A Hierarchical MEC Architecture: Experimenting the RAVEN Use-Case

D. Sabella
Intel Deutschland GmbH,
Munich, Germany

N. Nikaein, A. Huang
Eurecom,
Sophia Antipolis, France

J.Xhembulla, G. Malnati
Politecnico di Torino,
Turin, Italy

Salvatore Scarpina
TIM (Telecom Italia Group),
Turin, Italy

***Abstract*— Low latency communication with end-user and knowledge of real-time network information, such as radio conditions and network statistics, are among two key advantages of Multi-access Edge Computing (MEC) technology. In this paper, we propose a hierarchical MEC architecture and present a proof-of-concept (PoC) implementation of a Radio Aware Video optimization in a fully Virtualized Network (RAVEN) use case. The PoC has been assembled in a small-scale LTE network based on OpenAirInterface and commercial terminals, representing a scenario where a real-time adaptive video streaming service is provided at a Mobile Edge platform serving multiple eNBs. Real-time radio information is provided by eNB agents which act as local controllers co-located with eNB, collecting and providing realtime access to the requested RAN data. Agents can cooperate with each other to control the network in a distributed manner or can be delegated and/or controlled by a master controller entity. Information gathered by eNB agents is thus provided to the MEC system, allowing other MEC apps to consume it. The results of the RAVEN experimental activity demonstrate the benefit of recently ETSI MEC specified RNI (Radio Network Information) service in improving the perceived user quality of experience.**

***Keywords*— RAN, 5G, MEC, Multi-access Edge Computing, RNI API, Real-time radio information.**

I. INTRODUCTION

The evolution of mobile communications points to a spectacular growth of both the number of networked devices (8x smartphones, 15x tablets, triple digit for IoT devices) and mobile data traffic (8x) by 2020 [1]. This increasing traffic demand is also coupled with uneven distribution in time and space of such data sources and with heterogeneous devices, services and related QoS (Quality of Service) requirements. High bandwidth, ultra-low latency and real time access to radio information are the key features expected from next generation communication systems in order to enable a plethora of total new use cases: *consumer-oriented services*, like gaming, augmented/assisted reality, cognitive assistance; *third party services*, like active device location tracking, big data, security, safety, enterprise services; and *network performance services*, improving performance of the network, like content/DNS caching, performance optimization, and video optimization. In the architecture of new 5G systems, the edge of the network has a key role as convenient environment for hosting applications. In fact, while LTE network average latencies in some case can be limited to few tens of milliseconds (e.g. 38.4 ms according to [17]), usually E2E delay is actually much bigger since it is conditioned by the location of the end point of user application (which often is a server far away from the user).

In this scenario, Multi-Access Edge Computing (MEC) is gaining a growing interest from many stakeholders involved in joint standardization efforts in ETSI MEC ISG [2][3][4]. This technology is commonly considered as key ingredient of future 5G systems [5][6] that will provide operators with the flexibility and reconfigurability required to satisfy the increasing traffic demand. Low latency communication due to the proximity to end users and the availability of real-time network information (such as radio conditions and network statistics) are important advantages of MEC technology. These enablers can be realized by exploiting Radio Network Information (RNI) gathering through MEC framework and APIs [2][3][18], where context information from the RAN can be provided to user level applications or other services for network performance and QoS improvements.

In this paper, we introduce a hierarchical MEC architecture and report the proof-of-concept (PoC) implementation of the RAVEN ("Radio Aware Video optimization in a fully Virtualized Network") use case in the context of ETSI MEC standardization activities.

The rest of the paper is organized as follows. Section II describes the proposed hierarchical MEC architecture. In Section III, we describe the RAVEN use case and the main implementation aspects including LTE virtualized environment, the RNI API, and RAVEN apps. In Section IV, we show a subset of the results gathered during the experimentation. Finally, Section V provides the conclusion and highlights directions for future work.

II. HIERARCHICAL MEC ARCHITECTURE

This section gives a high-level overview of the MEC platform, the key contribution of this paper. We present the architecture in the context of LTE for concreteness and to match with its current implementation. As a consequence, we use the LTE terminology of eNodeBs and UEs, respectively for base stations and mobile devices. It is, however, important to underline that this work assume nothing specific about LTE, thus its design is general and equally suitable for future mobile RAN technologies. Fig. 1 provides a high-level scheme of the proposed architecture [16], composed by three main components/levels: FogRAN, Mobile Edge Host (ME Host) and Network Services located at the remote side. The FogRAN component includes multiple RAN data plane communication and flexible deployments [8] covering RAN function split in the context of CloudRAN. In the original vision of

CloudRAN, each remote radio head (RRH) is connected to the base band unit (BBU) pool with a point-to-point fronthaul (FH) link that may be daisy-chained whereas, in the evolved vision, the RRH becomes an active element that hosts a subset of additional baseband functions (e.g. OFDM (de-)modulation in LTE) and is renamed as remote radio unit (RRU). Furthermore, the BBU can be divided into distributed unit (DU) and centralized unit (CU) with a second, high level functional split in between as shown in Fig. 1.

The proposed hierarchical MEC is realized through FlexRAN software-defined RAN (SD-RAN) [9], which defines the southbound APIs and control protocol to facilitate communication between the RAN controller and the agents. As shown in the figure, it allows a two-way interaction between the controller and the agents. In one direction, an agent sends relevant messages to the master with eNodeB statistics, configurations and events, while in the other direction the master can issue appropriate control commands defining the operation of the agents. Unlike typical SDN controllers in the wired domain, the FlexRAN controller has been designed with support for time critical RAN operations (e.g., MAC scheduling) in mind. Due to this real-time aspect and in order to fully utilize the power of FlexRAN, ideally the communication channel between the agents and the master controller would be out-of-band, high-bandwidth and low-latency (e.g., optical fiber path). However it should be stressed that this is not a hard constraint, as the system provides enough flexibility to operate in non-ideal networking conditions with a small impact in its performance and capabilities [9]. Furthermore, upon the integration with FlexRAN, the ME Host coupling with the FogRAN component is able to really take advantage of real-time information and provide application environment at the network edge.

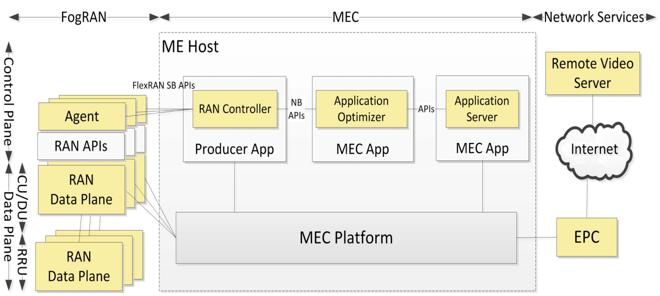

Fig. 1. Architecture of the MEC Platform implementation

In the above hierarchical architecture, eNB agents are located at the FogRAN level in order to gather real-time radio information from the underlying eNB, whereas the RAN controller, application optimizer and server are located at the MEC platform representing three different types of MEC app, each with different set of interfaces. For instance, the RAN controller is an eNB producer app and has interfaces to the eNB agent, MEC platform, and the optimizer.

III. THE RAVEN USE CASE

Our RAVEN prototype [10], realized and showcased as PoC in the framework of ETSI MEC activities, is a particular case of the more general hierarchical MEC architecture proposed in this paper. In fact, even if for practical needs of demonstrations the RAVEN PoC was implemented as a small scale prototype (consisting of one single eNB and a single eNB agent), the proof-of-concept can be easily extended to include multiple RAN nodes with their associated agents as explained in the previous section.

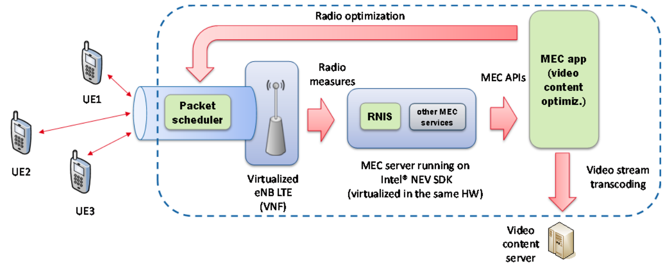

Fig. 2. Functional scheme of the RAVEN PoC.

The PoC demonstrates a RAN-aware video optimisation application, where MEC application is co-located with eNB and communicates with video server to dynamically adjust the quality of video streams based on the radio conditions of each user. As a result, the perceived user quality of video streams is significantly improved with the help of the MEC platform.

A. Implementation of the LTE virtualized environment

This small scale overlay LTE mobile network has been implemented in software, by using the OpenAirInterface (OAI) LTE platform [12] and a co-located MEC server virtualized in the same HW. In particular, the RNIS block depicted in the above figure (indicating the usage of MEC RNI API) can be considered as "RAN-domain service" aimed to improve the network and/or application-specific performance.

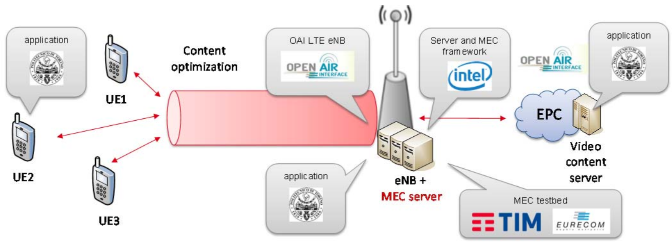

Fig. 3. Main components of the RAVEN proof-of-concept.

The above scheme depicts also the main entities of the RAVEN PoC, which has been demonstrated at the MEC Congress and MWC in 2017 [10][11].

B. Implementation of the RAVEN app

The RAVEN use case is related to a video streaming service shared among local or remote users, where a user (sender) produces a realtime video stream for other users (peer receivers) that are subscribed to this service. Based on Fig. 3, the steps are outlined below:

1. A UE records and sends a video stream to the video server in uplink (e.g. UE1);

2. The server receives and distributes the video stream to the registered UEs in downlink (UE2 and UE3 in Fig. 3);

3. The MEC platforms captures the congestion or channel quality event and send a trigger to the application optimiz-

er, so that it adjusts accordingly the quality of the video streams at the application server.

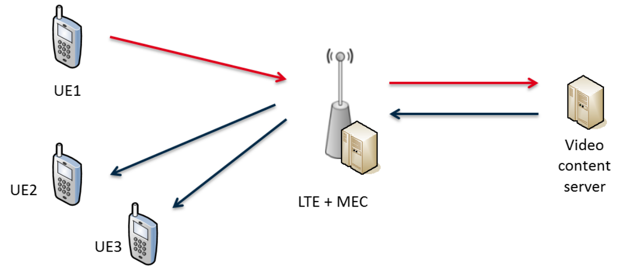

Fig. 4. Main entities involved in the RAVEN use case demonstration

Fig. 5 summarizes the main characteristics of the RAVEN Android application implemented and installed on commercial LTE smartphones for the demonstration purpose.

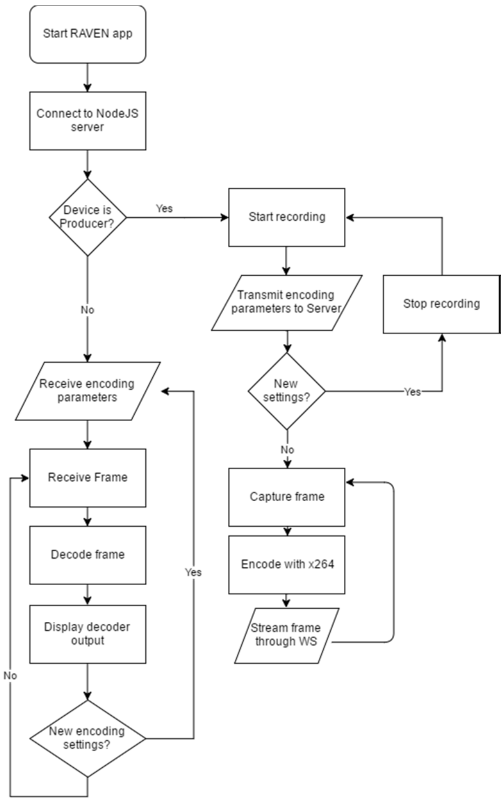

Fig. 5. Flowchart of the RAVEN application

The RAVEN app implemented for this demonstration targets smartphones running the Lollipop version of the Android [13] operating system. The application is responsible for capturing a live video stream from the device's camera and compressing it on the fly with the x264 [14] encoder library. The encoded video stream is then transmitted through a WebSocket connection to an ad-hoc webserver created with NodeJS [15]. The server acts as a relay between the different devices as it pushes the frames from the producer (UE1) to the consumers (UE2, UE3). The RAVEN app can change the encoding parameters, video resolution and compression rates, upon a request coming from the eNB agent.

The receiving device decodes the incoming video stream by using the Android MediaCodec framework. A consumer can be attached to the stream at any time since the RAVEN server sends the stream configuration parameters, the encoding details, in the first message. From that moment, the receiver app can decode the incoming frames of the video stream.

C. Implementation of the RNI API

The PoC presents an early version of Radio Network Information API, based on the ETSI MEC specification [18]. The main characteristics of this early RNI API implementation is the support of eNB status and configuration monitoring through a RESTFUL application interface, which can be triggered either as event-driven, periodic, and one-shot event.

An architectural overview of this RNI implementation is also shown in Fig. 6, where Mp1 interface is defined between the MEC app (eNB producer) and the RNIS block (constituting the actual MEC service). A rich set of information is produced at application level thanks to RNIS API: Radio Information base (RIB), different types of statistic report (event-driven, Periodic, One-off). Video Optimizer subscribes to a set of events at the eNB producer to further adapt video streaming rate. According to the different MEC deployment options considered in the PoC, the video optimizer can be either local or remote as the communication is performed through the REST APIs to provide the required flexibility and interoperability with the eNB producer app.

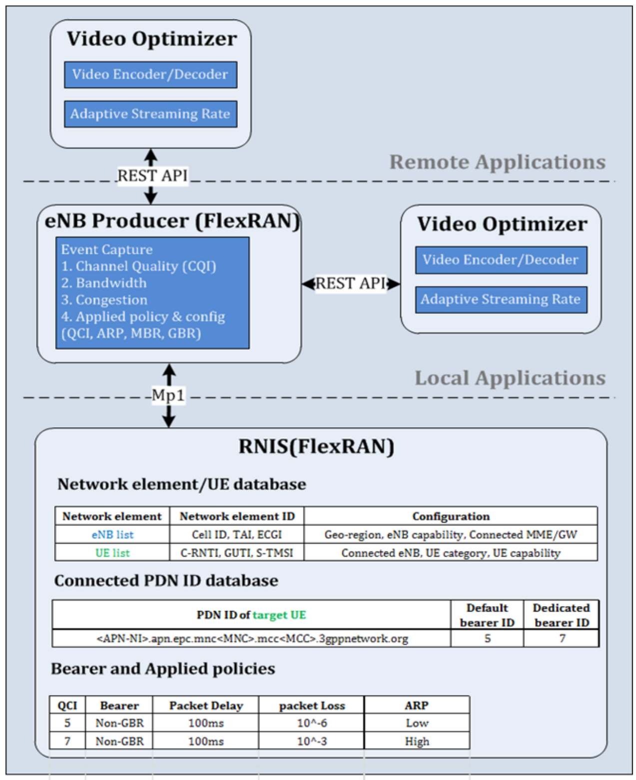

Fig. 6. Architectural view of the RAVEN implemention of RNI API

IV. Performance Results

In order to assess the benefits of our system, we made some preliminary tests based on the LowLatency MEC platform (LL-MEC) [19], by measuring the mapping between CQI index and bi-directional TCP bitrate identified during experiments (refer to Table 1 below). These results are also indication of the performance of our RNI implementation, that is the sustainable capacity for application in terms of throughput, in relationship to radio network information (e.g. CQI index) coming from the network: CQI index is definitely one of the meaningful radio parameters. In addition, any potential parameter can be easily considered in video streaming optimization and QoE. Various possible statistical method can be also applied to the raw data, thanks to the well-defined abstraction layer.

The measured results showed in Table 1 reveals the maximum sustainable bitrate as a function of user channel quality indicator (CQI), which is combined with the cell congestion level to adapt the streaming quality.

Table 1 Measured maximum sustainable TCP bitrate with discrete congestion level based on CQI

CQI	Congestion Level	Downlink (Mb/s)	Uplink (Mb/s)
11- 15	Low	15.224	8.08
9- 11	Low	11.469	6.04
7- 9	Medium	9.88	4.47
4- 7	Medium	5.591	2.49
0-4	High	1.08	0.69

Moreover, by exploiting the flexibility of the Video Optimizer implementation, we assessed our PoC by comparing two MEC deployment options and their related performances, e.g. in terms of RTT (Round Trip Time): an ideal scenario where MEC is used locally, compared with the case of a remote server. In particular, in order to characterize the RTT in terms of different placements of the content, we conducted experiments on the RTT through different traffic patterns generated by ping utility, namely 64, 768, 2048, 4096, 8192 packet sizes in bytes, and 1, 0.8, 0.4, 0.2 inter-departure time (IDT). Four different combinations of content placements have been considered: the first three cases was measured between UE and remote servers (located in eNodeB, Italy, and USA). The rest case considered a remote server in Italy with MEC deployed at PDN. The measured RTT in box-plot of these four placements are showed in Fig. 7. A commercial LTE UE terminal (Huawei E392 USB dongle) located 10 meters from an operational OAI LTE eNodeB (5MHz bandwidth, FDD, SISO) was used.

We can observe that the influence of MEC server placement is proven to be significant in this preliminary results and various factors play important roles when investigating content caching based on MEC. In particular:

- When considering the difference between the top left picture (local MEC) and the top right (remote server in Italy) we can notice that the delay loss is consistent although limited. This case can be consider as a sort of upper bound for remote connections (even if of course the local MEC is always performing best).
- On the other extreme, when moving from the top left picture (local MEC) and the bottom left (remote server in USA) we can notice that the delay loss is huge and highly variable (in all configurations). This case can be consider as a sort of worst case for remote connections (or, better, typical case of remote data centers, that generally are located very far from the edge of the network).
- Finally, in the last case of Fig. 7 it can be observed that the latency has little or no difference in relation to size of sent packets. In other words, this result suggests that the benefits of MEC deployed before network reaching the edge is more related to flexibility and programmability and this is the place where SDN is involved.
- All in all cases, the experiment conducted in a real-time LTE environment co-located with MEC server shows not only the benefits of MEC concept but also the feasibility of deploying at different locations, exploiting different levels of eNB agents. Furthermore, to the best of our knowledge, the study of MEC server deployed in a real environment and the comparison of different locations of MEC server has not been addressed in the literature. Thanks to the OAI, FlexRAN, and LL-MEC opensource platforms, this experiment gives valuable inputs for industry as well as research community and potentially higher the market acceptance.

I. Conclusions

In this paper, we proposed a hierarchical MEC architecture and report a proof-of-concept (PoC) implementation of the RAVEN use case. The PoC has been assembled in a small-scale open-source LTE environment with commercial terminals with the objective of demonstrating a real-time adaptive video streaming empowered by the MEC platform at the edge of the network.

In the considered scenario, real-time radio information is provided by eNB agents, which act as local controllers co-located with eNB and collecting requested RAN data. Agents may cooperate with each other to control the network in a distributed manner or may be delegated and/or controlled by a master entity. Realtime radio information gathered by eNB agents is thus provided to the MEC system through the proposed hierarchical architecture.

The results of the experimental activity demonstrate the benefit of radio information in optimizing the video streaming in a cell: video streams and the quality perceived by users are improved with the help of RNI service following the standardization efforts within the ETSI MEC working group [18].

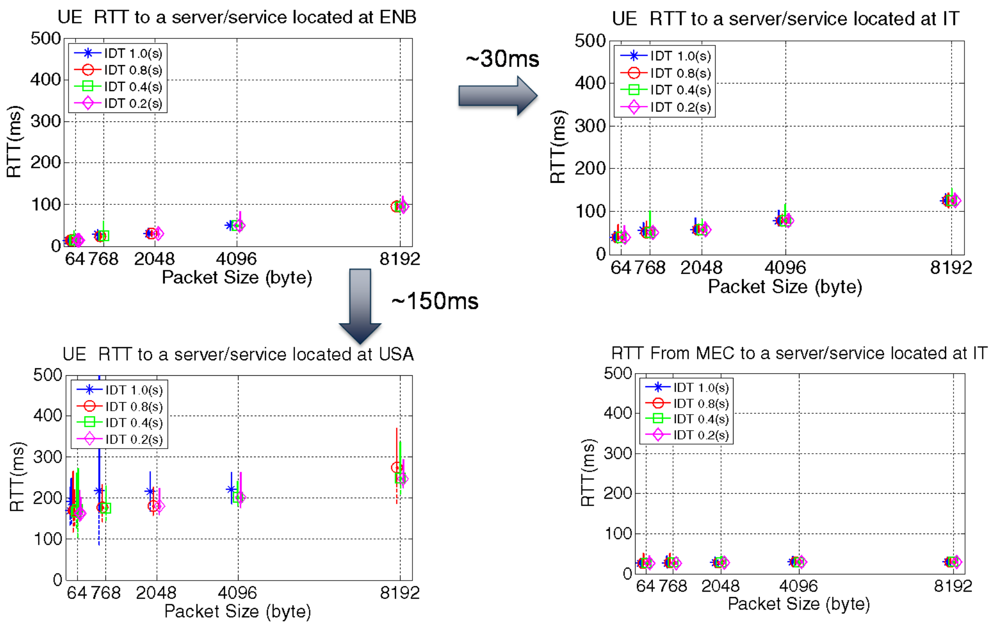

Fig. 7. RAVEN PoC: Measured packets RTT

The work reported in this paper can be extended in several directions. The first step will be the analysis of possible scenarios of MEC based architecture by implementing new PoCs: this is necessary, in fact, in order to foster the industry to prototype solutions and, in the meantime, to spread architectural concepts, to study new services and to exploit the new unlocked features. The next step, necessary to gain a wider market acceptance of the proposed MEC architecture, will be the definition of possible architectural updates in ETSI MEC, in addition to further RNI API enhancements, as follow-up of the lessons learned from the realization of the RAVEN PoC.

Acknowledgment

The Work is a result of collaboration between Intel, Eurecom, Politecnico di Torino and TIM, in the context of ETSI MEC PoC, named RAVEN ("Radio aware video optimization in a fully virtualized network")[10], and partially supported by the H2020 SliceNet project under grant agreement number 761913.

The authors are grateful to Luigi Renda, Krishna Murthy, Marco Caretti, Michela Meo and Joe Cahill for their great support and kind availability during all the phases of this joint project.

References

[1] Cisco White Paper "Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2016–2021", Feb. 2017, Available: www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/mobile-white-paper-c11-520862.pdf

[2] ETSI GS MEC 002 V1.1.1 (2016-03): "Mobile Edge Computing (MEC); Technical Requirements"

[3] ETSI GS MEC 003 V1.1.1 (2016-03): "Mobile Edge Computing (MEC); Framework and Reference Architecture"

[4] ETSI white paper: "Mobile Edge Computing: A key technology towards 5G", September 2015; www.etsi.org/images/files/ETSIWhitePapers/etsi_wp11_mec_a_key_technology_towards_5g.pdf

[5] 3GPP TS 23.501 V0.4.0 (2017-04), System Architecture for the 5G System; Stage 2, (Release 15), April 2017

[6] 5G Infrastructure Association: Vision White Paper, February 2015. http://5g-ppp.eu/wp-content/uploads/2015/02/5G-Vision-Brochure-v1.pdf

[7] T. Taleb, *et al.*, "On Multi-Access Edge Computing: A Survey of the Emerging 5G Network Edge Cloud Architecture & Orchestration", IEEE Communications Surveys and Tutorials Journal, 2017

[8] D.Sabella *et al* "A Flexible and Reconfigurable 5G Networking Architecture Based on Context and Content Information", EUCNC2017

[9] X. Foukas, et al., FlexRAN: A Flexible and Programmable Platform for Software-Defined Radio Access Networks, CONEXT 2016, 12th International on Conference on emerging Networking EXperiments and Technologies, Irvine, California, USA.

[10] PoC RAVEN: Radio aware video optimization in a fully virtualized network, Link: http://mecwiki.etsi.org/index.php?title=Ongoing_PoCs

[11] HPE & Intel Webinar: 5G-Ready Telco Edge featuring TIM, Eurecom, Politecnico di Torino, & Wind River. April 2017. Link: https://www.sdxcentral.com/hpe/csp/partner-ecosystem/hpe-intel-5g-ready-telco-edge-webinar/

[12] OpenAirInterface website: http://www.openairinterface.org/

[13] Android reference : https://www.android.com/

[14] VideoLAN x264 Encoder refernce website http://www.videolan.org/developers/x264.html

[15] NodeJS reference website : https://nodejs.org/en/

[16] C. Chang, *et al.*, "MEC architectural for LTE/LTE-A networks", IEEE International Conference on Communications 2017

[17] opensignal.com/reports/2017/05/italy/state-of-the-mobile-network

[18] ETSI GS MEC 012 V1.1.1 (2017-07): "Mobile Edge Computing (MEC); Radio Network Information API";

[19] A. Huang, et al., "Low latency MEC framework for SDN-based LTE/LTE-A networks, IEEE ICC 2016.