# LL-MEC: Enabling Low Latency Edge Applications

Navid Nikaein, Xenofon Vasilakos and Anta Huang
Communication Systems Department, EURECOM, Biot, France 06410
firstname.lastname@eurecom.fr

***Abstract*—We present *LL-MEC*, the *first open source* Low-Latency Multi-access Edge Computing (MEC) platform enabling mobile network monitoring, control, and programmability while retaining compatibility with 3GPP and ETSI specifications. LL-MEC achieves *coordinated resource programmability* in end-to-end slicing scenarios by leveraging SDN towards an appropriate allocation of resources, thus drastically improving the performance of slices. We evaluate LL-MEC in three practical case studies, namely, (i) end-to-end mobile network slicing, (ii) RAN-aware video content optimization and (iii) IoT gateway, and show that it achieves a *2-4x lower* user plane *latency* compared to LTE, while enabling low latency edge applications to operate on a per *millisecond* basis. Also, we highlight the benefits of RAN-aware applications in improving user Quality of Experience (QoE), showing a significant user latency reduction along with a much lower variability compared to legacy LTE. Last, a compatibility evaluation of LL-MEC over the OpenAirInterface real-time LTE platform demonstrates the *scalability* merits of LL-MEC due to the use of an OpenFlow Virtual Switch for the user plane function, rather than a Linux kernel in typical LTE setups.**

***Index Terms*—MEC, SDN, Programmability, LTE**

## I. Introduction

One of the key challenges towards a 5G era in mobile networking is the ever-increasing demand for resource-hungry, content-rich services such as HD video streaming and augmented reality, which require both low latency and high reliability. Another challenge stems from the Internet of Things (IoT) use cases, which demand throughput provisioning, reliability, and low-latency connectivity among a large number of devices. Rather than redesigning the architecture, several popular solutions try to address these challenges via network slicing [1]–[3] and IoT gateway [4]. In this context, the ETSI-specified Multi-access Edge Computing (MEC) provides a low latency cloud environment for applications at the network's edge to monitor and control the underlying networks in close proximity with the users, hence posing a remedy for the aforementioned challenges. As an example, video streaming requests by User Equipment (UE) can be served via MEC-hosted applications, as MEC can program data paths and redirect traffic to local or remote serving nodes that improve the perceived user experience in a totally transparent manner.

MEC is characterized by its proximity to the Radio Access Network (RAN) as well as for providing real-time access to radio network information to applications. This feature, in particular, highlights *low latency* as a key point in MEC, while Multi-RAT connectivity provides interoperability and coordination to cater the needs of different access technologies through appropriate network abstraction, enabling a unified User Plane (UP) convergence that is reflected in the term *"Multi-access"*. Last, besides its technical benefits, MEC opens the network to authorized third-parties who can rapidly deploy innovative applications, thus creating a new market and an unprecedented value chain in mobile networking.

978-1-5386-6831-3/18/$31.00 

Considering that *programmability* is a key MEC requirement, Software-Defined Networking (SDN) poses a promising approach that is already extensively used in non-mobile networks. Along these lines, we can leverage the well-defined OpenFlow [5] SDN protocol for moving the *Control Plane (CP)* away from physical devices and for abstracting the underlying infrastructure, creating an unparalleled series of innovation and customization opportunities of network applications. Also, the success of SDN in non-mobile networking gives the right motivation to apply SDN in the *Core Network (CN)* of LTE [6]. By *separating the CP from the UP*, SDN virtualizes the mobile network components such as the Mobility Management Entity (MME), the Control planes of the Serving-Gateway (S-GW-C) and the Packet-Gateway (P-GW-C) as potential MEC applications. In essence, MEC can leverage CN *programmability* and further extend it in the RAN segment to further delegate control decisions.

### A. Contribution

A considerable research interest on SDN and MEC has focused on conceptual frameworks, yet *in absence* of an open source platform for evaluating the benefits of SDN-enabled MEC services. This *motivates us* to explore and demonstrate coordinated network programmability through our MEC platform and an ecosystem of network applications running on top of it. The main points of our contribution can be listed as:

**1) LL-MEC:** We contribute the *first* open source 3GPP-compliant implementation of a *Low Latency Multi-access Edge Computing (LL-MEC)* platform that covers multiple APIs aligned to ETSI MEC specifications. LL-MEC is an extended and concrete implementation of [7], using the extended OpenFlow [5] and FlexRAN [8] protocols, and addressing three types of latency: (i) *User latency*, defined as an end-to-end user transport latency; (ii) *Control latency*, which captures the latency of the underlying network MEC to perform an action on behalf of an edge application, e.g. for control and/or monitoring; and (iii) *Application latency*, which represents the latency for performing MEC actions by edge applications.

**2) Network slicing:** We enable network slicing in both the *edge* and *core* network segments by sharing physical resources across multiple logically isolated networks. Building upon MEC and SDN, LL-MEC achieves a *coordinated* resource

(radio resources, switching bandwidth) and UP programmability for end-to-end slicing, hence drastically improving performance through appropriate resource allocation.

**3) Platform evaluation & validation:** We evaluate the performance of LL-MEC in three different *practical case studies*: (i) end-to-end mobile network slicing; (ii) RAN-aware video content optimization; and (iii) IoT gateway featuring dedicated core network (DCN), paving the way for the 5G community to engage into new investigation directions via our platform in further case studies. In a nutshell, our thorough results validate that LL-MEC yields a *significant* user *latency reduction* along with a much lower variability compared to legacy LTE. Also, a compatibility evaluation over the OpenAirInterface (OAI) real-time LTE platform [9] proves that LL-MEC exhibits significant scalability merits against a massive number of UEs, due to the Open Virtual Switch (OVS) used for the UP functionality, rather than a Linux kernel in legacy LTE setups.

### B. Paper structure

We discuss the implementation of the LL-MEC platform for SDN-based mobile networks in Sec. II, providing coordinated CP and UP programmability. In Sec III, we provide a compatibility assessment of LL-MEC over the OAI [9], using Commercial Off-The-Shelf (COTS) UEs. Also, we present a system performance assessment in terms of CPU load and traffic latency, showing significant user latency gains and good scalability features. Next, Sec. IV presents a thorough evaluation of the LL-MEC platform in three practical case studies Finally, Sec. V outlines the related work before we conclude and discuss our future work goal in Sec. VI.

## II. LL-MEC Design & Implementation

This section provides an overview of the architecture and identifies the design challenges for realizing a low latency MEC platform. Fig. 1 portrays that the MEC application manager stands as the foundation for the upper-most layers, providing the (Mp1) programming interface for applications. The middle layer includes two *core* components, namely, the Radio Network Information Service (RNIS) and Edge Packet Service (EPS) that manage the RAN and CN network services, respectively, based on the CP and UP APIs in the abstraction layer, respectively. Standing at the bottom-most layer, eNBs and OpenFlow-enabled switches comprise the UP functions. FlexRAN and OpenFlow comprise the CP functions and abstract all information and expose it via the abstraction API (Mp2). The platform operates on a software-defined mobile network consisting of multiple LTE eNBs and physical or software OpenFlow-enabled switches. Following the separation of CP from UP, we adopt "*X*-GW-C" and "*X*-GW-U" as the corresponding notations for 4G service (S-GW) and packet (P-GW) gateways and 5G session management function (SMF) and user-plane function (UPF). Note that According to ETSI specifications the Mp1 and Mp2 reference points comprise interfaces between the different layers.

A key point in LL-MEC relies on its software-development kit (SDK) providing a *unified applications development* environment that allows to apply coordinated control decisions across the RAN and CN segments. The FlexRAN and OpenFlow abstraction protocols used for the RAN and the CN, respectively, facilitate communication among the different network elements. Along with their corresponding APIs, FlexRAN and OpenFlow are integrated in LL-MEC to enable a two-way interaction between them, hence allowing to fulfill requests from a limitless set of edge applications and to execute precise tasks onto the underlying networks. Besides this, LL-MEC is designed to support *time critical RAN* operations and the deployment of different priority-level applications when interacting with the platform.

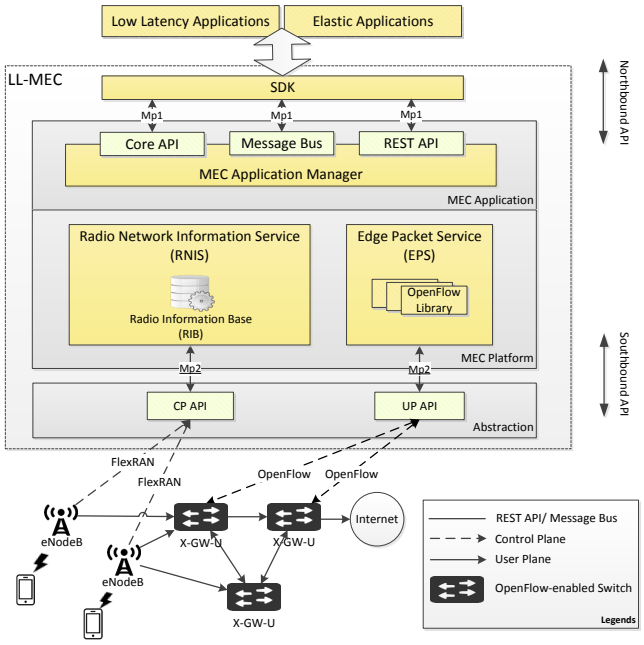



Fig. 1: High-level schematic of LL-MEC.

### A. Design Challenges

A first key design challenge for realizing LL-MEC refers to the separation of the CP from the UP throughout the RAN and the CN, whereas a second challenge regards the coordinated CP and UP programmability across the RAN and the CN with real-time access to RAN information. Another challenge regards scalability with the large number of users and services and, finally, a fourth challenge refers to the flexibility of registering low latency applications and services in order to support time-critical control decisions, priorities and deadlines.

### B. Mobile Network Abstraction

The abstraction layer models the required operations for the underlying network through a unified interface. In LL-MEC, the CP API and UP API comprise the abstraction layer for RAN and CN, respectively, by providing the necessary information for MEC platform and applications development. LL-MEC control protocols are divided into: (i) RAN enabled by FlexRAN [8] and (ii) CN through OpenFlow [5]. FlexRAN provides an abstract view of the radio network status (e.g., signal strength) by extracting parameters with a required granularity level. Also, it enables to modify and control the

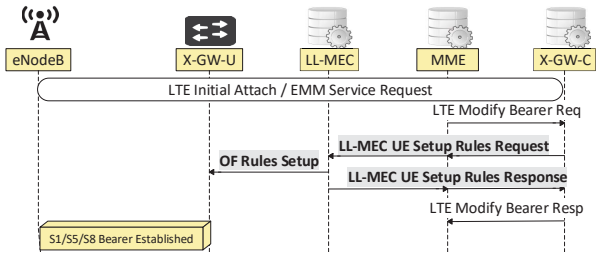



Fig. 2: Sequence diagram of S1/S5/S8 bearer setup.

state of the underlying network, passing control decisions per subframe, e.g. for reconfiguring resource blocks for UEs. OpenFlow, provides a fine-grain programmable UP through the abstraction of the underlying data paths, allowing the switch to handle GPRS Tunneling Protocol (GTP) packets in the CN.

### C. Edge Packet Service

*Edge Packet Service* (EPS) is a main component, bringing a native IP-service end-point for MEC applications to meet a specific purpose. For example, UE incoming traffic is routed after the rules setup in the switches by EPS and can be altered dynamically to optimize routing. Being a core LL-MEC entity, EPS offers the *interfaces* towards its *northbound* and *southbound*, namely, *Mp1* and *Mp2* in Fig. 1.

**Mp1** is the interface for MEC applications to instruct basic and advanced functionalities in the underlying network such as default/dedicated bearers (re-)establishment, QoS for GBR traffic and specific requests. Considering LTE legacy compatibility, Fig. 2 shows the relevant messaging for the S1/S5/S8 bearer establishment procedure. Upon issuing a *Modify Bearer Request* by either the *LTE attach procedures* or *EPS Mobility Management Service Request*, X-GW-C notifies LL-MEC for bearer establishment via a *UE Setup Rules Request*, allowing it to trigger OpenFlow rules for setting up the switch. When X-GW-C receives a *UE Setup Rules Response*, the bearer establishment is confirmed. The message for calling the *UE Setup Rules Request* API must include user identities like uplink/downlink tunnel ID and bearer ID. Likewise is the procedure for supporting QoS for GBR traffics through OpenFlow meter and group tables.

**Mp2** is the interface for MEC applications to instruct the UP on routing traffic via OpenFlow rules. The types of rules that EPS maintains in OpenFlow handler (Fig. 1) are: (i) *default rules* pushed to OpenFlow-enabled switches on connection establishment for handling Address Resolution Protocol (ARP) and Domain Name System (DNS) queries; (ii) *UE specific rules* for establishing the default and dedicated bearers for UE; (iii) and *MEC application rules* pushed to OpenFlow-enabled switches on events registered by applications. With a well-defined full set of rules, UP gets fully separated from CP, thus improving user latency (Sec. II-A and III-B).

### D. Radio Network Information Service

Specified by ETSI MEC[1] RNIS in LL-MEC exposes real-time RAN information to MEC applications such as radio bearer statistics, UE-related measurements and state changes, or power measurements, by interacting with CP API. It is possible to adjust the granularity of information per cell, UE or radio access bearer, and to request information once, periodically or when an event triggered. The CP API defines a set of functions used by the UP to notify the CP about events such as the initiation of a new Transmission Time Interval (TTI). In order to have a clean separation of RAN CP and UP, the FlexRAN protocol and the RAN Information Base (RIB) [8] are integrated to LL-MEC. FlexRAN acts as an abstraction layer allowing the management of the higher-level control operations in a technology-agnostic way likewise to how OpenFlow abstracts the datapath in the wired network. The RIB maintains all statistics and configuration details about the RAN that are accessed by applications. Furthermore, RNIS can have direct and high priority access to the RIB on per millisecond basis to ease control latency due to the integration of RIB in LL-MEC. Thus, an edge application can, e.g., query each user link quality to provide a quasi real-time throughput indication in the next time window.

### E. MEC Application Framework

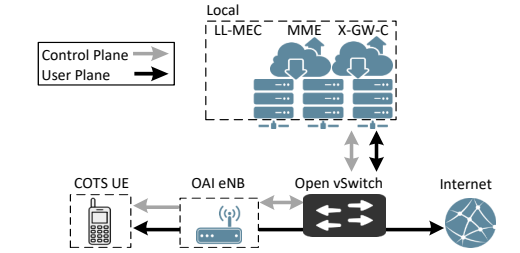
MEC applications can be developed for any purpose and without a detailed knowledge of the underlying network due separating CP from UP. Mp1 and the SDK built on top of it facilitate a programming environment, with the SDK offering a uniform interface and means for platform communication while abstracting the multiple choices of Mp1 including a REST API, a message bus and a local API for different application requirements. Examples include monitoring and acquiring information through the message bus, managing traffic rules via the REST API within 100 ms, or optimizing content based on radio quality within a single ms. Applications can also access basic LL-MEC functionalities through Mp1 such as a service registry and an event mechanism [7]. Another pivotal LL-MEC feature regards its different application scheduling recipes like round-robin or deadline-based for adapting different task priorities. This *significantly* lowers application latency and meets with control deadlines.

## III. System Evaluation

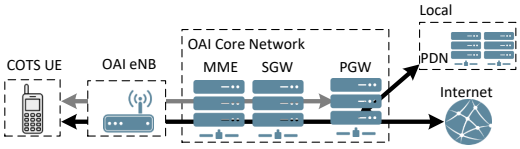
We deploy a LL-MEC platform with one and multiple OpenFlow-enabled switches, using Open vSwitch v2.5.1 under OpenFlow protocol v1.3 for handling UP traffic.[2] In order to have GTP tunneling functionality, we applied an OVS patch to the Open vSwitch 2.5.1 implementation. It is, however, important to note that a physical switch with OpenFlow and GTP supports can also work with LL-MEC. Fig. 3 illustrates two different setups: SDN-based LTE with LL-MEC and legacy LTE. All components, such as eNB, CN, Open vSwitch and LL-MEC run on a commodity Linux-based PC equipped with dual-core i5-661 CPU at 3.3GHz and 4GB of RAM. Depending on the experiment, LL-MEC is connected either to a massive S1-U packet generator via Gigabit Ethernet or

[1]ETSI GS MEC 002 MEC;Technical Requirements

[2]http://mosaic-5g.io/ll-mec/




(a) Setup for LL-MEC in SDN-based LTE.




(b) Setup for legacy LTE.

Fig. 3: The evaluation setup.

to OpenAirInterface (OAI) LTE eNB with a radio front-end (Ettus B210 USRP) and COTS UEs (Nexus 6p and HUAWEI E392 4G LTE dongle). The massive S1-U packet generator is based on the Python Scapy library to craft and send customized GTP packets with different packet sizes and inter-departure times down to every ms. The massive S1-U packet generator can send traffic with GTP encapsulated up to 10000 UEs simultaneously. This way, we assess LL-MEC user latency compared to legacy LTE under high-load conditions. We conduct two types of experiments: *(i) Compatibility* with measurements taken in a real LTE network with COTS UEs to evaluate the throughput performance; and *(ii) Scalability* with measurements taken in a real LTE CN with generated UE-to-eNB traffics to evaluate the benefits of CN offloading when redirecting both the CP and UP traffic.

### A. Compatibility

We setup two testbeds: (i) an SDN-based LTE network with LL-MEC deployed and (ii) a legacy LTE network (Fig. 3). We use the OAI in both setups as a real-time 3GPP compliant LTE environment for attaching one COTS UE over the air. OAI allows to verify that our SDN-based LL-MEC can operate with full LTE functionality, thus providing the related latency measurements. All measurements use the same eNB configuration, namely, FDD with transmission mode 1 (SISO) and 5 MHz channel bandwidth in band 7. In addition, in SDN-based LTE setup LL-MEC requires the user identities along with appropriate rules to be applied onto the OpenFlow-enabled switches for establishing the UP.

TABLE I: Throughput with 5MHz channel bandwidth

| | Setup | Mean (MB/s) | Std. dev. | Min | Max |
|---|---|---|---|---|---|
| Downlink | Legacy LTE | 15.691 | 1.648 | 11.5 | 18.9 |
| | LL-MEC | 15.112 | 0.67 | 14.9 | 16.7 |
| Uplink | Legacy LTE | 8.214 | 1.059 | 4.19 | 11.5 |
| | LL-MEC | 8.197 | 0.644 | 7.34 | 9.44 |

Table I shows throughput recorded with *iperf* over a 60 sec period on a per second measurement basis. COTS UEs in either setup have full Internet access reaching maximum throughput (15MB/s in downlink and 8Mb/s in uplink) for a 5MHz channel bandwidth. We observe a *better stability* in both downlink and uplink throughput (lower standard deviation) for LL-MEC due to the separation of CP and UP.

### B. Scalability

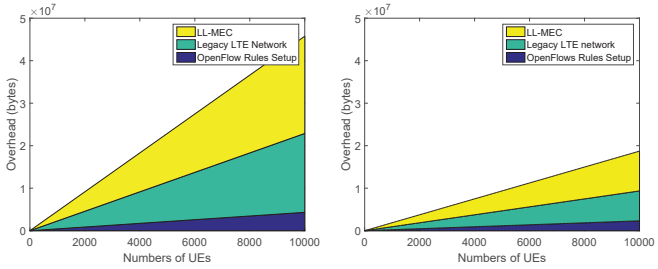
We study LL-MEC scalability with the number of UEs. In what follows, we consider both CP and UP scalability.

**Control Plane:** Establishing default and dedicated bearers in LL-MEC (see Sec. II-C) takes place through an interaction among X-GW-C, EPS and switches. This induces *extra control signaling* compared to the establishment of UP in legacy LTE (e.g., tunnel end-points setup and iptables) due to UE, QoS and OpenFlow setup rules. We measure the total payload in terms of transmitted bytes [3] used to establish default and dedicated bearers as a function of the number of UEs for both LL-MEC and the legacy LTE network. To assess the total signaling overhead, the X-GW-C is placed outside the LL-MEC platform. In addition, we characterize the contribution of OpenFlow setup rules to the total overhead. Fig. 4 shows that LL-MEC introduces a signaling overhead for both default and dedicated bearer establishment due to the messages for setting up UE-specific, QoS and OpenFlow rules between X-GW-C, EPS and switches (see Fig. 2 for the message exchanges). This is the cost of providing UP programmability towards applications. However, this overhead can be significantly reduced if X-GW-C is deployed as a service on top of LL-MEC, in which case the traffic of S11 and S5/S8 can be transmitted locally. Thus, the only remaining overhead is for OpenFlow setup rules from EPS to OpenFlow-enabled switches, which significantly lowers signaling (see Fig. 4, labeled as *OpenFlow rules setup*).

**User Plane:** We deploy a massive S1-U packet generator to transmit a large number of GTP encapsulated ICMP packets in order to determine the load of SDN-based core entities. We generate traffic as soon as the generator and MME are instantiated to gradually increase the number of UEs as shown in Table II. LL-MEC traffic originates from different numbers of UEs every 100 ms with 1400KB payload. The Round Trip Time (RTT) is measured upon receiving the *Echo-Reply* ICMP packet. Our setup is likewise to the one in Fig. 3, only the massive S1-U packet generator acts as both UEs and eNBs sending GTP traffic over the S1-U interface directly through the wired network. For the CN, we use OAI with the *MME scenario player* feature, allowing the network to emulate the attachment procedure of a large number of UEs.

Figure 5(a) shows that S-GW CPU load increases drastically, reaching ∼50% for 100 UEs contrary to LL-MEC for which it remains 5%. CPU usage is measured for the entire S-GW machine, with a 50% of CPU resource usage translating to a single fully-loaded processing core. Therefore, when the number of UEs reaches to 5000 the S-GW is already overloaded, contrary to only ∼6% in the case of LL-MEC. This is because the OAI S-GW-C implementation is a single-threaded process, thus it cannot scale with the number of available cores. Nevertheless, the S-GW CP is not the real bottleneck, as it is only in charge of the establishment

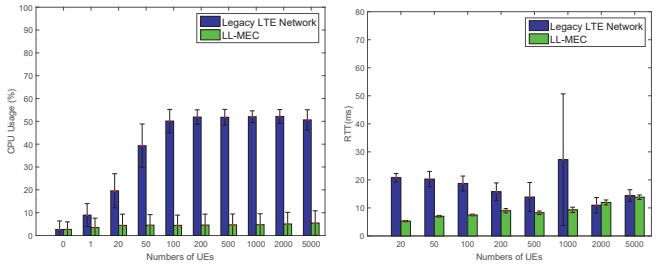
[3] Sum of each message payload related to LL-MEC, LTE, and OpenFlow.




(a) Default bearer (b) Dedicated bearer

Fig. 4: Control Signaling Overhead: colored regions show the contribution of LL-MEC (yellow), legacy LTE network (green) and OpenFlow rules setup (purple) to the total overhead.

of the UP. Data traffic is actually handled by the Linux GTP kernel module, its corresponding library and iptable rules used in OAI. Any observed performance loss is mainly due to the Netfilter [10], which is the main cause of CPU overloading. This result *reveals the benefits of CP and UP separation*, as the CPU overhead for context switching can be avoided to drastically improve scalability and performance, thus highlighting the benefits of MEC in terms of dynamic traffic offloading, scalability and resource efficiency.




(a) CPU usage (b) Latency

Fig. 5: CPU usage and Latency against the number of UEs.

Regarding RTT measurements, Fig. 5(b) indicates that LL-MEC reduces user latency significantly and with a much lower variability compared to legacy LTE. Also, LL-MEC latency is ∼5ms for 20 UEs and ∼12ms for 5000 UEs with 0% packet drop rate. Nevertheless, legacy LTE latency exhibits a spike for 1000 UEs, i.e during the time when a massive packet drop takes place (see Table II) due to CPU overloading (see Fig. 5(a) and 5(b) together). This implies that processing each packet in the legacy LTE setup requires more computing resources than LL-MEC and poses scalability issues in general. The outcome of this part of our evaluation is that the SDN-based design of LL-MEC lowers UP latency by a factor of 2 on average and improves performance as an overall.

TABLE II: ICMP packet drop rate per number of UEs.

| #UEs | 20 | 50 | 100 | 200 | 500 | 1000 | 2000 | 5000 |
|---|---|---|---|---|---|---|---|---|
| Legacy LTE | 0% | 0% | 0% | 0% | 1.72% | 7.86% | 77.13% | 77.09% |

## IV. Case Studies

For all of the considered use cases presented next, we rely on the OAI platform as an evolved LTE platform and keep the same configuration as presented in Fig. 3.

### A. End-to-End Mobile Network Slicing

Network *slicing* is a key enabler for sharing physical network resource across multiple logically isolated networks in 5G, hence supporting a wide range of vertical segments with a diverse set of performance and service requirements. LL-MEC is a platform that enables slicing for achieving isolation and performance guarantee in the UP by leveraging partially the 3GPP enhanced dedicated core network (eDECOR)[4] concept. A network *slice* is essentially *a group of UEs* having the *same requirements* or belonging to the *same administrative domain*, with *no* traffic or policy differentiation within a slice.

Figure 6 shows how LL-MEC enables network programming for creating *two network slices*; one that is served locally by MEC and gets a higher over-the-air performance, and another one that gets a best-effort performance via direction to a back-end server. We use an *over-the-air* LL-MEC in SDN-based LTE. Slices are created with 1 COTS UE each, while the percentage of radio resources and switching bandwidth for each slice follows the corresponding slice-specific policies.

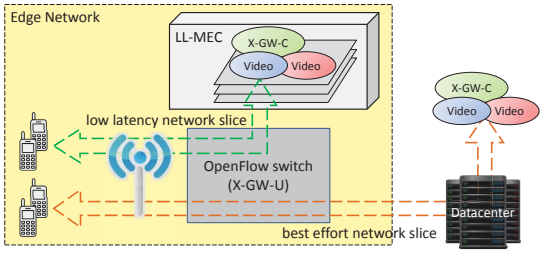
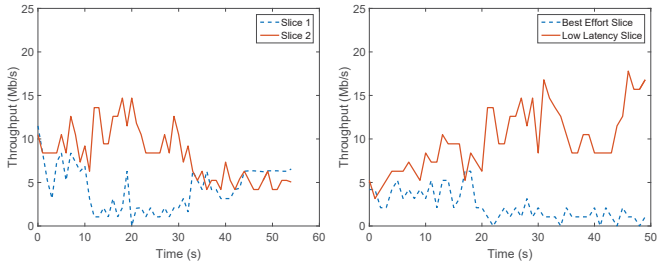



Fig. 6: Overview of end-to-end network slicing.

We design a *slice policy enforcement algorithm* to apply different RAN resource allocation strategies, and implement a *low latency* MEC application interfacing with the LL-MEC platform through the SDK. The EPS does dynamic routing management at the edge through the SDK, while real-time control can be delegated back to the RAN. To demonstrate the benefits of end-to-end slicing, we change the enforced policy on-the-fly and measure the resulted downlink throughput. Specifically, we consider an *uncoordinated* and a *coordinated* end-to-end resource programmability scheme in terms of radio resources for the RAN and switching bandwidth for the CN parts, with corresponding performance results appearing in Fig. 7. For the case of uncoordinated programmability we enforce a policy at time t=10s that implies 1 Mbps for slice 1 and 15 Mbps for slice 2. Then, we apply a second policy at t=20s, this time *only to RAN* in order to lower the rate down to 50% of radio resources (i.e., 8 Mbps) for each slice. Finally, we enforce a third policy *only to CN* at t=33s to increase the switching bandwidth to 6 Mbps. For coordinated programmability, however, only one policy is enforced at t=18s to both the RAN and the CN, so as to create a best-effort slice with 1 Mbps and a low latency slice with 15 Mbps.

The results *confirm the benefits of MEC and unified SDK* for enabling coordinated programmability and network slicing. For uncoordinated slicing (Fig. 7(a)), bandwidth is *not* used

[4]3GPP 23.707 Release 13; Stage 2 (2014); 3GPP 23.711 Release 14 (2015).




(a) Uncoordinated slicing (b) Coordinated slicing

Fig. 7: Mobile network slicing.

efficiently due to the *asynchronous/uncoordinated resource allocation* between RAN and CN. For coordinated slicing (Fig. 7(b)), however, the anticipated performance gap between the "Low-Latency Slice" and the "Best-Effort Slice" is evident, while resources get appropriately allocated to each slice in accordance to their specific requirements.

### B. RAN-aware Video Content Optimization

We consider video optimization as a low latency application and study the benefits of RAN-aware applications on improving user QoE. We monitor the cell load status and radio link quality obtained from RNIS in order to adjust content quality (e.g., via transcoding) at the server, parallel to enforcing a new resource allocation policy to the underlying RAN. This allows to jointly improve both network efficiency and user QoE (e.g., by avoiding buffer freeze). Note that in this use case we use a *low latency network slice*, as previously described in Fig. 6.

TABLE III: CQI index mapped as max TCP throughput

| CQI | Downlink (Mb/s) | Uplink (Mb/s) |
|---|---|---|
| 15 | 15.224 | 8.08 |
| 11 | 11.469 | 6.04 |
| 9 | 9.88 | 4.47 |
| 7 | 5.591 | 2.49 |
| 4 | 1.08 | 0.69 |

We implement a simple HTTP video streaming application on top of LL-MEC and the choose channel quality indicator (CQI) as a flag to reflect radio link quality for each UE. When a UE accesses the video service, the LL-MEC can (i) program the routing path and redirect traffic to one of the MEC applications if the requested service is matched, e.g., based on the destination IP address), and (ii) adapt the rate according to the estimated UE throughput. There are multiple approaches to throughput guidance on the top of the LL-MEC RNIS module like exponential moving average or even a discrete link-quality to throughput mapping. Table III shows the maximum video TCP bitrate through a discrete mapping between CQI and a sustainable TCP throughput identified during experiments. This value serves as a predicted user throughput allowing the server to adjust transcoding. As a follow-up to Sec. IV-A, coordinated slicing and joint programmability managed by authorized MEC applications results in an effective mobile network. Also, note that the timescale of detecting CQI changes is significantly less than the one in the TCP congestion mechanism. Instead of a reactive, a proactive adaptation of the service demand is also feasible through RNIS.

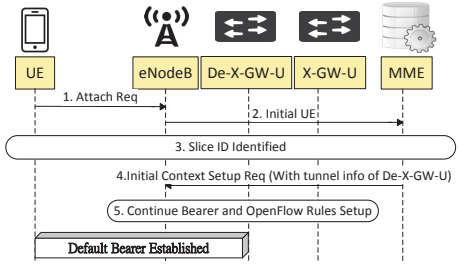



Fig. 8: Sequence diagram of DCN

### C. IoT Gateway

In this last case study we consider LL-MEC as a platform to deploy an IoT Gateway at the edge, leveraging again (see Sec. IV-A) the eDECOR concept for CN slicing. Fig. 8 portrays a simplified sequence diagram on how user traffic is directed to a dedicated X-GW-U (De-X-GW-U) in LL-MEC based on slice IDs. Upon the reception of an attachment request containing the slice ID, MME/X-GW-C maps the UE slice ID (stored in HSS) to the De-X-GW-U and initiates a set of OpenFlow rules for this newly instantiated switch. Then, the tunnel information setup for De-X-GW-U is included in a *Initial Context Setup Request* sent to the eNB. At this point, the dedicated user UP is established between the eNB and the corresponding switch. We consider 2000 devices equally split into two slices of a 1000 devices each. We use our massive S1-U packet generator for sending sensory data to dedicated switches in accordance to device slice IDs. Latency measurement shown in Fig. 9 for both with and without slicing indicate that with a dedicated UP we can achieve not only traffic isolation and scalability, but also improve performance greatly by lowering traffic latency and the variability.

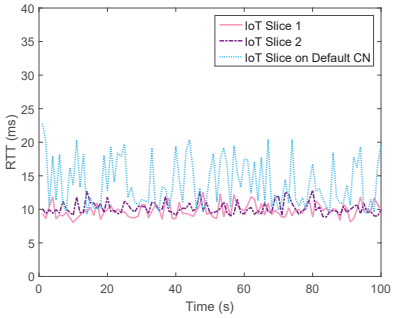



Fig. 9: Latency measurements of isolated IoT slices.

## V. Related Work

MEC attracts a considerable research interest [11]–[14] form academy and industry with some specifications completed and work in progress. Initially, ETSI presented the MEC ecosystem and main service scenarios in [15] to provide a cloud computing environment for applications and content in close proximity to the RAN. In addition, several MEC services are proposed to offload tasks from mobile devices to further reduce power consumption [16], [17]. The work of [18] proposes a hierarchical MEC architecture leveraging cloud computing and migrating mobile workloads for remote execution at the cloud. A comparison among MEC, fog computing and Cloudlet can be found in [19] and a complete

conceptual MEC architecture considering full functionalities, interfaces, and applications in [20]. Likewise to ETSI MEC, Cloudlet aims to provide computing resources at the networks' edge. However, it is loosely-coupled with the underlaying network, as it does not specify interfaces and data models for the applications to interact with the network.

SDN is a building block of MEC featuring the decoupling of CP and UP, the consolidation of the CP, and network programmability through well-defined APIs. SDN came with the invention of the OpenFlow concept [5] and has been extensively used in wired networking. Using SDN for the CN of LTE is an intuitive first step, with much work exploring this concept [21]–[23] in mobile networks from different aspects such as scalability, 3GPP interoperability, and performance evaluation. SoftRAN [24] is a centralized CP for RAN that abstracts all base stations into a virtual big base station. FlexRAN [8] provides a flexible CP to build real-time RAN control applications and remains flexible to realize different degrees of coordination among RAN infrastructure entities.

## VI. Conclusion and Future Work

We propose LL-MEC, a low-latency MEC platform that exploits SDN to facilitate low-latency edge-routed user traffic flows in mobile networks. Towards a desired performance, LL-MEC employs the required flexibility and programmability to *coordinate* decisions across different network segments while remaining compliant with 3GPP specifications and ETSI MEC ISG functionalities. Performance results reveal the benefits of LL-MEC in reducing user and application latency in three case studies, confirming its applicability in emerging IoT use cases, content optimization, and network slicing. Future work includes focusing on use cases deserving a more profound study such as policy and charging control, location-aware services and intelligent management towards a self-organizing MEC platform, inspired by machine learning and works like on congestion pricing [25]–[27] or [28]–[30]. Finally, LL-MEC will support an earlier version of OpenFlow with interesting features like meter action and extensible flow entry statistics.

## Acknowledgment

This work has been funded in part through the European Union's H2020 program under grant agreement No 761913: project SliceNet, and by the French Government (National Research Agency, ANR) through the "Investments for the Future" Program reference #ANR-11-LABX-0031-01.

## References

[1] N. Nikaein, E. Schiller, R. Favraud, K. Katsalis, D. Stavropoulos, I. Alyafawi, Z. Zhao, T. Braun, and T. Korakis, "Network store: Exploring slicing in future 5G networks," pp. 8–13, 2015.

[2] A. Ksentini and N. Nikaein, "Toward Enforcing Network Slicing on RAN: Flexibility and Resources Abstraction," *IEEE Communications Magazine*, vol. 55, no. 6, pp. 102–108, 2017.

[3] X. Foukas, M. Mahesh K., and K. Kontovasilis, "Orion: RAN Slicing for a Flexible and Cost-Effective Multi-Service Mobile Network Architecture," in *23rd Annual International Conference on Mobile Computing and Networking (MobiCom '17)*. ACM, 2017.

[4] M. Vögler, J. M. Schleicher, C. Inzinger, and S. Dustdar, "A scalable framework for provisioning large-scale IoT deployments," *ACM Transactions on Internet Technology (TOIT)*, vol. 16, no. 2, p. 11, 2016.

[5] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "OpenFlow: Enabling Innovation in Campus Networks," *SIGCOMM Comput. Commun. Rev.*, 2008.

[6] X. Jin, L. E. Li, L. Vanbever, and J. Rexford, "SoftCell: Scalable and Flexible Cellular Core Network Architecture," in *ACM Conference on Emerging Networking Experiments and Technologies*, 2013.

[7] A. Huang, N. Nikaein, T. Stenbock, A. Ksentini, and C. Bonnet, "Low Latency MEC Framework for SDN-based LTE/LTE-A Networks," in *Proc. of the IEEE International Conference on Communications*, 2017.

[8] X. Foukas, N. Nikaein, M. M. Kassem, M. K. Marina, and K. Kontovasilis, "FlexRAN: A Flexible and Programmable Platform for Software-Defined Radio Access Networks," in *Proc. of 12th International Conference on Emerging Networking EXperiments and Technologies*, 2016.

[9] Nikaein, Navid and Marina, Mahesh K. and Manickam, Saravana and Dawson, Alex and Knopp, Raymond and Bonnet, Christian, "Openairinterface: A flexible platform for 5g research," *SIGCOMM Comput. Commun. Rev.*, 2014.

[10] R. Niemann et al., "Performance Evaluation of netfilter: A Study on the Performance Loss When Using netfilter as a Firewall," *CoRR*, 2015.

[11] P. Mach and Z. Becvar, "Mobile edge computing: A survey on architecture and computation offloading," *CoRR*, 2017.

[12] A. Ahmed and E. Ahmed, "A Survey on Mobile Edge Computing," in *International conf. on intelligent Systems and COntrol (ISCO)*, 2016.

[13] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "Mobile edge computing: Survey and research outlook," *CoRR*, 2017.

[14] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE Internet of Things Journal*, Oct 2016.

[15] Y. C. Hu, M. Patel, D. Sabella, N. Sprecher, and V. Young, "Mobile edge computing a key technology towards 5g," *ETSI*, 2015.

[16] M. T. Beck, M. Werner, S. Feld, and T. Schimper, "Mobile edge computing: A taxonomy," in *Proceedings of International Conference on Advances in Future Internet*, 2014.

[17] M. T. Beck and S. Feld et al., "ME-VoLTE: Network functions for energy-efficient video transcoding at the mobile edge," in *18th International Conference on Intelligence in Next Generation Networks*, 2015.

[18] L. Tong, Y. Li, and W. Gao, "A hierarchical edge cloud architecture for mobile computing," in *Proceedings of the 35th Annual IEEE International Conference on Computer Communications*, 2016.

[19] R. Roman, J. Lopez, and M. Mambo, "MEC, Fog et al.: A Survey and Analysis of Security Threats and Challenges," *CoRR*, 2016.

[20] C.-Y. Chang, K. Alexandris, N. Nikaein, K. Katsalis, and T. Spyropoulos, "MEC Architectural Implications for LTE/LTE-A Networks," in *Workshop on Mobility in the Evolving Internet Architecture*, 2016.

[21] Kostas Pentikousis, Yan Wang and Weihua Hu, "Mobileflow: Toward Software Defined Mobile Networks," *IEEE Comm. Magazine*, 2013.

[22] M. Martinello, M. R. N. Ribeiro et al., "Keyflow: a prototype for evolving SDN toward core network fabrics," *IEEE Network*, 2014.

[23] Van-Giang Nguyen and Younghan Kim, "Signaling Load Analysis in Openflow-enabled LTE/EPC Architecture," in *Proc. of Intern. Conference on Information & Communication Technology Convergence*, 2014.

[24] A. Gudipati, D. Perry et al., "SoftRAN: Software Defined Radio Access Network," in *ACM SIGCOMM Workshop on Hot Topics in SDN*, 2013.

[25] V. A. Siris, X. Vasilakos, and G. C. Polyzos, "Efficient proactive caching for supporting seamless mobility," in *Proceeding of IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks, WoWMoM 2014, Sydney, Australia, June 19, 2014*, 2014, pp. 1–6.

[26] X. Vasilakos, V. A. Siris, and G. C. Polyzos, "Addressing niche demand based on joint mobility prediction and content popularity caching," *Computer Networks*, vol. 110, pp. 306–323, 2016.

[27] X. Vasilakos, M. Al-Khalidi, V. A. Siris, M. J. Reed, N. Thomos, and G. C. Polyzos, "Mobility-based Proactive Multicast for Seamless Mobility Support in Cellular Network Environments," in *Workshop on Mobile Edge Communications, MECOMM '17*. ACM, 2017, pp. 25–30.

[28] V. Giannaki, X. Vasilakos, C. Stais, G. C. Polyzos, and G. Xylomenos, "Supporting mobility in a publish subscribe internetwork architecture," in *Proceedings of the 16th IEEE Symposium on Computers and Communications, ISCC 2011, Kerkyra, Corfu, Greece, June 28 - July 1, 2011*, 2011, pp. 1030–1032.

[29] K. Poularakis and L. Tassiulas, "Code, Cache and Deliver on the Move: A Novel Caching Paradigm in Hyper-Dense Small-Cell Networks," *IEEE Trans. Mob. Comput.*, vol. 16, no. 3, pp. 675–687, 2017.

[30] S. Zhang, P. He, K. Suto, P. Yang, L. Zhao, and X. Shen, "Cooperative Edge Caching in User-Centric Clustered Mobile Networks," *IEEE Trans. Mob. Comput.*, vol. 17, no. 8, pp. 1791–1805, 2018.