ESI and CSI are still required for decision making; thus the MEC system must be able to acquire this information from the RNIS.

4.2.3. MEC Server Selection for Computation. Designing EE computational offloading mechanisms for MEC networks is a challenge, as mobile devices have to decide where to offload their task under wireless channel quality fluctuations. The selection strategies that have been proposed so far [43] aim at minimizing the energy consumption of the offloading system, by taking into account the cost associated with *task computing* and *data transmission* (depending on the multiaccess characteristics of 5G systems). The proposed algorithm optimizes energy consumption for task offloading, under latency constraints. However, this problem can be extended by allowing mobile devices to acquire knowledge of the computational power of the MEC servers within their proximity, in addition to the above energy costs. This is expected to result in energy efficient MEC server selection that will also take the availability and the resources of the servers into account.

4.2.4. Mobility Management in MEC Networks. Mobility management has been extensively investigated assuming a single mobility anchor per mobile network, which becomes a single point of failure. Distributed mobility anchors can also be deployed within different geographical areas, so that when a mobile device crosses the boundary, handover (HO) requests are triggered to guarantee high data rate and low bit error rate. However, these policies cannot be directly applied to MEC systems with moving users, since they neglect the effects of the computation resources at edge servers on the HO policies. In current mobile networks, HO is handled as follows: when the mobile node signal strength becomes insufficient (weak), HO is initiated either by the mobile node (mobile-initiated HO) or by the network (network-initiated HO). The current serving BS communicates with neighbor BS, target BS, via the X2 interface. In the meantime, IP packets are buffered to minimize packet loss. Once the signaling is over, the mobile node is handed over to the target BS, and a new link is formed. Mainly, the goal of the mobility management protocol is to minimize packet loss and to select the target BS for the UE; that is, only the HO target (BS to connect to) is considered in this case. In MEC systems, computational capacity, current workloads, computational costs, and energy side information are crucial in determining the HO target (offloading target in this case). Thus, it is important to extend HO and mobility management procedures to MEC networks, keeping these additional variables into account. The mobility patterns observations in [52–55] have shown that it is possible (and sensible) to make use of the observed information in designing MEC mobility management procedures. However, this introduces the use of distributed databases to keep and process the observed mobility information, to extract relevant patterns. This might pose a challenge, as it might add to the delay constraints, as decision making will rely on drawing conclusions from the observed states and to being able to predict the device movement around the edge.

5. Conclusions

In this article, we have provided an analysis of existing softwarized mobile network architecture proposals, discussing the benefit that softwarization will bring about, along with some open research questions. We have also presented the new mobile edge computing (MEC) architectural concept, which entails bringing computation, caching, and applications closer to the network edge. Such paradigm will transform the edge network into an intelligent service hub, capable of delivering personalized services at the edge when the MEC platform is colocated with the base stations or in their close proximity. Furthermore, we have outlined the challenges and open issues related the MEC platform for future mobile networks. From our perspective, the combination of energy efficient techniques, energy harvesting capability for MEC systems, and the virtualization of EPC network functions can yield improved network management, while availing dynamically reconfigurable networks with improved performance, adaptability to new traffic conditions, and easy network reconfiguration.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was funded by the European Union Commission under SCAVENGE Project no. 675891-H2020-MSCA-ITN-2015/H2020-MSCA-ITN-2016.

References

[1] M. Casado, T. Koponen, S. Shenker, and A. Tootoonchian, "Fabric: A retrospective on evolving SDN," in *Proceedings of the 1st ACM International Workshop on Hot Topics in Software Defined Networks, HotSDN 2012*, pp. 85–89, Finland, August 2012.

[2] 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 Communications Surveys & Tutorials*, vol. 16, no. 3, pp. 1617–1634, 2014.

[3] G. Lu, C. Guo, Y. Li et al., "ServerSwitch: A Programmable and High Performance Platform for Data Center Networks," in *Proceedings of the in USENIX Conference on Networked Systems Design and Implementation*, Boston, Mass, USA, 2011.

[4] N. McKeown, T. Anderson, H. Balakrishnan et al., "OpenFlow: enabling innovation in campus networks," *Computer Communication Review*, vol. 38, no. 2, pp. 69–74, 2008.

[5] M. Xia, M. Shirazipour, Y. Zhang, H. Green, and A. Takacs, "SOLuTIoN: SDN-based OpticaL traffic steering for NFV," in *Proceedings of the 3rd ACM SIGCOMM 2014 Workshop on Hot Topics in Software Defined Networking, HotSDN 2014*, pp. 227-228, USA, August 2014.

[6] M. Ciosi, D. Clarke, C. Cui et al., "Introductory white paper: network functions virtualization," in *Proceedings of the SDN and OpenFlow World Congress*, Darmstadt, Germany, 2012.

[7] Y. Wu, Y. Chen, J. Tang et al., "Green transmission technologies for balancing the energy efficiency and spectrum efficiency trade-off," *IEEE Communications Magazine*, vol. 52, no. 11, pp. 112–120, 2014.

[8] N. Bhushan, J. Li, D. Malladi et al., "Network densification: the dominant theme for wireless evolution into 5G," *IEEE Communications Magazine*, vol. 52, no. 2, pp. 82–89, 2014.

[9] G. Fettweis and E. Zimmermann, "ICT energy consumption-trends and challenges," in *Proceedings of the in 11th International Symposium on Wireless Personal Multimedia Communications*, Lapland, Finland, 2008.

[10] D. Zordan, M. Miozzo, P. Dini, and M. Rossi, "When telecommunications networks meet energy grids: cellular networks with energy harvesting and trading capabilities," *IEEE Communications Magazine*, vol. 53, no. 6, pp. 117–123, 2015.

[11] S. Sesia, M. Baker, and I. Toufik, *LTE-the UMTS Long Term Evolution: From Theory to Practice*, John Wiley & Sons, Hoboken, NJ, USA, 2nd edition, 2011.

[12] B. Rinor, “Virtualization of the Core Network of LTE,” Tech. Rep., Telecommunications Research Group, Berlin, Germany, 2014.

[13] Docker for the Virtualization Admin, 2016, https://www.docker.com/what-container.

[14] “5G Mobile Communications Systems for 2020 and beyond,” Fifth generation mobile promotion forum, Japan, 2016.

[15] A. Nakao, D. Ping, and K. Masayuki, “Flare: Deeply programmable network node architecture,” EU-Japan Collaboration Project, University of Tokyo, Japan, 2016, http://www.5gsummit.org/berlin/docs/slides/Aki-Nakao.pdf.

[16] H. Hawilo, A. Shami, M. Mirahmadi, and R. Asal, “NFV: state of the art, challenges, and implementation in next generation mobile networks (vepc),” *IEEE Network*, vol. 28, no. 6, pp. 18–26, 2014.

[17] C. Yang, Z. Chen, B. Xia, and J. Wang, “When ICN meets C-RAN for HetNets: An SDN approach,” *IEEE Communications Magazine*, vol. 53, no. 11, pp. 118–125, 2015.

[18] X. Jin, L. E. Li, L. Vanbever, and J. Rexford, “SoftCell,” in *Proceedings of the the ninth ACM conference*, pp. 163–174, Santa Barbara, California, USA, December 2013.

[19] Open vSwitch, http://openvswitch.org/.

[20] “Unlock New Business Opportunities With vEPC,” Tech. Rep., VMware, Palo Alto, Calif, USA, 2016.

[21] P. K. Agyapong, M. Iwamura, D. Staehle, W. Kiess, and A. Benjebbour, “Design considerations for a 5G network architecture,” *IEEE Communications Magazine*, vol. 52, no. 11, pp. 65–75, 2014.

[22] The Xen Project Powers, https://www.xenproject.org/.

[23] “ETSI GS NFV-INF 004: Network Functions Virtualisation (NFV); Infrastructure; Hypervisor Domain,” Tech. Rep., ETSI, Sophia-Antipolis, France, 2015.

[24] A. Nakao, “Software-defined data plane enhancing SDN and NFV,” *IEICE Transactions on Communications*, vol. E98B, no. 1, pp. 12–19, 2015.

[25] M. Fukushima, Y. Yoshida, A. Tagami, S. Yamamoto, and A. Nakao, “Toy block networking: Easily deploying diverse network functions in programmable networks,” in *Proceedings of the 38th Annual IEEE Computer Software and Applications Conference Workshops, COMPSACW 2014*, pp. 61–66, Sweden, July 2014.

[26] M-CORD Open Reference Solution Paves the Way for 5G Innovation, 2016, http://opencord.org/tag/m-cord/.

[27] H. Zhang, N. Liu, X. Chu, K. Long, A. Aghvami, and V. C. Leung, “Network Slicing Based 5G and Future Mobile Networks: Mobility, Resource Management, and Challenges,” *IEEE Communications Magazine*, vol. 55, no. 8, pp. 138–145, 2017.

[28] R. Nathuji and K. Schwan, “VirtualPower: coordinated power management in virtualized enterprise systems,” in *Proceedings of the 21st ACM SIGOPS Symposium on Operating Systems Principles, SOSP’07*, pp. 265–278, Stevenson, Wash, USA, October 2007.

[29] M. Patel, Y. Hu, P. Hédé et al., “Mobile edge computing introductory technical white paper,” Tech. Rep., ETSI, Sophia-Antipolis, France, 2014.

[30] K. M. S. Huq, S. Mumtaz, J. Bachmatiuk, J. Rodriguez, X. Wang, and R. L. Aguiar, “Green HetNet CoMP: Energy Efficiency Analysis and Optimization,” *IEEE Transactions on Vehicular Technology*, vol. 64, no. 10, pp. 4670–4683, 2015.

[31] “Vision on Software Networks and 5G SN WG,” The 5G Infrastructure Public Private Partnership (5G-PPP) consortium, Portugal, 2017.

[32] B. Gabriel, “Mobile edge computing use cases and deployment options,” Heavy Reading, 2016.

[33] ETSI GS MEC: Mobile Edge Computing (MEC); Framework and Reference Architecture, ETSI, Sophia-Antipolis, France, 2016.

[34] ETSI GS MEC 002: Mobile Edge Computing (MEC); Technical Requirements V1.1.1, ETSI, Sophia-Antipolis, France, 2016.

[35] “A Universal Access Gateway for Fixed and Mobile Network Integration,” Convergence of Fixed and Mobile Broadband Access/Aggregation Networks (COMBO) Consortium, France, 2016.

[36] S. Sudevalayam and P. Kulkarni, “Energy harvesting sensor nodes: Survey and implications,” *IEEE Communications Surveys & Tutorials*, vol. 13, no. 3, pp. 443–461, 2011.

[37] S. Ulukus, A. Yener, E. Erkip et al., “Energy harvesting wireless communications: a review of recent advances,” *IEEE Journal on Selected Areas in Communications*, vol. 33, no. 3, pp. 360–381, 2015.

[38] M. Lin, Z. Liu, A. Wierman, and L. L. H. Andrew, “Online algorithms for geographical load balancing,” in *Proceedings of the 2012 International Green Computing Conference, IGCC 2012*, USA, June 2012.

[39] A. Costanzo, M. Dionigi, D. Masotti et al., “Electromagnetic energy harvesting and wireless power transmission: a unified approach,” *Proceedings of the IEEE*, vol. 102, no. 11, pp. 1692–1711, 2014.

[40] J. Garnica, R. A. Chinga, and J. Lin, “Wireless power transmission: From far field to near field,” *Proceedings of the IEEE*, vol. 101, no. 6, pp. 1321–1331, 2013.

[41] C. You, K. Huang, and H. Chae, “Energy efficient mobile cloud computing powered by wireless energy transfer,” *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 5, pp. 1757–1771, 2016.

[42] Z. Chang, J. Gong, Y. Li et al., “Energy Efficient Resource Allocation for Wireless Power Transfer Enabled Collaborative Mobile Clouds,” *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 12, pp. 3438–3450, 2016.

[43] K. Zhang, Y. Mao, S. Leng et al., “Energy-efficient offloading for mobile edge computing in 5G heterogeneous networks,” *IEEE Access*, vol. PP, no. 99, 2016.

[44] L. Bonati, A. F. Gambin, and M. Rossi, “Wireless power transfer under the spotlight: Charging terminals amid dense cellular networks,” in *Proceedings of the 2017 IEEE 18th International Symposium on " World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, pp. 1–9, Macau, China, June 2017.

[45] Y. Shi, J. Zhang, B. O’Donoghue, and K. B. Letaief, “Large-scale convex optimization for dense wireless cooperative networks,” *IEEE Transactions on Signal Processing*, vol. 63, no. 18, pp. 4729–4743, 2015.

[46] J. Xu and S. Ren, “Online learning for offloading and autoscaling in renewable-powered mobile edge computing,” in *Proceedings of the 59th IEEE Global Communications Conference, GLOBECOM 2016*, USA, December 2016.

[47] Y. Mao, J. Zhang, and K. B. Letaief, “Dynamic Computation Offloading for Mobile-Edge Computing with Energy Harvesting Devices,” *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 12, pp. 3590–3605, 2016.

[48] X. Cao, F. Wang, J. Xu, R. Zhang, and S. Cui, “Joint Computation and Communication Cooperation for Mobile Edge Computing,” https://arxiv.org/abs/1704.06777, 2017.

[49] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, Cambridge, UK, 1st edition, 2004.

[50] S. Boyd, “Ellipsoid method, Stanford Univ., Stanford, CA, USA,” 2014, https://stanford.edu/class/ee364b/lectures/ellipsoid_method_notes.pdf.

[51] J. W. Chinneck, B. Kristjansson, and M. J. Saltzman, *Operations Research and Cyber-Infrastructure*, Operations Research/Computer Science Interfaces Series, Springer, New York, NY, USA, 1st edition, 2009.

[52] K. Lee and I. Shin, “User mobility model based computation offloading decision for mobile cloud,” *Journal of Computing Science and Engineering*, vol. 9, no. 3, pp. 155–162, 2015.

[53] W. Su, S. Lee, and M. Gerla, “Mobility prediction in wireless networks,” in *Proceedings of the IEEE Military Communications Conference (MILCOM’00)*, pp. 491–495, Los Angeles, Calif, USA.

[54] Y. Zhang, D. Niyato, and P. Wang, “Offloading in Mobile Cloudlet Systems with Intermittent Connectivity,” *IEEE Transactions on Mobile Computing*, vol. 14, no. 12, pp. 2516–2529, 2015.

[55] C. Wang, Y. Li, and D. Jin, “Mobility-assisted opportunistic computation offloading,” *IEEE Communications Letters*, vol. 18, no. 10, pp. 1779–1782, 2014.

[56] M. J. Neely, “Stochastic network optimization with application to communication and queueing systems,” *Synthesis Lectures on Communication Networks*, vol. 7, pp. 1–199, 2010.

[57] P. Auer, N. Cesa-Bianchi, and P. Fischer, “Finite-time analysis of the multiarmed bandit problem,” *Machine Learning*, vol. 47, no. 2-3, pp. 235–256, 2002.

[58] J. Xu, Y. Sun, L. Chen, and S. Zhou, “ E 2 M 2 : Energy efficient mobility management in dense small cells with mobile edge computing ,” in *Proceedings of the ICC 2017 - 2017 IEEE International Conference on Communications*, pp. 1–6, Paris, France, May 2017.

[59] Cisco visual networking index: Forecast and Methodology, 2016–2021, Cisco, San Jose, Calif, USA, 2017.

[60] J. Erman and K. Ramakrishnan, “Understanding the super-sized traffic of the super bowl,” in *Proceedings of the the 2013 conference*, pp. 353–360, Barcelona, Spain, October 2013.

[61] J. S. Chase, D. C. Anderson, P. N. Thakar, A. M. Vahdat, and R. P. Doyle, “Managing energy and server resources in hosting centers,” in *Proceedings of the 18th ACM Symposium on Operating Systems Principles*, Alberta, Canada, October 2001.

[62] A. Kansal, F. Zhao, J. Liu, N. Kothari, and A. A. Bhattacharya, “Virtual machine power metering and provisioning,” in *Proceedings of the 1st ACM Symposium on Cloud Computing (SoCC ’10)*, pp. 39–50, June 2010.

[63] S. Ren and Y. He, “COCA: Online distributed resource management for cost minimization and carbon neutrality in data centers,” in *Proceedings of the 2013 International Conference for High Performance Computing, Networking, Storage and Analysis, SC 2013*, Denver, Colo, USA, November 2013.

[64] L. Chen, S. Zhou, and J. Xu, “Energy efficient mobile edge computing in dense cellular networks,” in *Proceedings of the ICC 2017 - 2017 IEEE International Conference on Communications*, pp. 1–6, Paris, France, May 2017, https://arxiv.org/abs/1701.07405#.

[65] C. Gu, H. Huang, and X. Jia, “Power metering for virtual machine in cloud computing-challenges and opportunities,” *IEEE Access*, vol. 2, pp. 1106–1116, 2014.

[66] C. Jiang, H. Zhang, Y. Ren, Z. Han, K.-C. Chen, and L. Hanzo, “Machine Learning Paradigms for Next-Generation Wireless Networks,” *IEEE Wireless Communications Magazine*, vol. 24, no. 2, pp. 98–105, 2017.

[67] P. V. Klaine, M. A. Imran, O. Onireti, and R. D. Souza, “A survey of machine learning techniques applied to self organizing cellular networks,” *IEEE Communications Surveys & Tutorials*, vol. PP, no. 99, pp. 1–40, 2017.

[68] O. Chapelle, B. Schölkopf, and A. Zien, *Semi-Supervised Learning*, MIT Press, Cambridge, Mass, USA, 2006.

[69] F. Chen and X. Zheng, “Machine-Learning Based Routing Pre-plan for SDN,” in *Multi-disciplinary Trends in Artificial Intelligence*, vol. 9426 of *Lecture Notes in Computer Science*, pp. 149–159, Springer International Publishing, Cham, Switzerland, 2015.

[70] D. Wood, “Scaffolding, contingent tutoring, and computer-supported learning,” *International Journal of Artificial Intelligence in Education*, vol. 12, pp. 280–293, 2001.

[71] M. Pratama, E. Dimla, C. Y. Lai, and E. Lughofer, “Metacognitive learning approach for online tool condition monitoring,” *Journal of Intelligent Manufacturing*, 2017, https://arxiv.org/pdf/1705.02477.

[72] V. Mnih, K. Kavukcuoglu, D. Silver et al., “Human-level control through deep reinforcement learning,” *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[73] Z. Yi, Y. Peng, T. Wang, X. Zhang, and W. Wang, “Traffic scenario recognition and analysis for wireless cellular system: From social network perspective,” in *Proceedings of the 2016 IEEE Canadian Conference on Electrical and Computer Engineering, CCECE 2016*, Canada, May 2016.

[74] V. D. Blondel, J. Guillaume, R. Lambiotte, and E. Lefebvre, “Fast unfolding of communities in large networks,” *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2008, no. 10, Article ID P10008, 2008.

[75] G. D. Abowd, A. K. Dey, P. J. Brown, N. Davies, M. Smith, and P. Steggles, “Towards a better understanding of context and context-awareness,” in *Handheld and Ubiquitous Computing: First International Symposium, HUC ’99 Karlsruhe, Germany, September 27–29, 1999 Proceedings*, vol. 1707 of *Lecture Notes in Computer Science*, pp. 304–307, Springer, Berlin, Germany, 1999.

[76] A. Redondi, I. Filippini, and A. Capone, “Context management in energy-efficient radio access networks,” in *Tyrrhenian International Workshop on Digital Communications-Green ICT (TIWDC)*, Genoa, Italy, September 2013.

[77] M. Bshara, U. Orguner, F. Gustafsson, and L. van Biesen, “Fingerprinting localization in wireless networks based on received-signal-strength measurements: a case study on WiMAX networks,” *IEEE Transactions on Vehicular Technology*, vol. 59, no. 1, pp. 283–294, 2010.

[78] M. Mühlhäuser, A. Ferscha, E. Aitenbichler, and D. Plexousakis, “A survey of semantics-based approaches for context reasoning in ambient intelligence,” in *Proceedings of the European Conference on Ambient Intelligence*, Darmstadt, Germany, 2007.

[79] C. Mingzhe, C. Ursula, S. Walid, Y. Changchuan, and D. Mérouane, “Machine Learning for Wireless Networks with Artificial Intelligence: A Tutorial on Neural Networks,” https://arxiv.org/abs/1710.02913, 2017.

[80] E. Zeydan, E. Bastug, M. Bennis et al., “Big data caching for networking: Moving from cloud to edge,” *IEEE Communications Magazine*, vol. 54, no. 9, pp. 36–42, 2016.

[81] J. H. Lee, T. Delbruck, and M. Pfeiffer, “Training deep spiking neural networks using backpropagation,” *Frontiers in Neuroscience*, vol. 10, article 508, 2016.

[82] Sustainable Energy use in Mobile Communications, Ericsson, Stockholm, Sweden, 2015.

[83] E. Oh, K. Son, and B. Krishnamachari, "Dynamic base station switching-on/off strategies for green cellular networks," *IEEE Transactions on Wireless Communications*, vol. 12, no. 5, pp. 2126–2136, 2013.

[84] E. Oh, B. Krishnamachari, X. Liu, and Z. Niu, "Toward dynamic energy-efficient operation of cellular network infrastructure," *IEEE Communications Magazine*, vol. 49, no. 6, pp. 56–61, 2011.

[85] 3GPP TS 32.2.298 version 13.5.0 Release 13, Charging management; Charging Data Record (CDR) parameter description, ETSI, Sophia-Antipolis, France, 2016.

[86] J. Fourie, Realizing real-time charging, Ericsson, Stockholm, Sweden, 2006.

[87] X. Jin, L. E. Li, L. Vanbever, and J. Rexford, "SoftCell: Scalable and flexible cellular core network architecture," in *Proceedings of the 2013 9th ACM International Conference on Emerging Networking Experiments and Technologies, CoNEXT 2013*, pp. 163–174, USA, December 2013.

[88] D. Basu, A. A. Hussain, and S. F. Hasan, "A distributed mechanism for Software-based mobility management," in *Proceedings of the 7th IEEE International Conference on Software Engineering and Service Science, ICSESS 2016*, pp. 321–324, China, August 2016.

[89] G. Araniti, J. Cosmas, A. Iera, A. Molinaro, R. Morabito, and A. Orsino, "OpenFlow over wireless networks: Performance analysis," in *Proceedings of the 2014 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting, BMSB 2014*, China, June 2014.

[90] B. Heller, R. Sherwood, and N. McKeown, "The controller placement problem," in *Proceedings of the 1st ACM International Workshop on Hot Topics in Software Defined Networks, HotSDN 2012*, pp. 7–12, Finland, August 2012.

[91] 3GPP TR 123 401 V14.4.0, General Packet Radio Service (GPRS) enhancements for Evolved Universal Terrestrial Radio Access Network (E-UTRAN) access, ETSI, Sophia-Antipolis, France, 2017.

[92] L. A. Barroso and U. Hölzle, "The case for energy-proportional computing," *The Computer Journal*, vol. 40, no. 12, pp. 33–37, 2007.