Connecting the IoT – Opportunities for Using SDN and NFV to Realize Flexible LTE Network Architectures

Benedikt Christoph Wolters
benedikt.wolters@rwth-aachen.de

ABSTRACT

With the steady and rapid increase of network traffic, service innovation and pressure to reduce costs mobile operators nowadays face severe challenges while their currently deployed architectures are hardware-centric, monolithic and inert to upcoming innovation. Among the current applications that use the mobile operator's network, the introduction of the Internet of Things (IoT) yields new challenges by the adaption of Machine Type Communication (MTC) applications. Those new applications comprise of machine type devices potentially high in number. The sheer plurality of devices may lead to network congestion due to high signaling messages. Network Function Virtualization (NFV) aims to operate network services on virtualized environments that can be deployed at locations in the network as required. The concept of Software Defined Networking (SDN) provides a abstraction layer between data-plane forwarding and control-plane. The emergence of SDN and NFV allow mobile network operators to control and dimension their networks according to their traffic demands within a fine-grained granularity. Furthermore, these concepts steer to potential cost-savings that avoid over- and under-utilization of resources. Additionally, more flexibility, adaptivity, and faster establishment of network innovations is promised.

In this paper we briefly highlight the challenges mobile operators face with respect to their Long Term Evolution (LTE) infrastructures. Subsequently, we then elucidate four proposed architectural concepts that utilize the SDN paradigm or the concept of NFV at different levels. These new architectures were designed to allow mobile operators to cope with those new challenges by introducing flexibility, programmability and a better resource utilization to their mobile carrier networks.

1. INTRODUCTION

Throughout the course of history the Internet has continuously experienced rapid growth, currently being a network with over 9 billion devices [21]. In recent years especially mobile carrier networks have gained a lot of importance [17] in connecting mobile users throughout the world to the Internet. Momentarily on the way towards the fifth generation of mobile broadband networks it is expected that traffic demand will constantly-rise [5]. The present mobile cellular networks are capable of offering high data rates, integrating more and more devices and ensuring a high Quality of Experience (QoE). Nevertheless, while mobile network operators are steadily trying to satisfy their projected rapid increasing traffic demands [17, 5], further challenges lay ahead with the emergence of the IoT [12]. A large portion of this estimated traffic increase is expected to be assigned to connecting communicating machines to the Internet over mobile networks (e.g., sensors or actuators). There are a multitude of possible practical applications for those Machine to Machine (M2M) or MTC scenarios, e.g., home automation, natural disaster prevention, industrial automation, energy saving, or, healthcare monitoring [3]. The number of MTC devices is potential large, estimates exceeding 60 billions M2M connections by 2020 [20]. Hence supplementary revenue could be generated for mobile operators to cope with their relatively stagnant current average revenue per user [15, 20]. Albeit, connecting an enormous number of IoT devices through mobile carrier networks will impose a large load on the current network infrastructure that can lead to congestion or even failure, while negatively impacting the QoE of regular mobile users [18].

Apart from the tenfold-increase in traffic [6], mobile operators need to address multiple other requirements such as increasing energy-efficiency, resource-efficiency, spectrum-efficiency and cost-efficiency. However, the deployment of new network services and features has high costs for the integration as well as the operation, since network functions typically come with separate hardware entries. This monolithic, expensive and tightly integrated networking equipment is complex in optimal configuration and maintenance as well as troubleshooting [15]. Mobile carrier equipment is throughout highly standardized by organizations such as the Third Generation Partnership Project (3GPP). Although this specification process is favorable, it hinders early adopters of innovative technologies to await long standardization processes being complete until vendors start delivering the desired equipment, thus leading to long time-to-market periods for the operator [10, 15, 2] for offering new services. But market competition requires fast adoption and deployment as well as a certain elasticity in changing service requirements dynamically [2, 15]. At the same time operators face an "end-of-profit", where the cost to build and maintain a network are exceeding the revenue [17, 14].

Additionally, the current inflexible networking architectures prevent the research community from developing new paradigms with real-world networking equipment. Meanwhile, operators may end up in vendor lock-ins, where their entire network infrastructure must originate from a specific vendor for maximum efficiency [15].

Besides this, even with the proper networking equipment installed certain networking elements might undergo periods of network over- and underutilization. Mobile opera-

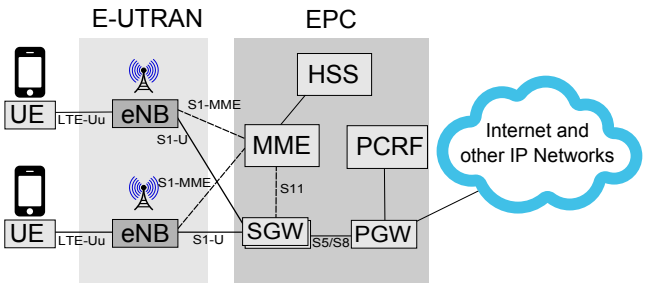

Figure 1: The Evolved Packet Core High-Level Architecture. Multiple UEs are connected through their associated eNBs. The eNB tunnel their traffic through the SGW and PGW, respectively. The MME is coordinating the handover and mobility processes[19].

tors solve this issue by over-provisioning their infrastructures leading to an inefficient use of the available resources, ultimately resulting in decreasing revenue for the mobile network operator [9, 1, 2].

Additionally, the state-of-the-art hardware-centric architectures centralize data-plane networking functions such as monitoring, access control, and Quality of Service (QoS). This leads to high capital expenditures, e.g., a centralized Cisco packet gateway costs six million dollars) [10]. Beyond that, the networking equipment suffers from complex decentralized control-plane protocols. A change in routing can require several networking elements to reconfigure (e.g, update of routing tables). While the reconfiguration process will eventually converge, it still leaves room for misrouted packets [8].

All of this leads to a reduction in the operators revenue or a limited QoE. To cope with those challenges two concepts are currently in the focus of academia [16]: Network Function Virtualization (NFV) and Software Defined Networking (SDN). In this paper we will focus on LTE mobile networks and present four different approaches proposed from the academic community leveraging those concepts to allow for a more flexible and scalable mobile network architecture.

Organization

The rest of the paper is structured as follows: In Section 2 we provide basic background information. Subsequently, in Section 3 we will describe the selected networking architectures and discuss the approaches in Section 4. Finally, in Section 5 we conclude our work.

2. BACKGROUND

In this section, we provide basic background information required to grasp the concepts of the latter introduced architectural approaches.

2.1 Long Term Evolution (LTE)

Since we restrict our focus to architectures that deal within the LTE [19] context, we describe the LTE high-level architecture in the following. LTE is the latest evolution of 3GPP's mobile network standard. According to the specification LTE supports peak downlink rates of 300 Mbps and peak uplink rates of more than 75 Mbps. The LTE system supports a scalable bandwidth from 1.4 MHz up to 20 MHz, thus allowing operation on lower frequency ranges. Furthermore, LTE supports a novel all-IP network that does not – as contrary to the 2G/3G-predecessors – fall back to a circuit-switched architecture design. The architecture of LTE – the Evolved Packet System (EPS) – is comprised of the Evolved Packet Core (EPC) and the air interface E-UTRA as depicted in Figure 1. The User Equipment (UE), typically a mobile phone or a tablet, connects via the radio access network E-UTRAN to the evolved NodeB (eNB) radio station. The eNB is connected to the EPC, which we describe in the following.

Evolved Packet Core (EPC)

The evolved packet core is an aggregation network for forwarding the mobile users traffic to an Packet Data Network Gateway (PDN-Gateway). The EPC has interoperability features for interacting with legacy 2G and 3G network services. The EPC acts as a unifying routing fabric in the core network used by different 3GPP-standardized radio technologies as well as non-3GPP access technologies such as IEEE 802.11, among others. The functions of the EPC are:

- Aggregating traffic from different fixed and mobile access points to a single Internet gateway router.
- Managing mobility of the user equipment between the base stations. The management of mobility is crucial to ensure packet network connectivity when a device switches from base station to another base station.
- Manage bandwidth and congestion in order to provide better QoS for applications such as voice messaging. This is necessary since the resources in wireless networks are severly constrained in how much bandwith for an UE is available.
- Handle Authentication, Authorization and Accounting (AAA) of user traffic.

After the UE connected to the eNB, the traffic is directed through a serving gateway (SGW) over the GPRS Tunneling Protocol (GTP). The corresponding SGW is a local mobility anchor, i.e. if the user switches from local eNB to another the SGW is not changed and the communication is not interrupted. Thus the SGW handles frequent changes of user's location, and stores a large amount of state since users retain their IP addresses when they move. Additionally, the SGW tunnels traffic to the Packet Data Network Gateway (PGW). The PGW enforces quality-of-service policies and monitors traffic to perform billing. The PGW also connects to the Internet and other cellular data networks, and acts as a firewall that blocks unwanted traffic. Policies at the PGW can be very fine-grained, based on whether the user is roaming, properties of the user equipment, usage caps in the service contract, parental control, etc.

Aside from the previously mentioned data-plane functionality that involves tunneling the user's traffic, the eNBs, SGWs, and PGWs also are in charge of some control-plane functions: By coordinating with the Mobility Management Entity (MME), they perform signaling to handle session setup, teardown, and reconfiguration, as well as mobility.

For example, when an UE requests to have a dedicated session setup, the PGW will send QoS and other session

information to the SGW. Thereupon the SGW will notify the MME, which will then contact the eNB to etablish the needed resources and setup the connection to the UE. Another example of control-plane interaction is the handover of an UE between two eNBs. The UE sends measurement reports to its associated eNB. If the eNB discovers the target eNB in the measurement report, and recognizes a high signal strength, it initiates a handover decision. If the eNB moves from a source eNB to a target eNB the source eNB will send a handover request to the target eNB. If the handover request was acknowledged, the target eNB will notify the MME that the UE is now in the area of responsibility of the source eNB. Additionally, the target eNB notifies the source eNB to release resources [10]. The SGW and PGW are also involved in routing, running decentralized routing protocols such as OSPF.

The Policy Control and Charging Function (PCRF) manages charging at the PGW domain. Furthermore, the PCRF also provides the QoS authorization that decides how to treat each traffic flow, based on the user's subscription profile. QoS policies can be dynamic, e.g., based on time of day. This must be enforced at the PGW.

The HSS is a centralized database that contains user-related and subscription-related information. The functions of the HSS include functionalities such as mobility management, call and session establishment support, user authentication and access authorization. The HSS is based on Home Location Register (HLR) and Authentication Center (AuC) that is a relict from the 3GPP's LTE predecessors. The HSS contains subscription information for each user, such as the QoS profile, any access restrictions for roaming, and the associated MME. In times of cell congestion, a base station reduces the max rate allowed for subscribers according to their profiles, in coordination with the P-GW.

We recall from Figure 1 that in EPC, the SGW and PGW are user plane elements, while, e.g., the MME or the PCRF are purely control plane elements.

The user's traffic is encapsulated by GTP or PMIP through the core network. This encapsulations help operators support IP mobility in low-latency, higher data-rate, all-IP core networks that support real-time packet services over multiple access technologies. LTE was designed to support mobility between multiple Radio Access Networks, both legacy 3GPP networks such as 2G/3G as well as so called non-3GPP networks such as WiFi [19, 15].

The EPS is a connection-oriented transmission network, which requires a "virtual" connection between two endpoints to be established (e.g., the UE and the PGW), before any data can be sent between those endpoints. This virtual connection is called a bearer. A bearer is characterized by the two endpoints that connect (e.g., UE and eNB), a set of QoS attributes, that describe the type of service (e.g., voice, video stream, best effort QoS etc.), a flow specification that describes the guaranteed and maximum bitrate filter specification that describes the traffic flows (in terms of IP addresses, protocols, port numbers, etc.) for which the transport service is provided between the two endpoints. An EPC bearer is composed of two parts, namely the S1 bearer and the S5/S8 bearer. The S1 is responsible of the traffic between the eNB and SGW [19, 15]. The S5/S8 bearer are for the traffic between the SGW and PGW. If the UE becomes inactive, the S1 bearer is released to save radio link resources. However, the S5/S8 bearers remain active to provide a so called 'always-on' feature. When an UE becomes active again, a corresponding service request is re-establishing the S1-U bearer. Additional service requests can be performed to instantiate a dedicated bearer if extra quality parameters are needed. E.g., a new S1-U and S5/S8 bearers would be created that support the higher demand service [15, 18].

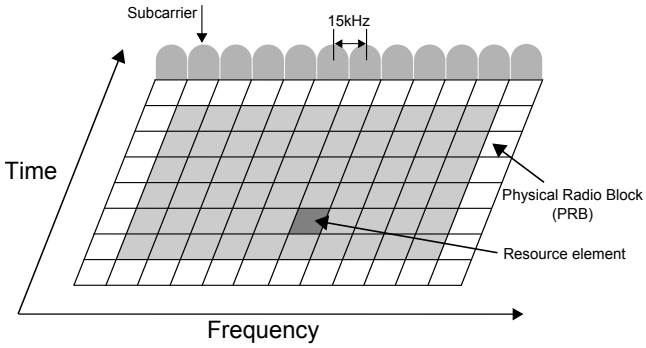

Figure 2: An OFDM Physical Radio Block (PRB) which is comprised of multiple Resource elements, which contain OFDM symbols. In OFDMA PRBs are distributed amongst clients [22].

eNodeB Air Interface

In order to reduce the peak-to-average ratio and increase the efficiency of the power amplifier and save battery life of the UE, LTE uses different access modes for uplink and downlink. In the downlink OFDMA is used, while Single Carrier FDMA (SC-FDMA) is utilized in the uplink. For brevity, we only focus on OFDMA: OFDMA is a multi-user version of orthogonal frequency-division multiplexing (OFDM) digital modulation scheme. The transmitted signal is modulated onto different orthogonal frequencies, so called sub-carriers, in parallel. Since the sub-carriers are send in parallel the carrier can be send at a lower symbol rate, which makes OFDM much more robust against interference or attenuation. Multiple access is achieved in OFDMA by assigning different sub-carriers to different users. This allows simultaneous low data rate transmission from several users [7, 22]. In the LTE medium access control the smallest unit for the Medium Access Control (MAC) scheduler are so called Physical Resource Blocks (PRB), which consists of seven OFDMA symbols and twelve sub-carriers as depicted in Figure 2 [22]. The data-rate for a user is dependent on the modulation scheme that is used for the OFDMA symbols, e.g., QPSK, 16QAM, or 64QAM [22].

2.2 Software Defined Networking (SDN)

Software-defined networking (SDN) [13] is an emerging innovative network architecture paradigm. SDN introduces a new layer of abstraction into the networking architecture, which decouples the data-plane of a network from the control-plane. A logically centralized network entity called controller is introduced in a control-plane that is responsible for managing and steering of the arising traffic in the data-plane transportation network. An overview of the decomposition layers in SDN is depicted in Figure 3.

The main idea is that the controller maintains all the intelligence of the network by having a full view on the network as opposed to traditional decentralized routing schemes

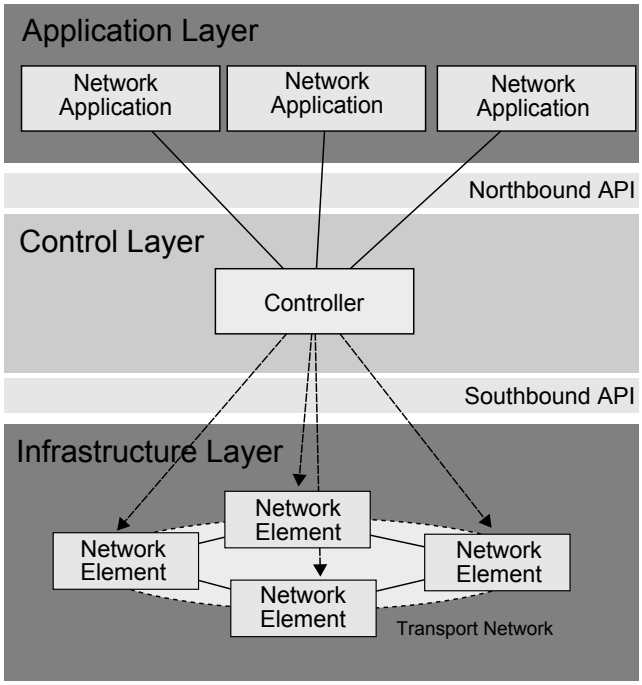

Figure 3: SDN High-Level Architecture: The network is compartmentalized into control-plane and data-plane. The SDN Controller instructs standardized network elements in the data-plane, while networking applications use the northbound API of the controller [6].

such as OSPF/BGP/STP etc., which are operated on single network routers. Hence, the controller is aware of the current network state. Apart from the controller, there are very simple networking elements, which reside in the transportation network (data-plane) and forward the incoming traffic based on rules that are given to them by the controller through a standardized southbound API. This networking elements have little intelligence themselves and are assumed to be standardized cheap common-off-the-shelf (COTS) equipment, that is easily installed and replaceable. The controller features a northbound API that can be utilized by higher level networking applications. For those applications the controller provides an united view of all the resources available to higher level applications. This way networking applications based on the controller's northbound API can be developed that operate on the full network view. E.g., this enables the operator of the network to steer her traffic at specific points which are known to be less utilized than others, thus allowing the operator to utilize the available resources more efficiently. Traditionally, only equipment vendors can modify the software of their hardware-based network elements with new networking services [15]. Hence the introduction of SDN leads to a programmable network, which means novel network applications can be developed very fast and easily through modern agile programming methodologies [6], which leads to a shorter time-to-market period. For example, if an operator wants to experiment with a new routing protocol, without SDN the operator would need to wait for a vendor to upgrade the complete networking equipment to support that routing protocol. However, with SDN the operator can simply deploy a networking application that implements this routing protocol and does not need to update all the specialized networking equipment. This leads to more flexibility in control and innovation.

A widely accepted SDN-enabling protocol is OpenFlow [11] that defines how the control plane can be configured and controlled by the central controller. In traditional networks, switches or routers have forwarding information stored in miscellaneous formats (MAC tables or routing tables) with run complex routing algorithms that operate on these tables. OpenFlow standardizes a single and centralized protocol that can create and manage the flow tables on OpenFlow switches, replacing all other forwarding tables. The data-plane is then fully programmed by the establishment of flow tables on the OpenFlow switches through the OpenFlow controller. Incoming packets in the OpenFlow switch are either handled by existing flow tables that match a packet based on various Level 2 and 3 package headers, or the respective package is send to the controller for further action to decide what happens with the packet.

2.3 Network Function Virtualization (NFV)

A complementary technology to SDN is NFV [4]: NFV is a new paradigm that allows running virtual network functions (VNF), as software, on Virtual Machines (VMs) instantiated on general-purpose hardware rather than on standalone dedicated hardware. In NFV the network function of a device is implemented in a software package. In contrast to traditional server virtualization techniques a VNF, may consist of one or multiple VMs, on top of high volume servers, switches and storage, or even cloud computing infrastructure, instead of having custom hardware installed for each network function. Through the virtualization and the separation of software controlling the network function from the actual hardware machine(s) running the VNF additional flexibility is achieved since the software is easily maintainable and upgradeable [6]. Moreover, orchestration allows the automation of the instantiation, monitoring and reparation of network functions. This allows operators to dynamically create additional VNFs when needed in order to scale with user demands or to migrate a VNF on-the-fly to another machine, e.g., to optimize latency or to pool resources in times of under-utilization. From an economical point of view NFV allows network operators to cut capital expenditures for new specialized single-purpose networking equipment and reduce power consumption, e.g., by moving previously decentralized network functions to centralized data centers. Furthermore, the deployment, backup, testing and creation of new network functions becomes easier since only software needs to be started, deployed or exchanged.

3. PROPOSED NETWORK ARCHITECTURES

In this section, we present a selection of LTE architectures leveraging the concepts of SDN or NFV that have been proposed in the academia. We restrict our focus on those architectures that focus on the IoT and the previously the addressed challenges that were discussed in Section 1.

3.1 Mobile Flow

Pentikousis et al. [15] proposed a software-defined networking approach for mobile carrier networks. They provide an architectural blueprint for implementing current mobile

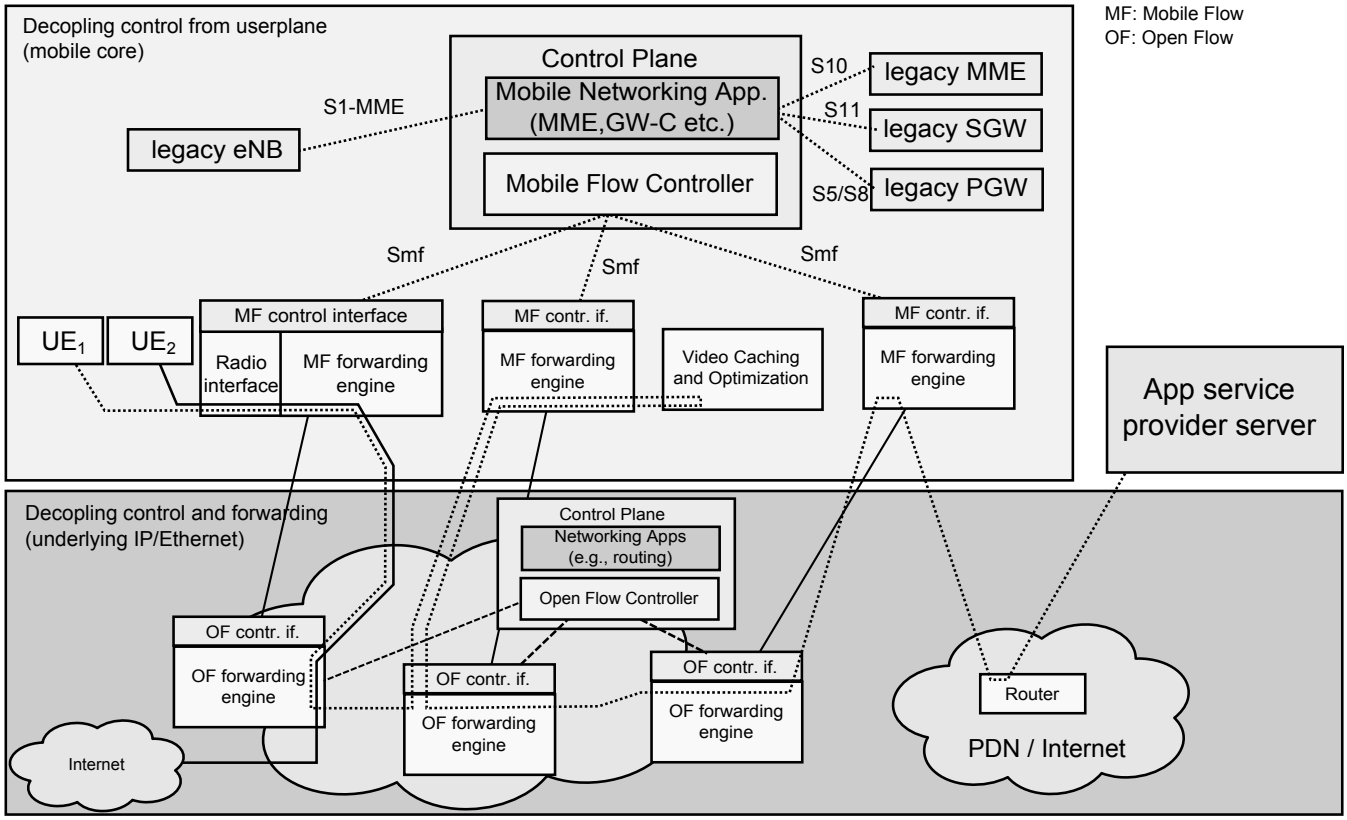

Figure 4: Adapted from [15], MobileFlow introduces a new control plane on top of the to conventional EPC architecture, while moving all the previous control-plane functions to mobile networking applications that run on the MobileFlow Controller. The mobile flow controller instructs MobileFlow Forwarding Engines that controls the flows of an OpenFlow switch. Value-added services (here: video caching) can be interposed by flow rules.

network architectures through SDN. Simultaneously, they claim their architecture is flexible and programmable enough to handle future network innovations.

3.1.1 Additional Controller Stratum

Pentikousis et al. introduce a so called Software Defined Mobile Network (SDMN). The main goal of this network is to have maximum flexibility as well as programmability for the operator in the core network, where instead of deploying specialized-hardware network, the network is comprised of standard IP/Ethernet-interconnected MobileFlow forwarding engines (MFFEs). Those forwarding engines take care of all the user-plane traffic forwarding. The MFFEs are controlled by a MobileFlow controller (MFC). Figure 4 depicts the MobileFlow architecture that reenacts the functions of today's EPC. The mobile flow controller is a layer on top of the forwarding network. The functionality of the classic user-plane SGW and PGW are replaced by MFFEs while control-plane functions such as the MME knowm from EPC are moved to applications that run on the controller level of the MFC. By introducing a generic MFFE, the user-plane becomes very simple, while control-plane applications can run on the controller level in a centralized fashion and utilize the full network view. The control of traffic forwarding is realized in software on the controller instead of hardware as in traditional architectures. This allows the operator to gain additional flexibility and adaptivity. For example the operator may steer the traffic to different service providers based (e.g., deep packet inspection, video caching, lawful interception) on rules defined that can be fully in software and that can quickly be adapted without changes in the hardware and firmware of the forwarding engines [15]. Also this design gives the operator the ability to whether or not to utilize network function virtualization, e.g., the operator can decide on the controller level to use a virtualized firewall that runs on a commodity-server in a datacenter or forward the traffic to a hardware-based firewall instead. Also note that this architecture introduces additional flexibility so that the operator is not bound to the particularities of the LTE architecture. That means the operator can quickly innovate to novel upcoming networking architectures. Furthermore, the controller layer can still interact with legacy hardware-centric network elements that allow the operator to incrementally deploy the new architecture without changing the underlying hadware

3.1.2 MFFE

The MFFEs on the data-plane layer are more complex than a standard OpenFlow switch, but much simpler than a router or a PGW, because the main control-plane functionality has been moved to the MFC. MFFE must support carrier-grade funtionality i.e. Layer 3 GTP/PMIP tunneling

(i.e. en-/decapsulating of IP packets) or flexible charging, which is not supported by convetional OpenFlow [15]. Similar to OpenFlow's flow tables, MFFE receive flow rules that are send via a lightweight control-layer protocol called Smf from the MFC. Based on this flow rules the MFFEs perform packet en-/decapsulation. This feature is necessary, to allow the interoperability of MFFEs with legacy networking equipment. In contrast to a OpenFlow switch an MFFE should be able to handle a multiple of the number of flow tables a OpenFlow switch can handle today. This is to allow future networking innovations not to be restricted in flexibility and being limited by current limitations. The functionality of the MFFE can either be implemented on a trimmed gateway or by extending the functionality of an OpenFlow switch. Furthermore, hybrid approaches are possible as shown in Figure 4, where the OpenFlow controller is extended by MFC functionality as well as MFFEs are combined with OpenFlow switches. Moreover, Pentikousis et al. envision that the MFFE could also be integrated with radio interfaces to allow the management of radio bearers.

3.1.3 MFC and Network Applications

Similiar to a classic OpenFlow controller, the MFC is comprised of a northbound and a southbound interface. The latter is used to interact with the MFFEs through Smf. The former is used by mobile network applications that use network-level abstraction functions (e.g, network resource monitoring) to control the network. However, the MFC features an additional network functions block e.g., GTP tunnel processing, charging and mobility anchoring. These functions can be utilized by the network applications. Additionally, a horizontal interface is deployed for interacting with other MFCs that the operator might run in different areas. It is notable that the MFC does not directly interact with any legacy hardware, e.g., the controller has no direct interface to a legacy eNB or a PGW. Instead, the networking application takes care of all the legacy interfacing, however still the MFFEs are instructed to carry out the respective operations. Hence, novel network applications can be developed on top of the northbound interface that lead to the advantages that were previously discussed in 2.2. For example, mobility management can be realized fully through the northbound interface [15].

3.1.4 Multi-tenant networks

The concept of the architecture allows for multi-tenant networks. That means one or multiple operators can use the same networking equipment in the data-plane, while having completely different networking applications running in the control-plane. This allows for better resource utilization and pooling of network resources. For example, an operator might operate a EPC based networking architecture, where each of the control-plane components such as eNB-C, MME, SGW-C, PGW-C, PCRF, HSS has been fully virtualized. We use *-C here to indicate that we refer to the control-plane specific functions of the corresponding entity running as a network application. Running those EPC network application on a cloud computing infrastructure, the operator simultaneously, might experiment with a novel network application that operates on the same MFFEs but controls a different part of the network. The same way multiple operators might share a common network, while having individual applications running for their clients on the control layer.

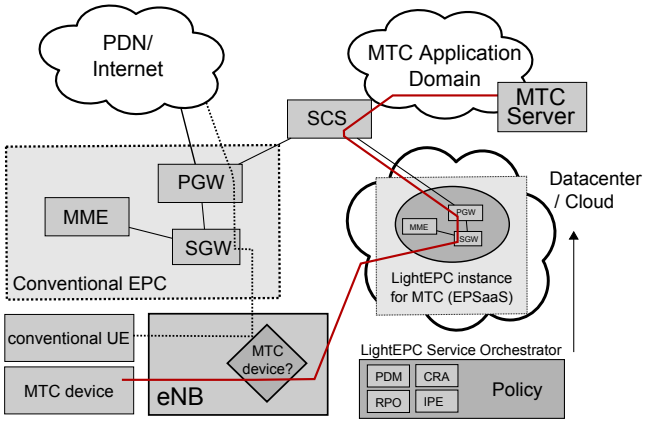

Figure 5: Adapted from [20]: While the standard UE traffic is routed through the conventional EPC architecture, the MTC traffic is intercepted at the eNB and routed through the LightEPC instance through the MTC Server.

The authors evaluated their proposed architecture by implementing a prototype testbed, where they showcase the network programmability and on-demand creation of multiple coexisting mobile architectures (3G, 4G) in the same network.

3.2 LightEPC

Taleb et al. [20] introduced an architecture that tackle the challenges (e.g., congestion or system overload) that arise when connecting multiple MTC attach simultaneously to a mobile network. Their proposed architecture, namely LightEPC, focuses on the orchestration of lightweight virtualized mobile core network instances that run in a cloud computing environment and simplifies the attach procedure of MTC devices for mobile networks. This concept allows operators additional scalability when coping with increasing MTC traffic. The massive increase [20, 12] of MTC in mobile networks imposes a lot of additional pressure on the radio network as well as the core network, resulting in potential congestion or overload. For example, when a multitude of sensors detect the same physical event in a region they all might connect simultaneously to an MTC Server in the Internet through the mobile operator, which might impact the QoE of regular non-MTC users. Another example, is an MTC server that might trigger actions on several MTC devices (e.g., actuators or metering requests), simultaneously. This will yield in the setup of multiple bearers which involve a lot of control plane signaling and thus may overload the network.

3.2.1 Towards Reducing MTC Signaling Overhead

It is expected that MTC devices attach to the LTE network (i.e. the eNB) and exchange data with an MTC server. The MTC servers are connected via a Service Capability Server (SCS) entity to the LTE core network. The MTC Inter Working Function (MTC-IWF), is connected to the SCS and is responsible for authorization of MTC Servers as well as the instruction of MTC devices to initiate a connection with the SCS. Several approaches have been introduced by 3GPP to cope with the problem of reducing the

signaling for small MTC data traffic. A simple solution is to embed the small data of MTC devices into SMS packets. In LTE the delivery of a SMS does not require the establishment of S5/S8 bearers and since the contents of the SMS packet are encapsulated in Non-Access-Stratum (NAS) control-plane messages. Such NAS messages are delivered to the MME and from there forwarded to an SMS-SC, which will take care of the proper routing to the MTC server. However, the packet length of an SMS is very limited for longer MTC data. Another solution is to use a dedicated protocol, namely Small Data Transmission (SDT). The MTC data is then encapsulated into SDT packets which are send over the control channel protocols to the MME. The MME will forward the SDT packets to an MTC-Inner Working Function (MTC-IWF). This entity will deliver the packet to the MTC server. A third solution is to use a combined gateway approach, where the MTC data is send through a special gateway that is different from the conventional PGW/SGW, thus removing the S5/S8 bearers [20] of this gateways to a special gateway.

EPSaaS

Independent from the fact that various mechanisms were proposed to handle MTC traffic, Taleb et al. aim at separating the MTC traffic from regular traffic. Furthermore, the costs for the operator to enter the MTC market should be reduced. In their envisioned architecture they exploit the features of NFV to realize EPS as a Service (EPSaaS). Figure 5 depicts the envisaged architecture. Here, the relevant parts of the networking functions of the EPS Core, are realized by virtual instances that can be run in an operator's datacenter or in a cloud computing environment. The MTC related traffic is then forwarded through this virtualized EPSaaS infrastructure while the regular user traffic is steered towards the original EPS architecture. This enables the operator to utilize the EPSaaS in times where there is a high network load originating from MTC usage, and to use the legacy EPS infrastructure for times for MTC traffic where there is lower utilization. According to Taleb et al. , the EPC network elements that suffer the most from a large number of MTC attach requests are the MMEs as well as the SGWs and PGWs, since here bearers and mobility anchors need to be maintained for a lot of MTC devices simultaneously. The introduction of EPSaaS separates MTC control plane from regular traffic control plane and is expected to lower the resource load on the regular EPS elements. Additionally, there might be MTC devices that only are active within a very specific time-frame, e.g., when periodic metering is performed. With this knowledge the additional resources may only be utilized in this time intervals, leading for a better resource utilization.

Modification of the eNB and SCS

In order to distinguish the traffic of MTC devices from regular traffic and send it to the respective EPSaaS instance the eNB needs to be modified. Taleb et al. propose in their envisioned architecture that eNBs and SCSs are assumed to be equipped with a new function dedicated to detect and identify MTC service types, namely MTC Service Type Detection Function (MTC-STDF). When an MTC device is triggered (i.e. a function of an MTC is requested by an MTC server) or an MTC device wants to signal an event, the MTC devices and/or MTC servers issue signaling messages to connect to the network and/or to trigger MTC devices to attach to the network. In such a scenario the signaling messages are intercepted and analyzed by the MTC-STDF function.

Thus in the proposed architecture the eNBs need to be modified with the ability to distinguish regular traffic from MTC traffic. For the identification part the authors suggest the eNBs may use the hardware identifiers of the MTC devices, e.g., the International Mobile Subscriber Identity (IMSI). Additionally, similar procedures must be established on the SCS. Upon the identification of MTC traffic the MTC-STDF notifies a Policy Enforcement Entity (PEE) entity about the characteristics of the MTC traffic.

The PEE is comprised of four entities: The (i) Policy Decision Making (PDM), which decides whether or not traffic is routed through a LightEPC instance. If the traffic should be routed through the LightEPC instance the (ii) Cloud Resource Assessor (CRA) is informed which takes care of the instantiation of proper cloud resources for the EPSaaS instance. The (iii) Individual Policy Enforcer (IPE) initiates the images of the VNF (such as SGW/PGW) on the VMs. Finally, the (iv) Run-Time Policy Orchestrator (RPO) will observe and adjust the LightEPC architecture during its instantiation by the analysis of MTC traffic patterns as well as available resource monitoring.

With this four entities the PEE controls the life-cycle and the scaling of the instantiated VMs. When changes in the VMs have occurred (e.g., a LightEPC instance was created) the SCS and eNB are informed to route the traffic through the LightEPC instance.

Finally, the authors evaluated their approach. They claim that with an increasing number of MTC devices the signaling (here, the attach request of the MME was evaluated) in the conventional core network is reduced, since the signaling now happens in the virtual LightEPC instance. Due to the ability to flexibly instantiate more LightEPC instances, the increasing amount of additional signaling is removed from the conventional mobile core network and offloaded to the virtual LightEPC instance.

3.3 Virtualized Gateways for MTC

While the previous approaches are concentrated on the use of SDN and NFV in the LTE core network, Samdanis et al. [18] propose a solution that tackles the challenges for MTC traffic on the MTC device level. Considering the potential large amount of MTC devices and the assumption that MTC devices send small and infrequent data, the authors come to the conclusion that need for an individual bearer for each MTC device imposes a lot of overhead, when comparing the small message to the amount of signaling messages to establish the bearer. This overhead may easily cause congestion considering the high number of MTC devices.

Virtual Bearers and Gateway Virtual Machine (GVM)

Samdanis et al. propose a *virtual bearer* solution that leverages the NFV paradigm, to reduce signaling messages for MTC devices. The architecture is depicted in Figure 6. The main idea is that a bearer, that has been established with the conventional methods for an MTC device, is shared amongst a group of MTC devices that have similar characteristics with respect to QoS and are geographically nearby. The shared bearer is established over the radio link, S1 and S5/S8 interface. Within this MTC group, one device then holds a

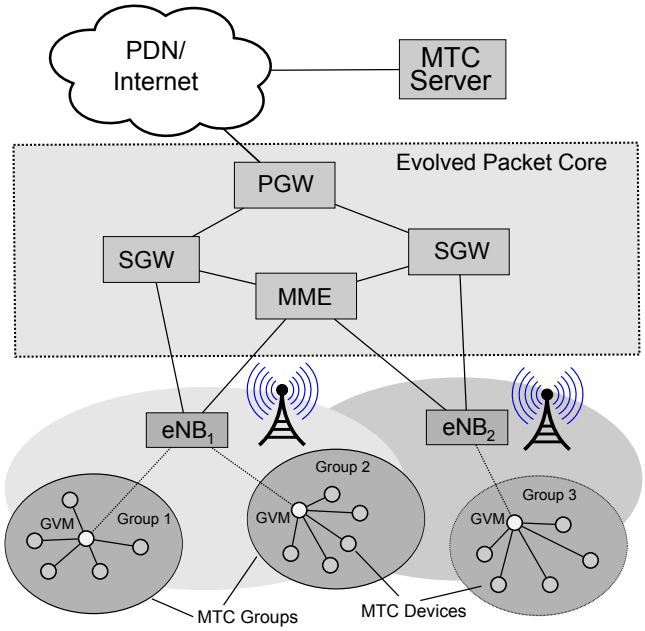

Figure 6: Multiple MTC devices are organized in groups based on their QoS requirements and location. MTC groups are connected via D2D communication, while a single device group member is holding the shared GVM function that ensures LTE connectivity [18].

Gateway Virtual Machine (GVM) function, that holds and uses negotiated bearers to transmit and receive data. The MTC device that holds the GVM function is preferably a device with low mobility and within good coverage of an eNB. The GVM is the network function that allows the group traffic to be treated as it would originate from a single LTE device. The MTC group devices are considered to be able to perform Device-To-Device (D2D) communication, for example by using WiFi-Direct or LTE-Direct. This enables MTC group devices in the vicinity, but out of eNB coverage, to communicate with the group member that holds the GVM function. After they have authenticated to the device that holds the GVM function, it will then redirect in-turn forward the data to the eNB. Members that are in eNB coverage, however, can request the GVM function to be able to send data over the virtual bearer themselves. Bearer related information, e.g., state information, as well as MTC group specific data such as group IMSI, TMSI, IP address and group keys, which any conventional LTE client would need to hold, are stored in the GVM.

However, the information about the presence of a single MTC device is not fully hidden from the core network. The MME is informed about the group information and when each MTC device attaches the network, the MME then suggests which MTC device should act as the holder for the GVM function or even distribute a sequence of devices that want to exchange the GVM function. This way the MME also acts as a scheduling entity.

Migration of GVM to another MTC device

After a device is done transferring and the MME already has provided the next device that should receive the GVM, the MTC device that has finished communicates with D2D technology the GVM configuration to the upcoming MTC device that should receive the GVM function. Is the next device unknown, the releasing device broadcasts that it no longer needs the GVM function and a D2D discovery process is initiated, to find a device that wants to host the GVM functionality next. It is expected that all devices that want the GVM functions respond to this broadcast indicating the urgency of their need for the GVM function. The releasing device would then choose a successor based on this criteria and transfer the GVM configuration to this device, which then will broadcast to all group members that it now holds the GVM functionality. The authors propose two possible ways to realize the transfer of the GVM either the device that has received the GVM informs the MME about the reception of the GVM, the MME then can update its records. After a period of idleness, the eNB may release the shared virtual bearer. The device that currently holds the GVM function in that time will notify all other group members via D2D, so that the next device that wants to use the GVM needs to send the proper signaling messages to re-create the radio and S1 bearer again.

In the author's analysis the proposed scheme of sharing the MTC function amongst group members is compared against a purely event-based and a scheduled coordinated triggering scenario. In case of low MTC network load the scheme introduces additional overhead due to supplementary D2D communication, however, when the MTC load in the network increases the congestion at the eNB is reduced.

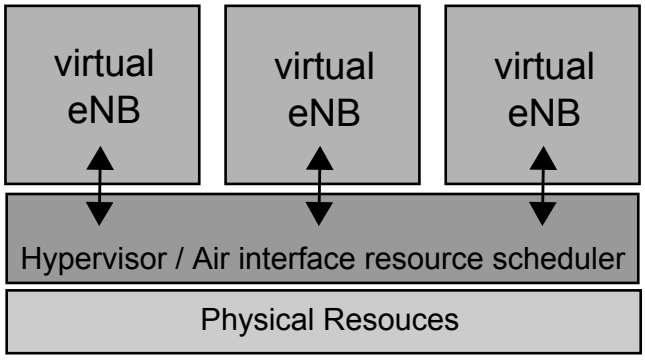

Figure 7: Multiple virtual eNBs are run by multiple operators on a shared eNB [22].

3.4 Wireless Virtualization on eNBs

Zaki et al. [22] investigated the use of NFV to allow operators to share the same physical resources and have coexisting operator networks on the same network infrastructure. They examine the use of a multi-tenant eNB, i.e. the wireless resources at the eNB are allocated amongst multiple (virtual) operators. It is envisioned that the sharing of the wireless resources will lead to a more efficient use of the scarce resources, while simultaneously reducing the total amount of eNBs. Lastly, this would allow novice mobile operators an easier entry to the mobile market. Furthermore, the author's require that this sharing of those resources should be fair respect to used spectrum, power-consumption, and, QoS parameters.

eNB Hypervisor

To allow multiple operators to use the wireless resources provided by the air interface of the eNB, a hypervisor stratum is created on the eNB that allows the operation of virtualized eNBs on top of the hypervisor as depicted in Figure 7. The hypervisor is responsible for the scheduling of real wireless resources between the virtual eNBs. Prioritization is done by complying with contracts that have been defined with the individual operator. Different contract types are defined: **fixed guarantee** where an operator request a fixed bandwidth regardless whether it is used or not, **dynamic guarantee** where an operator is assigned a maximum bandwidth, but only the bandwidth that is actually used is allocated, **best effort** where the operator is assigned a minimum and a maximum bandwidth where the minimum bandwidth is always allocated, **best effort with no guarantees** the operator would only be allocated resources if there are any left to assign.

Resource Scheduling

Based on this contract types the resource scheduler at the hypervisor distributes resources. In LTE those resources are the Physical Resource Blocks (PRB) for OFDMA as introduced in Section 2.1. Each operator running a virtual eNB will periodically send bandwidth-estimations to the hypervisor, that include whether more PRBs are needed or allocated PRBs can be released. With this estimations the hypervisior has the ability to split the resources between the operators. First the operators with fixed guarantee and the minimum bandwidth of best effort operators are assigned. The remainder is distributed across the other operators by a fairness factor, which is based on the ratio of total estimated bandwidth and the desired operator's bandwidth need.

The authors simulated their proposal in a network simulator. Their simulation showed that the dynamic guarantee operators can benefit from potential cost savings due to releasing unused PRBs and the best effort operators can benefit from using those released PRBs.

4. DISCUSSION

The proposed approaches we described differ in their domain of applying the NFV and SDN concepts and are merely comparable. Hence, we discuss each approach individually.

MobileFlow.

The MobileFlow architecture (c.f., Subsubsection 3.1) introduces a general architectural blueprint for maximum flexibility and programmability by exploiting the SDN and NFV paradigm. The proposed architecture allows the operator to innovate to novel controlling models by the introduction of new control plane applications that run on top of a MobileFlow controller, which controls the underlying transportation network. Furthermore, it allows multi-tenancy and co-existing network architectures. The architectural proposal clearly solves the problems of limited hardware-centric flexibility within current mobile networks. Additionally, it solves deployment aspects of novel technology. Moreover, not only the control-plane stratum is exchangeable, but also the transport stratum to switch to newer transport-plane hardware in case of latter innovation.

However, the authors envision that the functionality of the eNB is also controlled by the mobile flow controller, which remains future work. Furthermore, the authors only show a proof-of-concept implementation. Additional overhead in terms of network performance that is introduced by the additional decoupling was not examined. Finally, the challenges in terms of massive traffic increase as well as a rapid increase in new devices such as MTC devices are only partly solved by optimizing the current networks utilization and relying on future networking applications to solve this problems.

LightEPC.

The LightEPC framework (c.f., Subsubsection 3.2) copes with increasing MTC traffic by utilizing NFV and cloud technology to dynamically deploy virtualized network instances of the EPC specifically to reduce the networking load of the conventional EPC core network to handle EPC traffic. The architecture seems to solve the problems operators might face with respect to a large amount of MTC signaling traffic for short transmissions. Due to the introduction of orchestration LightEPC allows operators to save costs by only utilizing the resources currently needed.

However, they introduce additional traffic control logic at the eNB, which might be still overwhelmed by the large amount of MTC traffic. Furthermore, the arising problems are just offloaded to another level, namely the LightEPC instance. While the signaling overload is merely offloaded to a virtual software instance, it is still existent. At the same time introducing more and more virtualized EPC instances running in parallel, might lead to additional management overhead and redundancy, while the underlying technology still remains stale. Furthermore, the additional delay, which is introduced by outsourcing the traffic to the datacenter/cloud is not taken into account. Hence LightEPC seems to be only a temporary incremental approach in coping with the arising MTC challenges.

Virtualized Gateways for MTC.

Konstantinos et al. introduced an architecture for sharing a virtual bearer between a number of MTC devices, as described in Subsubsection 3.3. MTC groups are predefined based on equal quality parameters and use a shared gateway network function to transmit data via a virtual shared bearer. This solution tackles the increasing amount of MTC devices by grouping them and thus reducing the amount of resources that the core network needs to reserve. Although this solution reduces the signaling traffic in the core network it also increases the management and signaling on the layer of the MTC devices through D2D communications, thus indirecting the problem of increasing MTC traffic to more interfering D2D traffic. Furthermore, MTC devices might be small and restricted in resources. Thus this additional management may restrain battery life. Additionally devices within the MTC group may fail or become disconnected, which can lead to complex handover special cases to increase the robustness. In addition MTC-groups must be pre-configured, which restricts application scenarios with high mobility and raises the question why not to use a conventional local gateway e.g. a stationary dedicated point instead.

Lastly, in case of frequent transmission the creation of a MTC group may lead to congestion on the MTC device as it has to wait to receive the GVM making it infeasible for real-time applications.

Wireless Virtualization on eNBs.

Zaki et al. proposed to introduce a new virtualization layer on the eNB to allow operators to pool their eNB resources and thus allowing a one the one hand small operators to gain better coverage and on the other hand existing operators to get a better resource utilization. The proposal introduces an additional virtualization layer on top of the eNB. However, the approach does not solve any of the other issues regarding flexibility or programmability as the overall network architecture remains unchanged. Furthermore, only the downlink is considered, while in case of LTE different up- and downlink radio access technologies coexist.

5. CONCLUSION

In this paper we highlighted the problems that mobile operators currently face with the rapid increase in traffic and additional MTC growth through the Internet of Things. Furthermore, we provided a background insight of the inner workings of LTE and SDN as well as NFV. Finally, we described and discussed distinct approaches that try to tackle with the arising problems through the use of SDN and NFV. We have seen that the use of SDN can introduce additional network programmability, which can lead to future adaptivity to novel networking architectures as well as more flexibility in control in today's deployments. NFV on the other side allows the operator to deploy virtualization techniques at various levels and stages of the network be it running virtual instances of control plane entities in a datacenter, the shared use of a bearer in an eNB or even running network functions on the client devices. Hence, NFV introduces a lot of innovative opportunities for operators to scale their networks accordingly as well as using pooling effects.

Finally, we clearly realize that both technologies are complementary and can accompany the evolution of novel networking architectures in the mobile networking world.

6. REFERENCES

[1] A. Basta, A. Blenk, M. Hoffmann, H. J. Morper, K. Hoffmann, and W. Kellerer. SDN and NFV Dynamic Operation of LTE EPC Gateways for Time-varying Traffic Patterns. In *6th Int. Conf. on Mobile Networks and Management*, 2014.

[2] A. Basta, W. Kellerer, M. Hoffmann, H. J. Morper, and K. Hoffmann. Applying NFV and SDN to LTE Mobile Core Gateways, the Functions Placement Problem. In *Proc. of the 4th Workshop on AllThingsCellular*. ACM, 2014.

[3] M. Chen, J. Wan, and F. Li. Machine-to-machine communications: Architectures, standards and applications. *KSII Trans. on Internet and Inf. System*, 6(2), 2012.

[4] M. Chiosi, D. Clarke, J. Feger, C. Cui, J. Benitez, U. Michel, K. Ogaki, M. Fukui, D. Dilisle, I. Guardini, et al. Network Functions Virtualization: An Introduction, Benefits, Enablers, Challenges and Call for Action. In *SDN and OpenFlow World Cong.*, 2012.

[5] Cisco. Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2013-2018. *Cisco Public Information*, 2014.

[6] P. Demestichas, A. Georgakopoulos, D. Karvounas, K. Tsagkaris, V. Stavroulaki, J. Lu, C. Xiong, and J. Yao. 5G on the Horizon: Key Challenges for the Radio-Access Network. *IEEE Vehicular Technology Magazine*, 8(3), 2013.

[7] H. Holma and A. Toskala. *LTE for UMTS-OFDMA and SC-FDMA Based Radio Access*. John Wiley & Sons, 2009.

[8] J. Kempf, B. Johansson, S. Pettersson, H. Luning, and T. Nilsson. Moving the mobile Evolved Packet Core to the cloud. In *Wireless and Mobile Computing, Networking and Comm., 2012 IEEE 8th Int. Conf. on*. IEEE, 2012.

[9] A. Khan, W. Kellerer, K. Kozu, and M. Yabusaki. Network sharing in the next mobile network: TCO reduction, management flexibility, and operational independence. *Comm. Mag., IEEE*, 49(10), 2011.

[10] L. E. Li, Z. M. Mao, and J. Rexford. Toward software-defined cellular networks. In *Software Defined Networking (EWSDN), 2012 European Workshop on*. IEEE, 2012.

[11] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner. Openflow: enabling innovation in campus networks. *ACM SIGCOMM Computer Communication Review*, 38(2), 2008.

[12] D. Miorandi, S. Sicari, F. De Pellegrini, and I. Chlamtac. Internet of things: Vision, applications and research challenges. *Ad Hoc Networks*, 10(7), 2012.

[13] B. Nunes, M. Mendonca, X.-N. Nguyen, K. Obraczka, and T. Turletti. A survey of software-defined networking: Past, present, and future of programmable networks. *Communications Surveys Tutorials, IEEE*, 16(3), Third 2014.

[14] M. Page, M. Molina, and J. Gordon. The mobile economy (2013), 2013.

[15] K. Pentikousis, Y. Wang, and W. Hu. Mobileflow: Toward software-defined mobile networks. *Comm. Mag., IEEE*, 51(7), 2013.

[16] J. Qadir, N. Ahmed, and N. Ahad. Building programmable wireless networks: An architectural survey. *CoRR*, abs/1310.0251, 2013.

[17] M. R. Sama, S. Ben Hadj Said, K. Guillouard, and L. Suciu. Enabling network programmability in lte/epc architecture using openflow. In *WiOpt, 2014 12th Int. Symp. on*. IEEE, 2014.

[18] K. Samdanis, A. Kunz, M. I. Hossain, and T. Taleb. Virtual bearer management for efficient MTC radio and backhaul sharing in LTE networks. In *Personal Indoor and Mobile Radio Comm. (PIMRC), 2013 IEEE 24th Int. Symp. on*. IEEE, 2013.

[19] S. Sesia, I. Toufik, and M. Baker. *LTE - The UMTS Long Term Evolution: From Theory to Practice*. Wiley, 2 edition, Sept. 2011.

[20] T. Taleb, A. Ksentini, and A. Kobbane. Lightweight mobile core networks for machine type communications. *Access, IEEE*, 2, 2014.

[21] Á. L. Valdivieso Caraguay, A. Benito Peral, L. I. Barona López, and L. J. García Villalba. Sdn: Evolution and opportunities in the development iot applications. *Int. Journal of Distributed Sensor Networks*, 2014, 2014.

[22] Y. Zaki, L. Zhao, C. Goerg, and A. Timm-Giel. LTE wireless virtualization and spectrum management. In *Wireless and Mobile Networking Conf. (WMNC), 2010 Third Joint IFIP*, Oct 2010.