OULUN YLIOPISTO
UNIVERSITY of OULU

Latency-optimized edge computing in Fifth Generation (5G) cellular networks

University of Oulu
Department of Information Processing Science
Bachelor's Thesis
Juuso Haavisto
October 20, 2018

Abstract

The purpose of this thesis is to research latency-optimized edge computing in 5G cellular networks. In specific, the research focuses on low-latency software-defined services on open-source software (OSS) core network implementations.

A literature review revealed that there are few OSS implementations of Long Term Evolution (LTE) (let alone 5G) core networks in existence. It was also found out that OSS is essential in research to allow latency optimizations deep in the software layer. These optimizations were found hard or impossible to install on proprietary systems. As such, to achieve minimal latency in end-to-end (E2E) over-the-air (OTA) testing, an OSS core network was installed at the University of Oulu to operate in conjunction with the existing proprietary one.

This thesis concludes that a micro-operator can be run on current OSS LTE core network implementations. Latency-wise, it was found that current LTE modems are capable of achieving an E2E latency of around 15ms in OTA testing. As a contribution, an OSS infrastructure was installed to the University of Oulu. This infrastructure may serve the needs of academics better than a proprietary one. For example, experimentation of off-the-specification functionality in core networks should be more accessible. The installation also enables easy addition of arbitrary hardware. This might be useful in research on tailored services through mobile edge computing (MEC) in the micro-operator paradigm.

Finally, it is worth noting that the test network at Oulu University is operating at a rather small scale. Thus, it remains an open question if and how bigger mobile network operators (MNOs) can provide latency-optimized services while balancing with throughput and quality of service (QoS).

Tiivistelmä

Tämän opinnäytetyön tarkoituksena on tutkia vasteaikaoptimoitua reunalaskentaa 5G matkapuhelinverkoissa. Tarkemmin määritellen, työn tarkoituksena on keskittyä alhaisen latenssin palveluihin, jotka toimivat avoimen lähdekoodin ydinverkkoimplementaatioiden päällä.

Kirjallisuuskatsaus osoitti että vain pieni määrä avoimen lähdekoodin toteutuksia LTE verkkoimplementaatioista on saatavilla. Lisäksi havainnointiin että avoimen lähdekoodin ohjelmistot ovat osa latenssitutkimusta, jotka vaativat optimointeja syvällä ohjelmistorajapinnassa. Minimaalisen vasteajan saavuttamiseksi, avoimen lähdekoodin ydinverkko asennettiin Oulun yliopistolla toimimaan rinnakkain olemassaolevan suljetun järjestelmän kanssa.

Tämä opinnäytetyön johtopäätöksien mukaan mikro-operaattori voi toimia nykyisten avoimen lähdekoodin LTE ydinverkkojen avulla. Vasteajaksi kahden laitteen välillä saavutettiin noin 15ms. Kontribuutioksi lukeutui avoimen lähdekoodin radioverkkoinfrastruktuurin asentaminen Oulun yliopistolle. Tämä avoin infrastruktuuri voinee palvella tutkijoiden tarpeita paremmin kuin suljettu järjestelmä. Esimerkiksi, ydinverkkojen testaus virallisten määrittelyn ulkopuolisilla ominaisuuksilla pitäisi olla helpompaa kuin suljetulla järjestelmällä. Lisäksi asennus mahdollistaa mielivaltaisen laskentaraudan lisäämisen mobiiliverkkoon. Tämä voi olla hyödyllistä räätälöityjen reunalaskentapalveluiden tutkimuksessa mikro-operaattoreiden suhteen.

Lopuksi on hyvä mainita että Oulun yliopiston testiverkko toimii suhteellisen pienellä skaalalla. Täten kysymykseksi jää miten suuremmat mobiiliverkkojen tarjoajat voivat toteuttaa vasteaikaoptimoituja palveluita suoritustehoa ja palvelunlaatua uhraamatta.

Contents

Glossary

3GPP 3rd Generation Partnership Project. 10, 18

5G Fifth Generation. 1, 6–13, 15, 17, 19–21

5GTN+ 5G Test Network+. 7

AR augmented reality. 6, 12–14

AV1 AOMedia Video 1. 14

CoMP coordinated multi-point. 16

CP cyclic prefix. 15

DSP digital signal processor. 10, 11, 16

E2E end-to-end. 1, 7, 12–14, 18–21

eMBB enhanced mobile broadband. 10

EPC evolved packet core. 7, 10, 16, 18, 19, 21

FFT fast Fourier transform. 15

FPGA field-programmable gate array. 16

HARQ Hybrid Automated Repeat Request. 14

HMD head-mounted display. 6

IoT Internet of Things. 6, 8

LAN local area network. 18

LTE Long Term Evolution. 1, 7, 10, 12, 14, 16, 18, 19, 21

MEC mobile edge computing. 1, 7, 9, 11, 16, 19–21

MIMO multiple-input and multiple-output. 16

mMTC massive machine-type communications. 10

mmWave millimeter wave. 8, 10, 13

MNO mobile network operator. 1, 6, 8–10, 16, 18, 21

NFV network function virtualization. 8, 10, 16

NIC network interface controller. 17

OFDM Orthogonal Frequency Division Multiplexing. 15

OFDMA Orthogonal Frequency Division Multiple Access. 14, 15

OSS open-source software. 1

OTA over-the-air. 1, 7, 15, 18

PAPR peak-to-average power ratio. 15

QoS quality of service. 1, 14, 21

QUIC Quick UDP Internet Connections. 14

RAN radio access network. 6–10, 14–20

RTP Real-time Transport Protocol. 14

SDN software defined network. 8–11, 16, 17

TCP Transmission Control Protocol. 19

TTI Transmission Time Interval. 14

UDP User Datagram Protocol. 14

UDT UDP-based Data Transfer Protocol. 13

UE user equipment. 6, 7, 9, 13–15, 18, 19, 21, 22

URLLC ultra-reliable low-latency communications. 10

VOR Vestibulo-Ocular Reflex. 13, 14

VR virtual reality. 12–14

WAN wider area network. 13, 18

1 Introduction

The motivation of this thesis is to research how much resources (power consumption, bandwidth usage, hardware investments) can be saved if the computation capability of mobile devices are enhanced with edge computing. In specific, the idea is to find constraints under which holographic content of augmented reality (AR) applications could be displayed on a mobile head-mounted display (HMD). This research is deemed essential to map required areas of research and development in hardware and software towards an eyewear alike HMD form factor.

In general, edge computing technologies need high-bandwidth and low-latency connectivity (Kämäräinen, Siekkinen, Ylä-Jääski, Zhang, & Hui, 2017). On mobile devices, the latency on a typical connection will likely be a bottleneck, yielding little compute benefit (Simsek, Aijaz, Dohler, Sachs, & Fettweis, 2016). However, considering 5G, which reduces roundtrip network latency to one millisecond (Parvez, Rahmati, Guvenc, Sarwat, & Dai, 2018), one could disregard such bottlenecks (Deber, Jota, Forlines, & Wigdor, 2015).

Thus, this thesis researches 5G enabled edge computing from a latency-optimized perspective. The goal is an indistinguishable real-time augmentation of the computation capability of a mobile device through on-demand resource provisioning using a 5G radio access network (RAN). In other words, the idea is to ”loan” processing power from the 5G core network to enable the user equipment (UE) to run applications infeasible on its on-device hardware. This could, in effect, be also used to augment the capabilities of Internet of Things (IoT) devices. As an example, IoT vacuum cleaners could be loaded with object recognition to follow humans without hardware upgrades.

It is hypothesized that a shift in computation towards a future of edge computing is about to happen. In such future, on-device processing power is replaced with wireless modems. This does assume the existence of a network with extensive computation capability. However, that is envisioned to be handled by MNOs in the 5G era (Tran, Hajisami, Pandey, & Pompili, 2017). This research hopes, in part, to contribute insight into how such MNO might work.

2 Research method

This thesis consists of a literature review and an empirical study. Chapter 3 is a literature review. It gathers research insight for a latency-optimized evolved packet core (EPC) installation. The literature was found primarily through Google Scholar. Iris.ai was also used to find literature about 5G latency. The search with Iris.ai was seeded with cited research papers. Literature regarding software-based RAN optimizations turned out to be hard to find. The prior literature on RAN latency could have been considered to focus on radio hardware and positioning rather than software. In general, E2E OTA testing of 5G MEC on UE application layer was non-existent. To elaborate, the papers with research on latency optimizations of the EPC, such as (Mao et al., 2015), were limited to virtualized environments. This meant no results of UE's application layer latencies were offered. The most tangential research paper found, in regards to the UE application layer, could be considered to be (Kämäräinen et al., 2017). This work also included a similar testing premise of a locally controlled LTE access point. However, this work did not attempt to improve the server delay of the LTE connection through 5G MEC paradigm.

Chapter 4 is an empirical study. It puts the insight from the literature review into practice. More specifically, the EPC is virtualized on general-purpose hardware defined on appendix A, and the UE and the base station are physical hardware. Then, a set of OTA tests are conducted using Oulu University's spectrum licenses. This is to collect practical results. Finally, the thesis concludes with a reflection of our findings.

This research took place during a three month period in summer 2018. The research was funded by the Centre for Wireless Communications at Oulu University, under a project called 5G Test Network+ (5GTN+). 5GTN+ is part of Business Finland's 5thGear program.

3 Previous research

In this chapter, relevant terms and paradigms using prior literature are defined. In part, this is to reason and document the design choices for the open-source core network implemented in chapter 4.

3.1 5G

In addition to the demand for increased network capacity, there are many other features upcoming in 5G. These features include the move to millimeter wave (mmWave) spectrum, network slicing e.g. new market-driven way of allocating and re-allocating bandwidth, virtualization which starts from the core network and progressively spreads to the edges of the network, IoT-networks comprised of billions of miscellaneous devices, and the increasing integration of past and current cellular and WiFi standards to provide a ubiquitous high-rate, low-latency experience for network users (Andrews et al., 2014).

New performance requirements are thus placed on devices and networks. This applies especially to latency, peak throughput, and spectral efficiency. Second, new enabling technologies, software defined networks (SDNs) and network function virtualizations (NFVs) provide momentum for new design principles. These technologies shape how mobile networks will be developed, deployed and operated (Trivisonno, Guerzoni, Vaishnavi, & Soldani, 2015).

In that respect, the most relevant event is the movement of data to the cloud so that it can be accessed from anywhere and via a variety of platforms. This fundamentally redefines the endpoints and the time frame for which network services are provisioned. It requires that the network would be much more nimble, flexible and scalable. As such, together SDN and NFV represent the most significant advance in mobile communication networking in the last 20 years (Andrews et al., 2014).

3.1.1 RAN

An essential piece of the cellular network infrastructure is the RAN. It provides wide-area wireless connectivity to mobile devices. The fundamental problem the RAN solves is figuring out how best to use and manage limited spectrum to achieve this connectivity. In a dense wireless deployment with mobile nodes and limited spectrum, it becomes a difficult task to allocate radio resources, implement handovers, manage interference, and balance load between cells. As such, local geographical MNO networks may be needed to effectively perform load balancing and interference management, as well as maximize throughput, global utility, or any other goal (Gudipati, Perry, Li, & Katti, 2013).

Traditional radio access also has issues with power consumption and latency. This

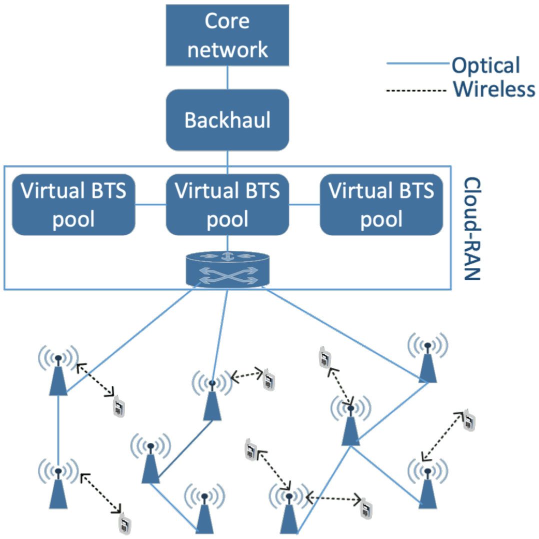

Figure 1: Cloud-RAN architecture in 5G networks.

Source: (Parvez et al., 2018)

is problematic with 5G services, such as edge computing. To elaborate, the power spent on mobile users far from a cell may be significant. Regarding edge computing, this substantial transmit power may nullify all potential energy savings. One way to address this problem is by bringing the computational resources closer to the UE (Barbarossa, Sardellitti, & Di Lorenzo, 2014) through increased cell deployments. In 5G, these increased deployments are closely-knit with the MEC paradigm. In general, MECs are not a new way of thinking: similar approaches have been studied under different names, such as cyberforaging, (Sharifi, Kafaie, & Kashefi, 2012), grid technology (Foster, Kesselman, & Tuecke, 2001), computation offloading (Fernando, Loke, & Rahayu, 2013), cloudlets (Satyanarayanan, Bahl, Caceres, & Davies, 2009), and fog computing (Yi, Hao, Qin, & Li, 2015). Conceptually MEC differs from all these by positioning the computation resources to the mobile base stations (Blanco et al., 2017).

To elaborate, MEC could be defined as an architecture, which provides computing, storage, and networking resources within the RAN. MEC is part of the SDN paradigm, and as such, is deployed on general-purpose hardware. MECs are defined to enable delay-sensitive, context-aware, high-bandwidth applications and insight, to be executed, and retrieved near the end-users. MECs alleviate backhaul usage and computation at the core network (Tran et al., 2017; Blanco et al., 2017), which in turn helps MNOs to improve profit margins and reduce network costs (Saguna & Intel, 2016). Alleviated backhaul usage also reduces latency and improves the mobile users' experience (Tran et al., 2017). MECs are owned and managed by the infrastructure provider, attached to the base stations (Blanco et al., 2017). Studies suggest that resources of MECs could also be rented between MNOs through network slicing (Samdanis, Costa-Perez, & Sciancalepore, 2016).

3.1.2 Hardware

From a hardware perspective, the wireless access is being developed with three broad use case families in mind: enhanced mobile broadband (eMBB), massive machine-type communications (mMTC), and ultra-reliable low-latency communications (URLLC) (Tullberg et al., 2016). Because spectral band often called ”beachfront spectrum” has become nearly occupied, these access technologies rely on a relatively unused spectrum in the mmWave range of 30–300 GHz, where wavelengths are 1–10 mm. The main reason that mmWave spectrum lies idle is that, until recently, it had been deemed unsuitable for mobile communications. This is due to very short range transmission. However, semiconductors are maturing, their costs and power consumption falling and the other obstacles related to propagation are now considered increasingly surmountable (Andrews et al., 2014).

This, in turn, means the number of base station installations also need to increase. This is because increased frequencies decrease the distance and materials which the radio signal can travel (Parvez et al., 2018). Other challenges related to hardware include that current cellular network services rely on digital signal processor (DSP) units. DSPs are proprietary devices that are purpose-built (Sherry et al., 2012; Wang, Qian, Xu, Mao, & Zhang, 2011). DSPs have caused a problem for MNOs: to make service additions or upgrades the whole DSP has to be replaced (Li & Chen, 2015).

Additionally, integration with LTE is necessary for quick and efficient deployment of 5G. In summary, 5G wireless access should be an evolution of LTE complemented with architecture designs and radio technologies (Parvez et al., 2018). As such, 5G-enabled devices having radios capable of LTE communication is envisioned (Andrews et al., 2014; Larew, Thomas, Cudak, & Ghosh, 2013). In such hybrid system, system information, control channel, and feedback are transmitted in the LTE system, making the mmWave spectrum available for data communication (Pi & Khan, 2011).

3.1.3 Software

Virtualization has been introduced to decouple software from hardware to address the aforementioned problems. Such installations are called SDNs which make use of NFVs. This essentially means that a RAN can function on general-purpose hardware (Chowdhury & Boutaba, 2009, 2010; Schaffrath et al., 2009). What follows is that abstractions like container-orchestration systems, developed for decades (Burns, Grant, Oppenheimer, Brewer, & Wilkes, 2016) in software engineering, can be used to overcome challenges of RANs in 5G, such as dynamic resource allocation (Li & Chen, 2015).

In Release 15, 3rd Generation Partnership Project (3GPP) is defining 5G and its core network as the Next Generation Core. This essentially means that as a 5G-novelty, base stations and EPCs are becoming intelligent edge computing resources. This can be considered to imply a change similar to which is envisioned with fog computing. As such, 5G also implies further decentralization of computer networks.

Figure 2: Analogy which compares DSP-based network functions with SDN.

Source: Website of (Zaostrovnykh et al., 2017).

To elaborate, fog computing proposes a shift from cloud computing on datacenters towards a large number of geographically widespread edge nodes as part of a distributed and collaborating cloud. In 5G this is done via the MEC-paradigm. Additionally, fog computing is defined to offer storage and to help offload traffic which would need to transverse the backbone. These are the same traits the MEC is described to solve. However, the notion of fog computing nodes is wide: it defines that any equipment with processing power and storage, e.g., switches and routers, base stations, datacenters, and cloud platforms can be qualified as a fog node (Taleb et al., 2017).

It could be considered that fog computing is the ”holy grail” of distributed information networks in 5G. However, in comparison to the MEC-paradigm, fog computing is envisioned to work in a trustless way. As such, the challenges are vast: at the edge of the network, user privacy and data security protection are the essential services that should be provided. Second is the ownership of the data collected from things at the edge. Just as what happened with mobile applications, the data of end user collected by things will be stored and analyzed at the service provider side (Shi, Cao, Zhang, Li, & Xu, 2016). However, with fog computing, the ”service provider” might be your neighbor’s refrigerator. As such, it can be considered a challenge whether leaving the data at the edge is any better for privacy than sending it to a data-silo. In other words, if nodes are interworking, how it can be programmatically ensured that the refrigerator protects user privacy?

In general, this problem domain could be defined as safe distributed trustless computing in decentralized networks. This implies more interest towards distributed network and application development. This field, in general, could be considered to have taken leaps of progression through projects like Ethereum, which allow applications on a decentralized, but singleton, compute resource. Ethereum calls this paradigm a transactional singleton machine with a shared-state (Wood, 2014). This progress could be considered significant in terms of 5G and fog computing as well, because Ethereum is solving problems, such as execution model in a distributed state system. In (Wood, 2014) this is called a quasi-Turing complete machine. The paper proposes a parameter called ”gas”, which defines the amount of computation a node can request and execute using another node. While Ethereum only allows programs to be executed within its virtual machine, it could be envisioned that in the future more general-purpose programs could

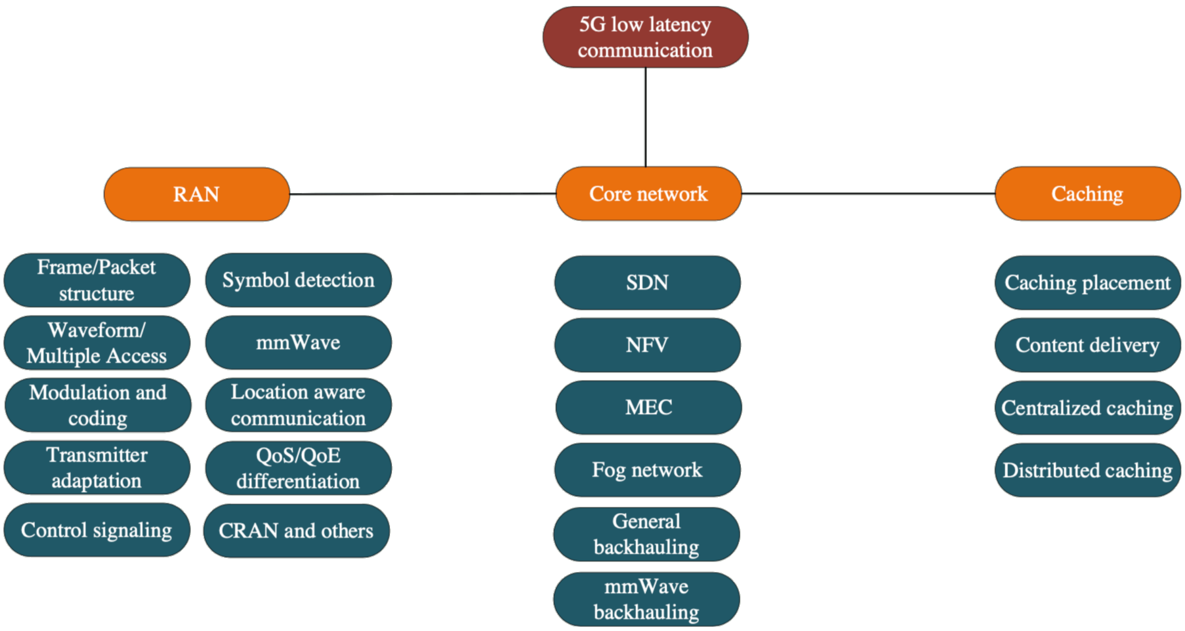

Figure 3: Categories of different solutions for achieving low latency in 5G.

Source: (Parvez et al., 2018)

be executed, but which are still bound by similar intrinsic constraints.

Current cryptocurrency projects like Ethereum are not explicitly addressing this for the use of 5G, but the parallelism is evident. However, creating generic software out of the bounds of a virtual machine may be hard, as pointed out by (Whitman & Mattord, 2011): technical software failure, such as (1) buffer, integer and stack overflow, (2) string formatting problems, (3) race conditions, and (4) access to uninitialized or deallocated memory are all common problems with software, problems that result in software that is difficult or impossible to deploy in a secure fashion.

3.2 Latency

E2E latency can be considered to consist the sum of time to (1) send the program execution information from the mobile device to the cloud, (2) run the program at the remote side (cloud), and (3) the time to return results from the cloud back to the mobile unit (Barbarossa et al., 2014). Current LTE E2E latencies are on the order of about 10-15 milliseconds and are based on the one millisecond subframe time with necessary overheads for resource allocation and access. Although this latency is sufficient for most current services, anticipated 5G applications include two-way gaming, novel cloud-based technologies such as those that may be touchscreen activated, and virtual reality (VR) and AR. As a result, 5G will need to support roundtrip latency an order of magnitude faster than LTE. In addition to shrinking down the subframe structure, latency constraints may have significant implications on design choices at several layers of the protocol stack and the core network (Andrews et al., 2014).

In edge computing, we could further break down E2E latency to be consisted of three different things, as measured in (Kämäräinen et al., 2017): (1) device delay, (2) access delay, and (3) server delay. Determining adequate latency with E2E edge computing can be considered to depend on the visual applications ran on the UEs. For example, applications in AR and VR can be tied to a neurological process called Vestibulo-Ocular Reflex (VOR). In VOR, the brain coordinates eye and head movements to stabilize images on the retina. This is critical to synchronizing virtual and real objects to create a coherent immersion in AR, for example. The VOR process takes the brain seven milliseconds (Zheng et al., 2014). VOR is related to motion-to-photon latency, which (Anthes, García-Hernández, Wiedemann, & Kranzlmüller, 2016) defines as the overall system lag. In VR, the maximum motion-to-photon latency should be 20 milliseconds. Frame rates lower than 20 milliseconds break the immersive experience and cause nausea (Anthes et al., 2016). This effectively means that the E2E latency should be less than 20 milliseconds.

A further literature review reveals that little attention has been paid to practical E2E reliability, latency, and energy consumption comprising both up and downlinks (Condoluci, Araniti, Mahmoodi, & Dohler, 2016). Studies like (Mezzavilla et al., 2018) conclude that E2E mmWave technology also needs innovations across all layers of the communication protocol stack. Similar conclusions were found from existing protocol whitepapers, like that of UDP-based Data Transfer Protocol (UDT). The UDT whitepaper (Gu & Grossman, 2007) concludes that the congestion control algorithm is the primary internal functionality to enable the use of high bandwidth effectively.

Considering the aforementioned, the subsections of this chapter go through each source of latency to better define the causes, and to evaluate possible bottlenecks and resolutions towards a zero latency 5G network.

3.2.1 Device delay

Combining the insight of (Kämäräinen et al., 2017; Jacobs, Livingston, et al., 1997) we can consider following factors affecting the UE: (1) input, (2) rendering, (3) display, (4) synchronization, and (5) frame-rate-induced delay. For this thesis, this could be further simplified into two categories determined by hardware components: (1) a gyroscope sensor and (2-4) the mobile graphics processing unit. We ignore frame-rate-induced delay because mobile displays achieving refresh rates of 144Hz and beyond are accessible by consumers today.

Prior art by (Lincoln et al., 2016) achieved rendering latency of 0.08 milliseconds per frame. This system used a local accelerated computing pipeline and could be considered, for this thesis, as the baseline rendering latency.

Study (Kämäräinen et al., 2017) used a remote rendering system with a Samsung S7 UE as the client. In this case, the application is fully rendered in the wider area network (WAN), to which the UE basically acts as a remote controller. This sort of remote computing using video can be considered an apt approach due to the interoperability of the medium. That is, as long as the content can be encoded as video, it does not matter whether the content consumed is AR, VR or video-games. The downside is that the UE needs to decode and render video frames, which took the UE about 25 milliseconds for a single h264 frame. Thus it can be concluded that with remote rendering systems looking to meet VOR deadlines, the latency optimizations should primarily address the transport protocol, and the underlying video codec and its decoder implementation.

Research in (Suznjevic, Slivar, & Skorin-Kapov, 2016) shows that Nvidia's remote rendering service uses Real-time Transport Protocol (RTP) over User Datagram Protocol (UDP). As such, we could consider some recent developments to improve latency. For example, (Liu, 2018) concludes that recent video codecs such as AOMedia Video 1 (AV1) achieve bit rate savings of around 40 to 50 percent over h264. The savings do not come free but are a trade-off of a much longer encoding process. Similarly, the transport protocol could be evaluated against novel approaches like Quick UDP Internet Connections (QUIC), which is designed to reduce latency (Langley et al., 2017) and have shown to improve QoS under LTE services when requesting web content (Qian, Wang, & Tafazolli, 2018). Initial studies on streaming content over QUIC confirm increased video performance, albeit only when the packets do not need to travel over long physical distances (Bhat, Rizk, & Zink, 2017).

On the other hand, even if we suppose that we can copy the performance gains realized in (Liu, 2018), we would still be left with an effective device delay of 12.5ms, considering the benchmarks of (Kämäräinen et al., 2017). However, this latency does not yet include input delay, which according to (Kämäräinen et al., 2017) was found to be 12.2ms for a gyroscope. The sum of these two factors is already over 20ms. Because access and network delay are not yet considered, it can be deduced that further optimizations are required in consumer-grade UE to provide immersive video-based experiences which make use of computation offloading in RANs.

3.2.2 Access delay

As depicted in figure 4, access delay could be considered the sum of encoding, travel, and decoding of packets of the modems. In other words, access delay could be considered to be the actual wireless communication part in an E2E setting. As such, latency optimizations in this interface could be considered very low-level in comparison to, e.g., rendering latency. Considering the categories defined in figure 3, the optimizations most accessible software-wise in this section could be considered to address frame and packet structure, and modulation and coding systems.

Due to its fundamental role in E2E applications, it is difficult to achieve significant latency improvements without significant impact on the air interface. This is because of the latency relevant aspects such as frame structure, control signaling timing, and Hybrid Automated Repeat Request (HARQ) form the critical building blocks of the air interface. For example, a round trip time requirement of 0.1 millisecond means that the Transmission Time Interval (TTI) length needs to be scaled down to 10-25 µs. This cannot be achieved with the current LTE Orthogonal Frequency Division Multiple

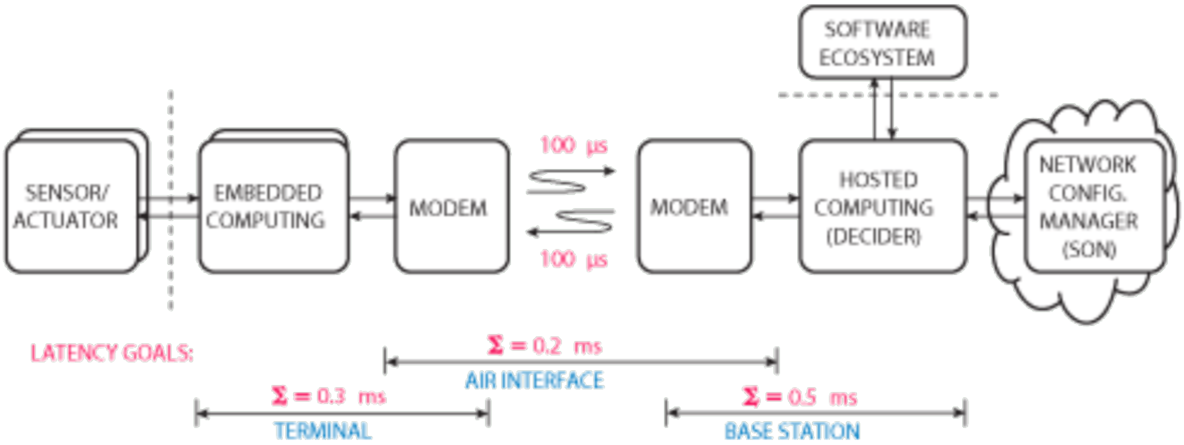

Figure 4: Breakdown of the 1 millisecond access delay in 5G UE.

Source: (Fettweis, 2012)

Access (OFDMA) numerology. As a solution to this problem, a scalable OFDMA design is offered as a means to optimize the radio performance. In specific, reduction of cyclic prefix (CP), the reference signal, and control signaling overhead are considered to provide a significant increase in throughput. It is worth noting that optimizations for efficiency and latency will be conflicting. Thus the system needs to provide adaptability to select configurations best supporting the specific services and applications used (Raaf et al., 2011).

It is considered that a tunable Orthogonal Frequency Division Multiplexing (OFDM) could be adapted in 5G. In particular, given the increasingly software-defined nature of radios, the fast Fourier transform (FFT) block size, the subcarrier spacing, and the CP length could change with the channel conditions: in scenarios with small delay spreads the subcarrier spacing could grow, and the FFT size and the CP could be shortened to lower (1) the latency, (2) the peak-to-average power ratio (PAPR), (3) the CP's power and bandwidth penalty, and (4) the computational complexity. In channels with longer delay spreads, that could revert to narrower subcarriers, longer FFT blocks, and a longer CP (Andrews et al., 2014).

It is also worth noting that, in comparison to device and network delay optimizations, optimizations in access delay may require spectrum licenses for empirical studies. For example, (Mezzavilla et al., 2018; Condoluci et al., 2016; Parvez et al., 2018) call for increased need for OTA testing. It could be deduced to mean testing which includes real-world air interface interference in comparison to virtualized tests. Moreover, as mentioned earlier, optimizations in other layers could be considered to be always bounded by access delay. Thus, it could be considered essential to not forget about testing new findings in real-world scenarios.

Also, because of the bounding nature of the access delay, it could be considered that 5G RANs could even dynamically optimize its processing of access delay in the base stations. To elaborate, should there be a discovery mechanism for available hardware of the core network, given the depicted time of base station packet processing of 0.5 ms

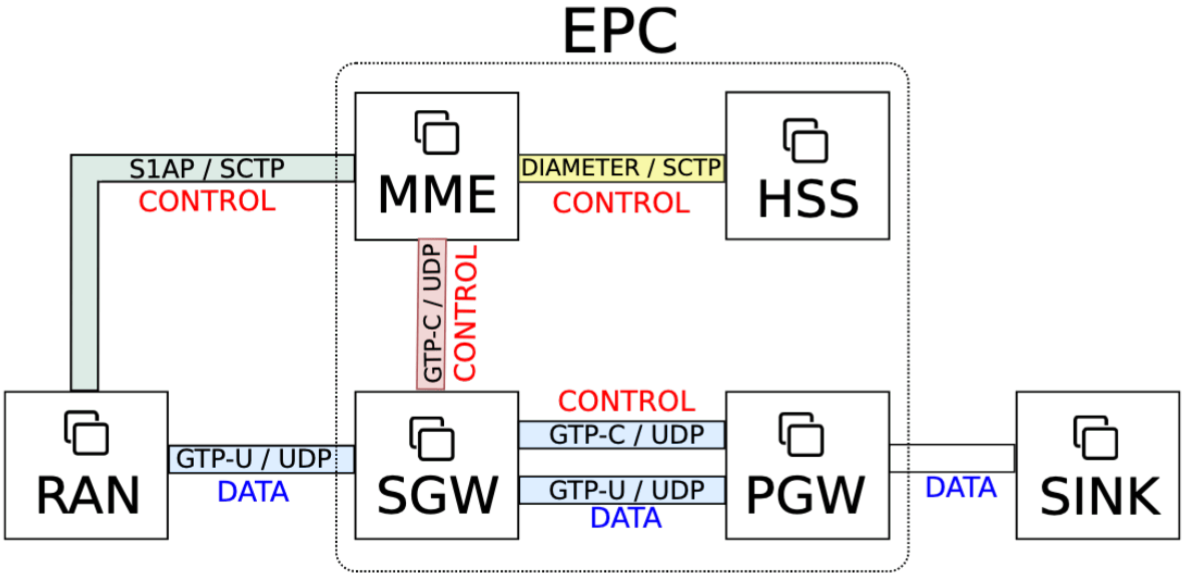

Figure 5: NFV-based LTE EPC implementation.

Source: (Jain et al., 2016)

in figure 4, it might make sense to use the optical connection to re-route the processing. In such a scenario, accelerated hardware residing in MECs, such as field-programmable gate arrays (FPGAs), could shorten not only the latency but also create more cost-effective base stations. To elaborate, if it could be assumed that base stations can schedule hardware from the MEC and such hardware exists in the network, then one could, similarly to the vision of this thesis, remove computing capabilities from base station and instead rely on the MEC to provide more cost-effective approaches for RANs. This could, in effect, used to reduce capital expenses of MNOs to provide service by reducing the unit price of base stations.

3.2.3 Network delay

Network delay is essentially the delay caused by the core network, i.e., the EPC in the backhaul of the base stations. Figure 5 demonstrates the essential parts of a LTE EPC and how these parts communicate technically between each other. Historically different parts of the EPC have been run on custom-purpose DSP hardware, hence the separation, but with SDN the EPC could technically operate under the same host computer.

In general, RANs are real-time applications, which require hard deadlines. This is to maintain protocol, frame, and subframe timings, and to perform transmission schemes such as beamforming, multiple-input and multiple-output (MIMO), coordinated multi-point (CoMP), and Massive MIMO (Nikaein et al., 2017). DSP systems can endorse these requirements through hardware design. However, this does not apply for software-based RANs which are powered by general-purpose processors. Instead, tuning the software environment to meet real-time processing is required. This essentially applies

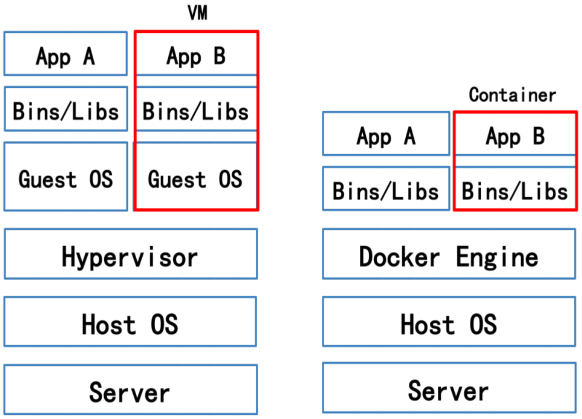

Figure 6: Architecture comparison of a virtual machine and a container

Source: (Zhang et al., 2018)

to the EPC as well. As so, (Mao et al., 2015) finds an optimized kernel essential for RANs. The study concludes that to overcome latency limitations of the Linux network stack (1) fine-tuned real-time kernel and (2) low-overhead virtualization hypervisor with kernel passthrough for network interface controllers (NICs) are needed.

To elaborate, the 5G RAN SDN-paradigm defines the software to run in virtualized environments. However, this introduces processing overhead to applications, as demonstrated in figure 6: regardless of whether virtual machines or containers are used, there still exists abstraction layers which affect processing time. By introducing kernel passthrough technologies as suggested by (Mao et al., 2015), one can essentially disregard the number of abstractions after the ”Server” step and skip directly to the application ”App” layer. These approaches reduce the processing so that the envisioned latency goals of 5G can be met through optimizations on the software-defined architecture (Mao et al., 2015).

4 Contribution

4.1 Review of OSS EPCs

The University of Oulu has spectrum licenses to do OTA testing. From the perspective of this thesis work, this means a LTE network operated by the university is accessible at the campus to LTE UEs. The university owns the hardware of the RAN and does not roam on an existing MNO. This means that the university could be considered an independent MNO and more specifically, a micro-operator (Ahokangas et al., 2016). The local infrastructure is detailed in more depth in thesis works, for example in (Arif, 2017).

At the start of this thesis work, the infrastructure at Oulu University could have been considered proprietary by a software engineer. To elaborate, the base stations and the EPC were using open and standardized protocols with abstractions offering software-based control. However, there was no kernel level access to these devices. In itself, this was not a limiting factor for latency studies in the UE application layer. However, the EPC of the infrastructure resided physically outside of the campus. This caused an E2E latency of 50 milliseconds within devices in the same room. The EPC was concluded to be the cause for the latency by intercepting the connection from a picocell to the EPC. It was concluded that should a commercial micro-operator offer latency-optimized services, an EPC residing in the WAN would not likely mimic a production system. Thus to remove the network hop to the WAN, it was deduced necessary to install an alternative EPC to the campus' local area network (LAN).

Additionally, the literature review on latency concluded that RANs require kernel optimizations (Mao et al., 2015). Moreover, having a customizable RAN was seen as useful in future research by faculty researchers. Considering these factors, an open-source alternative seemed useful. A brief literature review on open-source EPCs was conducted. The following implementations of EPCs were found and considered (listed in no particular order):

Project	Language	Anecdotal summary
NextEPC	C	Supports 3GPP Release 13
corenet	Python	Minimal 3G and LTE EPC
vEPC	C++	Developed with performance evaluation in mind (Jain et al., 2016)
openair-cn	C	"Probably the most complete open 4G project so far." (Laursen, 2015) "[openair-cn] seems to be the most feature-complete implementation of all [open-source] solutions right now." (Ricudis, 2017)
srsLTE	C++	"srsLTE have both the code elegance of openLTE and the completeness of [openair-cn]" (Cyberlog IT, 2018)
openLTE	C++	"Ben Wojtowicz, almost single-handedly developed openLTE, an open-source LTE software implementation, in his spare time." (Laursen, 2015)

For its ease of installation, NextEPC was picked. The installation is specified in appendix A.

An open-source EPC was found practical for micro-operator because of cost reductions. The same could apply to LTE on an unregulated spectrum (muLTEfire) and rural deployments akin to Nokia Kuha. In general, open-source might also help to avoid security holes in EPCs, which are known to exist (Shaik, Borgaonkar, Asokan, Niemi, & Seifert, 2015; Rupprecht, Kohls, Holz, & Pöpper, n.d.). To elaborate, if the development would be open and accessible, more people could contribute towards a safe software by design. Whether these statements hold would need further research.

4.2 OTA testing of latency-optimized OSS EPC

Software-wise, LTE and 5G are to coexist (Andrews et al., 2014; Larew et al., 2013; Pi & Khan, 2011). This means the software which translates a wireless signal to an Internet connection will be the same on LTE and 5G. From this thesis' perspective, it means it is technically correct to use an LTE UE to measure the latency of a 5G RAN. This is considered true because in general, 5G UE is capable of offering better latency than LTE UE because of an improved air interface. What follows is any offloaded application, concerning E2E latency, should perform better, or at least as well, on a 5G UE.

Due to focus constraints of this thesis, further application development is left for future studies. Albeit, the following test results on latency are measured using commodity hardware specified in appendix A:

Running iperf3 and analyzing the traffic with Wireshark reveals that from 337'000 packets around 64'000 get a warning telling that Transmission Control Protocol (TCP) frame is (suspected) to be out-of-order. In addition to this, around 2700 packets get a connection reset warning. This could be considered to imply performance bottlenecks in the EPC's software implementation or hardware.

Ping requests show that latency varies between 15 and 30ms when pinging from the EPC host to the UE. However, this ping varies on an either-or-basis, i.e., the ping tends to be either 15ms or 30ms, but not many results in between are seen.

4.3 Discussion

It can be deduced that MECs could also be used to form a new application distribution channel for software developers. Such applications could assume the low-latency guarantees and edge computing resources of the MEC. A novel benefit is that the MECs is part of the network layer. Thus, such a platform would be an abstraction independent of the operating system of the UE. For this reason, such a system could be considered a ubiquitous "App-Store layer" of 5G.

In agreement with (Saguna & Intel, 2016), it can be deduced that the MEC architecture will become an essential component of 5G. However, this might not solely be for the end-user benefits. Instead, we can consider the MEC also as an essential testing ground for E2E research. To elaborate, MEC can enable software engineers to take part in RAN development through E2E applications. This could be one natural vector of research interest towards technologies called upon in the literature review, regarding innovations in different parts of the RAN.

5 Conclusions

In the contribution chapter an LTE UE, a 5G MEC, a micro-operator, an open-source EPC, and software-based latency optimizations were combined. An empirical study proved that by utilizing MEC an E2E latency close to the practical limits of LTE modems can be achieved. This suggests that micro-operators could, as of today, provide services unrealized by current MNOs.

This thesis could be considered to pose grounds for further research to define the viability of the proposed infrastructure. For example, it could be practical to define the limits of QoS which small MNOs can offer in comparison to bigger ones. A continuation of latency research could also be considered. Either way, a more available spectrum policy would likely be helpful to realize the full capability of 5G radios to optimize the full E2E latency. It can also be deduced that without liberal spectrum policies the findings of these approaches are likely hindered. This would, in effect, then affect the interests of the consumer and the broader market potential of 5G.

A Relevant technical specification

Hardware-wise, the EPC consisted of Intel i3-6100T processor, Crucial 8GB DDR4 memory (CT2K4G4DFS824A), Asrock Z270 motherboard, and Samsung 250GB 850 EVO SSD.

The eNodeB used was Nokia FW2HHWC. The UE used was a OnePlus A5010.

Software-wise, the EPC was running on Ubuntu 18.04 with kernel 4.16.15-rt7. The NextEPC environment was installed from the source. The git commit used was d004770.

References

Ahokangas, P., Moqaddamerad, S., Matinmikko, M., Abouzeid, A., Atkova, I., Gomes, J. F., & Iivari, M. (2016). Future micro operators business models in 5g. *The Business & Management Review*, *7*(5), 143.

Andrews, J. G., Buzzi, S., Choi, W., Hanly, S. V., Lozano, A., Soong, A. C., & Zhang, J. C. (2014). What will 5g be? *IEEE Journal on selected areas in communications*, *32*(6), 1065–1082.

Anthes, C., García-Hernández, R. J., Wiedemann, M., & Kranzlmüller, D. (2016). State of the art of virtual reality technology. In *Aerospace conference, 2016 ieee* (pp. 1–19).

Arif, M. (2017). *Openepc integration within 5gtn as an nfv proof of concept* (Unpublished master's thesis). University of Oulu, Faculty of Information Technology and Electrical Engineering, Communications Engineering.

Barbarossa, S., Sardellitti, S., & Di Lorenzo, P. (2014). Communicating while computing: Distributed mobile cloud computing over 5g heterogeneous networks. *IEEE Signal Processing Magazine*, *31*(6), 45–55.

Bhat, D., Rizk, A., & Zink, M. (2017). Not so quic: A performance study of dash over quic. In *Proceedings of the 27th workshop on network and operating systems support for digital audio and video* (pp. 13–18).

Blanco, B., Fajardo, J. O., Giannoulakis, I., Kafetzakis, E., Peng, S., Pérez-Romero, J., ... others (2017). Technology pillars in the architecture of future 5g mobile networks: Nfv, mec and sdn. *Computer Standards & Interfaces*, *54*, 216–228.

Burns, B., Grant, B., Oppenheimer, D., Brewer, E., & Wilkes, J. (2016). Borg, omega, and kubernetes. *Queue*, *14*(1), 10.

Chowdhury, N. M. K., & Boutaba, R. (2009). Network virtualization: state of the art and research challenges. *IEEE Communications magazine*, *47*(7).

Chowdhury, N. M. K., & Boutaba, R. (2010). A survey of network virtualization. *Computer Networks*, *54*(5), 862–876.

Condoluci, M., Araniti, G., Mahmoodi, T., & Dohler, M. (2016). Enabling the iot machine age with 5g: Machine-type multicast services for innovative real-time applications. *IEEE Access*, *4*, 5555–5569.

Cyberlog IT. (2018, May). *Build a lte network with srslte and program your own usim card.* Retrieved from `https://cyberloginit.com/2018/05/03/build-a-lte-network-with-srslte-and-program-your-own-usim-card.html`

Deber, J., Jota, R., Forlines, C., & Wigdor, D. (2015). How much faster is fast enough?: User perception of latency & latency improvements in direct and indirect touch. In *Proceedings of the 33rd annual acm conference on human factors in computing systems* (pp. 1827–1836).

Fernando, N., Loke, S. W., & Rahayu, W. (2013). Mobile cloud computing: A survey. *Future generation computer systems*, *29*(1), 84–106.

Fettweis, G. P. (2012). A 5g wireless communications vision. *Microwave Journal*, *55*(12), 24–36.

Foster, I., Kesselman, C., & Tuecke, S. (2001). The anatomy of the grid: Enabling scalable virtual organizations. *The International Journal of High Performance Computing Applications*, *15*(3), 200–222.

Gu, Y., & Grossman, R. L. (2007). Udt: Udp-based data transfer for high-speed wide area networks. *Computer Networks*, *51*(7), 1777–1799.

Gudipati, A., Perry, D., Li, L. E., & Katti, S. (2013). Softran: Software defined radio access network. In *Proceedings of the second acm sigcomm workshop on hot topics in software defined networking* (pp. 25–30).

Jacobs, M. C., Livingston, M. A., et al. (1997). Managing latency in complex augmented reality systems. In *Proceedings of the 1997 symposium on interactive 3d graphics* (pp. 49–ff).

Jain, A., Sadagopan, N., Lohani, S. K., & Vutukuru, M. (2016). A comparison of sdn and nfv for re-designing the lte packet core. In *Network function virtualization and software defined networks (nfv-sdn), ieee conference on* (pp. 74–80).

Kämäräinen, T., Siekkinen, M., Ylä-Jääski, A., Zhang, W., & Hui, P. (2017). A measurement study on achieving imperceptible latency in mobile cloud gaming. In *Proceedings of the 8th acm on multimedia systems conference* (pp. 88–99).

Langley, A., Riddoch, A., Wilk, A., Vicente, A., Krasic, C., Zhang, D., ... others (2017). The quic transport protocol: Design and internet-scale deployment. In *Proceedings of the conference of the acm special interest group on data communication* (pp. 183–196).

Larew, S. G., Thomas, T. A., Cudak, M., & Ghosh, A. (2013). Air interface design and ray tracing study for 5g millimeter wave communications. In *Globecom workshops (gc wkshps), 2013 ieee* (pp. 117–122).

Laursen, L. (2015, November). *Software-defined radio will let communities build their own 4g networks.* Retrieved from `https://spectrum.ieee.org/telecom/wireless/softwaredefined-radio-will-let-communities-build-their-own-4g-networks`

Li, Y., & Chen, M. (2015). Software-defined network function virtualization: A survey. *IEEE Access*, *3*, 2542–2553.

Lincoln, P., Blate, A., Singh, M., Whitted, T., Lastra, A., Fuchs, H., et al. (2016). From motion to photons in 80 microseconds: Towards minimal latency for virtual and augmented reality. *IEEE Transactions on Visualization & Computer Graphics*(4), 1367–1376.

Liu, Y. (2018, April). *Av1 beats x264 and libvpx-vp9 in practical use case.* Retrieved from `https://code.fb.com/video-engineering/av1-beats-x264-and-libvpx-vp9-in-practical-use-case/`

Mao, C.-N., Huang, M.-H., Padhy, S., Wang, S.-T., Chung, W.-C., Chung, Y.-C., & Hsu, C.-H. (2015). Minimizing latency of real-time container cloud for software radio access networks. In *Cloud computing technology and science (cloudcom), 2015 ieee 7th international conference on* (pp. 611–616).

Mezzavilla, M., Zhang, M., Polese, M., Ford, R., Dutta, S., Rangan, S., & Zorzi, M. (2018). End-to-end simulation of 5g mmwave networks. *IEEE Communications Surveys & Tutorials*.

Nikaein, N., Schiller, E., Favraud, R., Knopp, R., Alyafawi, I., & Braun, T. (2017). Towards a cloud-native radio access network. In *Advances in mobile cloud computing and big data in the 5g era* (pp. 171–202). Springer.

Parvez, I., Rahmati, A., Guvenc, I., Sarwat, A. I., & Dai, H. (2018). A survey on low latency towards 5g: Ran, core network and caching solutions. *IEEE Communications Surveys & Tutorials*.

Pi, Z., & Khan, F. (2011). An introduction to millimeter-wave mobile broadband systems. *IEEE communications magazine*, *49*(6).

Qian, P., Wang, N., & Tafazolli, R. (2018). Achieving robust mobile web content delivery performance based on multiple coordinated quic connections. *IEEE Access*, *6*, 11313–11328.

Raaf, B., Zirwas, W., Friederichs, K.-J., Tiirola, E., Laitila, M., Marsch, P., & Wichman, R. (2011). Vision for beyond 4g broadband radio systems. In *Personal indoor and mobile radio communications (pimrc), 2011 ieee 22nd international symposium on* (pp. 2369–2373).

Ricudis, C. (2017, December). *[srslte-users] suggest opensource epc.* Retrieved from `http://www.softwareradiosystems.com/pipermail/srslte-users/2017-December/001261.html`

Rupprecht, D., Kohls, K., Holz, T., & Pöpper, C. (n.d.). Breaking lte on layer two. In *Breaking lte on layer two* (p. 0).

Saguna, & Intel. (2016). *Using mobile edge computing to improve mobile network performance and profitability* (Tech. Rep.).

Samdanis, K., Costa-Perez, X., & Sciancalepore, V. (2016). From network sharing to multi-tenancy: The 5g network slice broker. *IEEE Communications Magazine*, *54*(7), 32–39.

Satyanarayanan, M., Bahl, V., Caceres, R., & Davies, N. (2009). The case for vm-based cloudlets in mobile computing. *IEEE pervasive Computing*.

Schaffrath, G., Werle, C., Papadimitriou, P., Feldmann, A., Bless, R., Greenhalgh, A., ... Mathy, L. (2009). Network virtualization architecture: Proposal and initial prototype. In *Proceedings of the 1st acm workshop on virtualized infrastructure systems and architectures* (pp. 63–72).

Shaik, A., Borgaonkar, R., Asokan, N., Niemi, V., & Seifert, J.-P. (2015). Practical attacks against privacy and availability in 4g/lte mobile communication systems. *arXiv preprint arXiv:1510.07563*.

Sharifi, M., Kafaie, S., & Kashefi, O. (2012). A survey and taxonomy of cyber foraging of mobile devices. *IEEE Communications Surveys & Tutorials*, *14*(4), 1232–1243.

Sherry, J., Hasan, S., Scott, C., Krishnamurthy, A., Ratnasamy, S., & Sekar, V. (2012). Making middleboxes someone else's problem: network processing as a cloud service. *ACM SIGCOMM Computer Communication Review*, *42*(4), 13–24.

Shi, W., Cao, J., Zhang, Q., Li, Y., & Xu, L. (2016). Edge computing: Vision and challenges. *IEEE Internet of Things Journal*, *3*(5), 637–646.

Simsek, M., Aijaz, A., Dohler, M., Sachs, J., & Fettweis, G. (2016). The 5g-enabled tactile internet: Applications, requirements, and architecture. In *Wireless communications and networking conference (wcnc), 2016 ieee* (pp. 1–6).

Suznjevic, M., Slivar, I., & Skorin-Kapov, L. (2016). Analysis and qoe evaluation of cloud gaming service adaptation under different network conditions: The case of nvidia geforce now. In *Quality of multimedia experience (qomex), 2016 eighth international conference on* (pp. 1–6).

Taleb, T., Samdanis, K., Mada, B., Flinck, H., Dutta, S., & Sabella, D. (2017). On multi-access edge computing: A survey of the emerging 5g network edge cloud architecture and orchestration. *IEEE Communications Surveys & Tutorials*, *19*(3), 1657–1681.

Tran, T. X., Hajisami, A., Pandey, P., & Pompili, D. (2017). Collaborative mobile edge computing in 5g networks: New paradigms, scenarios, and challenges. *IEEE Communications Magazine*, *55*(4), 54–61.

Trivisonno, R., Guerzoni, R., Vaishnavi, I., & Soldani, D. (2015). Towards zero latency software defined 5g networks. In *Communication workshop (iccw), 2015 ieee international conference on* (pp. 2566–2571).

Tullberg, H., Popovski, P., Li, Z., Uusitalo, M. A., Hoglund, A., Bulakci, O., … Monserrat del Río, J. F. (2016). The metis 5g system concept: Meeting the 5g requirements. In *Ieee communications magazine* (Vol. 54, pp. 132–139).

Wang, Z., Qian, Z., Xu, Q., Mao, Z., & Zhang, M. (2011). An untold story of middleboxes in cellular networks. In *Acm sigcomm computer communication review* (Vol. 41, pp. 374–385).

Whitman, M. E., & Mattord, H. J. (2011). *Principles of information security*. Cengage Learning.

Wood, G. (2014). Ethereum: A secure decentralised generalised transaction ledger. *Ethereum project yellow paper*, *151*, 1–32.

Yi, S., Hao, Z., Qin, Z., & Li, Q. (2015). Fog computing: Platform and applications. In *2015 third ieee workshop on hot topics in web systems and technologies (hotweb)* (pp. 73–78).

Zaostrovnykh, A., Pirelli, S., Pedrosa, L., Argyraki, K., & Candea, G. (2017). A formally verified nat. In *Proceedings of the conference of the acm special interest group on data communication* (pp. 141–154).

Zhang, Q., Liu, L., Pu, C., Dou, Q., Wu, L., & Zhou, W. (2018). A comparative study of containers and virtual machines in big data environment. *arXiv preprint arXiv:1807.01842*.

Zheng, F., Whitted, T., Lastra, A., Lincoln, P., Maimone, A., Fuchs, H., et al. (2014). Minimizing latency for augmented reality displays: Frames considered harmful. In *Mixed and augmented reality (ismar), 2014 ieee international symposium on* (pp. 195–200).