Design and Evaluation on Mobile Edge Caching Testbed

Can Zhang, Mingyuan Zang, and Ying Yan

***Abstract*—Internet traffic is predicted to increase fast over the next years. A large portion of it will be generated by mobile video services. Such a data explosion puts higher requirements on the capacity of the mobile network. Deploying more bandwidth resources to increase the network capacity is one solution, but it also means high cost. Mobile Edge Caching (MEC) is a new solution put forward these years to deal with the drastic growth of video data by bringing the video resources close to users at the edge cache. Researches have been done on the design of MEC, and implementing it in an emulator is one of the ways to verify the design. An emulator can provide real-case protocol implementation and more credible results compared with simulator. This paper studies the LTE emulators available in the market and proposes an MEC testbed based on the OpenAirInterface platform. Two use cases of the testbed are demonstrated and their performances are evaluated separately.**

***Index Terms*—Testbed, mobile edge caching, LTE emulation, open air interface.**

I. Introduction

In the past few years, Mobile Edge Caching (MEC) technology has become a promising technology and received increasing attention from both industry and academia. The MEC is a promising solution to the case, where increasing volume of video content are streamed and exchanged with the stringent latency requirement. Compared with deploying extra resource in the backhaul network, caching in the radio access network (RAN) is effective and economical. Specifically, MEC shifts the burden of data transmission from the backhaul network to the fronthaul network in order to minimize bandwidth requirement and improve latency.

To evaluate different caching scenarios, various testing methods are adopted. Large-scale lab and field tests can be conducted by mobile network operators, which require resources on testing equipment and are usually time consuming. The most commonly adopted method for academic researchers is simulation. Network simulation tools are software-based programs and use statistical/numerical models for traffic generation and performance analysis. Popular network simulation tools are available such as Riverbed Modeler (previously known as OPNET), NS3, Omnet++, NetSim and so on. Network protocols and scenarios are implemented in a pure software environment. Testing goals are reached based on assumptions and the adjustment of simulation parameters. However, the simulation environments are not close enough to the real testing situations. Therefore, emulation tools show a better way to approach the real wireless communication system. Compared with the simulation method, the emulation system contains both hardware equipment and software protocol stacks. The emulation platform implements a full stack mobile communication system and involves a software defined radio (SDR) based radio communication environment. The LTE network emulators allow users to utilize lab equipment effectively and to test different network topologies with reduced deployment cost. Nowadays, several platforms are available to offer the emulate LTE network and to examine the transmission-level features. Common emulation platforms include OpenAirInterface (OAI), Amarisoft LTE, srsLTE, OpenEPC and OpenLTE. Details of these platforms are introduced in section II.

In this paper, an OAI-based LTE emulation testbed is set up. The testbed offers a software platform, developed based on the standardized LTE protocols. The proposed in-lab emulation platform can produce testing results closer to the real application while achieving repeatability and portability. This testbed presents an end-to-end communication system with access from multiple users. MEC functions have been implementation and tested with a steady performance.

Our contributions are summarized as follows:

- First, we conduct a survey of existing network emulators and related researches in the literature.
- Second, MEC testbed is prototyped and its design process is introduced in details with focusing on the setup procedure and modelling principles.
- Third, we present two basic tests to verify the proposed advantages of such an in-lab MEC testbed.

The remainder of this paper is organized as follows. Section II presents a survey of existing LTE network emulators. Section III provides a thorough description of testbed configurations and MEC design including the system implementation on control plane and user plane. Section IV discusses MEC performance under several testing scenarios. Section V summarizes our work and proposes the future work.

II. Survey of LTE Network Emulator

This section provides a preliminary survey on existing LTE network emulators and introduces their key. Moreover, related applications and deployment scenarios are presented based on literature study.

OpenAirInterface (OAI): OAI is an open-source emulation platform provided by OpenAirInterface Software Alliance of EURECOM. It is one of the most comprehensive and competitive open-source LTE network emulators based on Software Defined Radio (SDR) system. Its current implementation provides a complete wireless protocol stack as well as an IPv4/IPv6 network device interface under Linux system. All elements of the 4G LTE system such as user equipment (UE), evolved NodeB (eNodeB), and the Evolved Packet Core (EPC) are provided. The software program is

Manuscript received September 28, 2019; revised February 15, 2020.

Can Zhang, Mingyuan Zang, and Ying Yan are with Department of Photonics Engineering, Technical University of Denmark, 2800 Kgs. Lyngby, Denmark (e-mail: canzh@dtu.dk, minza@dtu.dk, yiya@fotonik.dtu.dk).

DOI: 10.7763/IJMO.2020.V10.741

installed and run on the host computer with an SDR card for wireless signal processing. Regarding MEC in LTE system, some researchers have built a single user access network on OAI platform to investigate from different aspects. An MEC architecture is proposed in [1] and a prototype for data caching use case is conducted. Authors in paper [2] deploy OAI platform to implement a collaborative caching system based on service function chaining method.

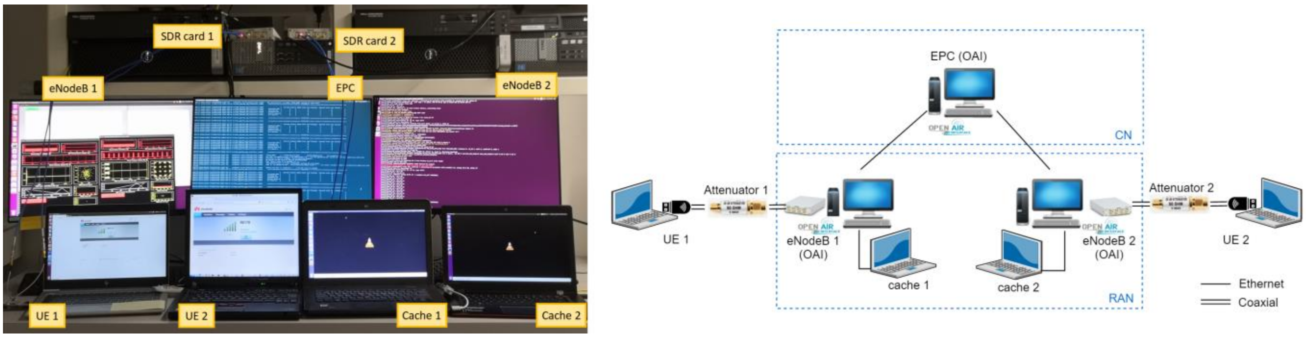

Fig. 1. In-lab MEC network platform setup and system architecture.

Amarisoft LTE: Amarisoft LTE is a commercial software product for LTE network emulation [3]. Similar to OAI in terms of complete implementation and emulation capabilities, Amarisoft LTE supports full stack LTE protocol in software. In [4] the authors conduct a detailed security analysis on different mobile networks including Global System for Mobiles (GSM), General Packet Radio Service (GPRS) and System Architecture Evolution/Long Term Evolution (SAE/LTE). An end-to-end mobile communication security testbed using open source components on a virtualized platform is set up. To implement the SAE/LTE network, LTE Amarisoft 100 is used as the EPC in the testbed. Authors in [5] propose a mobile relay architecture for public transport system. They evaluate the architecture with Amarisoft software suite.

srsLTE: srsLTE is a free and open-source software developed by Software Radio Systems (SRS) company [6]. It contains a complete SDR LTE eNodeB application (srsENB), a UE application featuring all layers (srsUE), a light-weight LTE core network implementation with MME, HSS and S/P-GW (srsEPC). The comparison between two open-source tools, OAI and srsLTE, has been conducted in research. In [7], testbeds using OAI and srsLTE are set up respectively, comprising one eNodeB and one UE. The authors measure the downlink throughput under different bandwidth configurations. CPU execution time and memory usage for key modules in OAI and srsLTE are tested and compared. The paper [8] reports performance assessments of srsLTE and OAI. The differences in throughput and resource consumption are investigated and quantified. Based on the results, the authors provide guidelines for deploying 5G testbeds with these tools. A testbed using the srsLTE library is designed in [9]. Investigations focus on the performance assessment of enhanced smart meter traffic in the grid over LTE network.

OpenEPC: OpenEPC project has been developed by Fraunhofer Institute FOKUS since 2008. It covers all functional elements in 3GPP EPC specifications and offers advanced IP mobility and QoS control. In [10], a flexible and extensible Diameter based testing system for next generation mobile network is designed. A novel architecture for mobile video surveillance system over EPC is proposed in [11], which allows the delivery of videos streams with different QoS and video quality profiles. [12] demonstrates a distributed network architecture using a custom off the shelf LTE small cell and OpenEPC. In [13], OpenEPC is added to an existing LTE based network as virtual core network. The Nokia small cells are deployed for the LTE radio. OpenEPC as a virtual LTE core network could be deployed as Network Function Virtualization (NFV) based EPC. However, this platform only focuses on the implementation of EPC. For a complete LTE system, eNodeB and UE functions are required from other partners.

OpenLTE: OpenLTE project is also an open source implementation of 3GPP LTE specifications and focuses on transmission and reception of the downlink. OpenLTE is limited with lack of a UE software and many features are still under development. [14] presents a data offloading solution using a coexistent LTE and WiFi testbed based on OpenLTE. No performance has been conducted from this paper since the OpenLTE package is not stable enough.

To a summary, various options are available with pros and cons. Due to the high compliance to 3GPP LTE standards, the support for all system elements, the reputation for stability and sufficient technical guidance from OAI community, we have chosen the OAI platform in this work.

III. System Implementation

A. *Testbed Layout and Setup*

The experimental environment of our in-lab MEC network platform is shown in Fig. 1. The testbed is set up according to the recommendations provided by OAI community.

For the software configuration, OAI software stacks are installed in eNodeB as openairinterface5g (v0.6.1) [15] and EPC as openair-cn (v0.4.0) [16]. Detailed configurations are listed as follows:

1) *eNodeB configurations* Configuration file *enb.band7.tm1.usrpb210.conf* defines all the parameters for eNodeB. Identification parameters and network interfaces in the configuration file are modified for the testbed.

2) *EPC configurations* The EPC implementation in OAI includes the functions of MME, HSS, SGW and PGW. MME performs mobility management and authentication; HSS offers a database with UEs' information; SGW handles data forwarding and switching on the user plane and PGW assigns IP address to UEs and serves as

gateway to the Internet. Configuration files *hss.conf, mme.conf* and *spgw.conf* contain parameters for all the above-mentioned entities, such as network interfaces, free diameter settings. To enable multiple eNodeBs to associate with the same EPC, the maximum number of eNodeBs in *mme.conf* needs to be defined.

3) *UE configurations* HUAWEI LTE dongle is used for the wireless connection of UE. A programmed SIM card is inserted so *that* the PC can get access to the LTE network with the dongle.

The UE and the SDR board are connected via a RF cable link with an attenuator to minimize the radiation from the neighbor eNodeB. The HUAWEI E3372 dongle has a pair of micro co-ax connector, which is connected to the SDR card with a pair of CRC9 to SMA cable. A 20dB attenuator is connected inline in each cable. Thus, it can provide a stable signal level adjustment in the LTE band 7 used in eNodeB. With such a setup and configuration, two UEs can establish a stable connection with two eNodeBs respectively.

B. Implementation Overview

The MEC platform is based on a full-stack LTE network consisting of four entities, namely EPC, eNodeB, UE and cache module.

In the MEC system, eNodeB administrates the flows of UE requests and directs the requests either to the remote server via EPC or directly to local cache. Communication and signaling procedures among 4 entities are illustrated in Fig. 2.

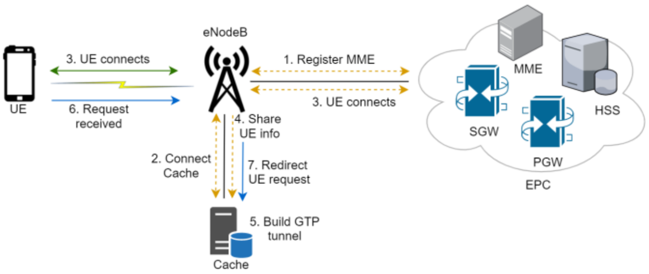

Fig. 2. System design of the testbed.

The key steps of realizing MEC functions are summarized as follows.

- *Connection establishment:* eNodeB establish the delivery of control/signaling messages with the MME and the local cache module (step 1 and 2).
- *UE attachment and registration*: when a UE attempts to attach, eNodeB exchanges the UE's identity information between UE and EPC, meanwhile eNodeB shares the same information with local cache module. (step 3 and 4)
- *Data exchanging establishment*: user data packets can be obtained to UE from either the remote server or the local cache through eNodeB. (step 5)
- *Request forwarding*: eNodeB makes a decision based on the received UE's request and forwards it to either the cache or the remote server. (step 6 and 7)

As defined in LTE network, the communication interface between eNodeB and EPC is split into control plane and user plane. Similarly, the communication between eNodeB and the local cache involves both signaling and data exchange. In this work, modifications and new functions have been implemented in the OAI protocol stack. An overview of relevant functions is listed and described in Table I. Detailed implementation on control plane and data plane are provided in the remainder of this section.

C. Control Plane Setup

4) *SCTP setup in eNodeB:* In LTE, the communication between eNodeB and EPC is through the *S1* interface. The signaling transport between eNodeB and MME on *S1* control plan is based on Stream Control Transmission Protocol (SCTP). In the OAI platform, a SCTP association is established between eNodeB and MME when starting the eNodeB. The *S1* Application Protocol (S1AP) handles all the signaling in this process. A new struct called *s1ap_eNodeB_instance_t* is firstly created, carrying all the information about eNodeB and the network information where the eNodeB is located. By calling the function *s1ap_eNB_register_mme*, eNodeB attempts to establish the connection with MME. This is initialized by assigning a set of parameters including the local and remote IP address, Application default port number for S1AP, SCTP Protocol Payload Identifier (PPID) for S1AP, and the number of inbound and outbound streams.

The SCTP association with the cache follows a similar way as described above. A new SCTP association request is prepared in which the remote address is assigned to the cache's address instead of the SGW's address. The corresponding association ID is increased by one compared to the original association ID to MME.

1) *SCTP setup in cache:* In the cache entity, a SCTP module in the Linux kernel is loaded at first. Then a SCTP server is set up, which listens to the specified port and receives SCTP packets from eNodeB.

2) *GTP tunnel setup preparation:* When a UE attempts to attach the network, eNodeB sends an initial UE message to initialize the process. After several message exchange in EPC, MME *responds* with a S1AP Initial Context Setup Request message, containing the assigned UE IP address, the uplink TEID and the UE S1AP ID provided by EPC. These identifiers are stored in a predefined globally accessible struct and processed in the function *s1ap_eNB_handle_initial_context_request* in eNodeB. To respond to the request from MME, eNodeB sends an S1AP Initial Context Setup Response message from the function *s1ap_eNB_initial_ctxt_resp*. A downlink TEID is generated and included in this response message. The obtained UE IP, uplink and downlink TEIDs are later used to build the GTP tunnel in the cache. In order to forward these identifiers to cache via the control plane, a SCTP message is generated and sent to cache. The message is constructed by 4 fields with a total length of 16 bytes, namely message type, TEID Downlink, TEID Uplink and UE IP address.

D. User Plane Setup

1) *Enabling GTP kernel in cach*e: the role of S1 user plan is to transport user data packets between eNodeB and the SGW. The interface is built by using GPRS Tunneling Protocol (GTP) tunnels. To implement the MEC function, a new GTP device is added with the support of libgtpnl tool. When a SCTP message is received from eNodeB, the cache entity parses the SCTP message and obtains UE IP address, downlink TEID and uplink TEID. The GTP

tunnel connected to this device is then configured with these identifiers. The UE IP address is added into the routing table of GTP device.

The cache entity can now communicate with eNodeB on user plane. It is able to handle the request encapsulated in GTP packet, access to the resource stored in cached server and respond back via GTP tunnel. A virtual interface is built in cache by setting up Linux namespaces to bind the cached server offering resource.

TABLE I: Overview of S1AP and GTPV1-U Functions in OAI

Source files	Function name	Function Descriptions
S1ap_eNB.c	*s1ap_eNB_register_mme*	eNodeB registers MME by setting up the SCTP association
	s1ap_eNB_handle_ sctp_association_resp	eNodeB handles the SCTP association response from MME
S1ap_eNB_handlers.c	*s1ap_eNB_handle_ initial_context_request*	Process the bearer parameters to set up UE context
S1ap_eNB_nas_procedures.c	*s1ap_eNB_initial_ctxt_resp*	Respond to MME to indicate the successful setup of E-RAB
gtpv1u_eNB.c	*gtpv1u_eNB_task*	Send and receive GTP packets

2) Caching scheme implementation: Having the SCTP and GTP-U tunnels paved for the control plane and user plane respectively, the cache is prepared and ready to offer its local resources to users. In the GTPv1-U codes of OAI, the function *gtpv1u_eNB_task* runs a loop to handle the GTP tasks received over ITTI interface. The handling process is illustrated by a flowchart in Fig. 3.

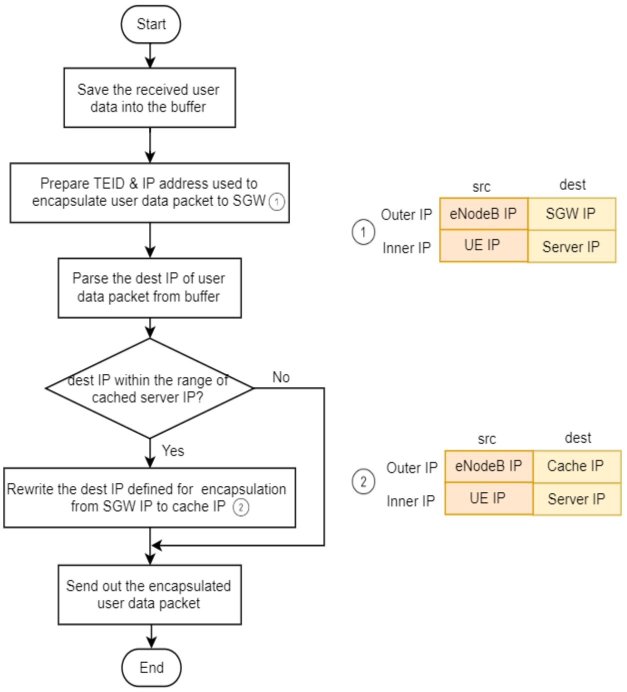
Start
Save the received user data into the buffer
Prepare TEID & IP address used to encapsulate user data packet to SGW ①
Parse the dest IP of user data packet from buffer
dest IP within the range of cached server IP?
No
Yes
Rewrite the dest IP defined for encapsulation from SGW IP to cache IP ②
Send out the encapsulated user data packet
End
①
src
dest
Outer IP
eNodeB IP
SGW IP
Inner IP
UE IP
Server IP
②
src
dest
Outer IP
eNodeB IP
Cache IP
Inner IP
UE IP
Server IP

Fig. 3. Flowchart of caching scheme implementation.

The user data packet is first stored to a buffer. Afterwards, information to encapsulate the packet into a GTP packet is generated. By default, the outer destination IP address is only assigned to the SGW address. To achieve the caching scheme, we implement a decision process with a predefined cashing server address list. If the destination address matches one in the predefined cached server address list, the outer destination IP address is rewritten to the address of local cache. Otherwise, the packet is directly forwarded to the SGW with the address unchanged.

IV. Testing and Performance Evaluation

A. Testing Scenarios

The proposed in-lab MEC testbed can be used to evaluate network performances. With the usage of OAI platform, the mobile transmission features are taken into account in the end-to-end communication system. In this section, we show two basic test cases to investigate the network performance.

Case A: Coding rate The coding rate can affect the size of a video and also the quality perceived by users. High coding rate can offer better visual experience. However, the large size of the video will also burden the link. In this case, we focus on whether introducing the cache into the network can improve the video transmission under different coding rates. The test scenarios include transmission of a video with coding rate of 100 kbps and 500 kbps from either the remote server or the local cache. We inspected on the frame loss rate and delay to evaluate the quality of video perceived by users.

Case B: multiple UEs access The number of UEs access to video server has an impact on the quality of video service. To investigate if adding edge cache could offload the traffic, we analyze the test case in which two UEs attached with two eNodeBs respectively and two eNodeBs connect to the same EPC (as shown in Figure 1). There are three scenarios designed to compare the results:

1) Scenario 1: only remote server available.
2) Scenario 2: local cache 1 is available for UE 1.
3) Scenario 3: both local cache 1 and 2 are available.

The video clip offered in this case has 2400 frames and has a coding rate of 1 Mbps. It is streamed over the emulated LTE network using Real Time Streaming Protocol (RTSP). Since the testbed is built in lab, there is little geographical distance in the backhaul network. Thus, 1% jitters and 10ms delay are added at EPC network interface on purpose to simulate the real-world network transmission.

B. Results Analysis

Case A Coding rate: Fig. 4 (a) and (b) show the results of coding rate tests. In general, caching the video clip locally can improve the Quality of Experience by reducing the jitters and delay. Considering the coding rates, the larger coding rate results in a better improvement from using the local cache. When the coding rate is 100 kbps, introducing the cache helps to reduce the delay by 0.8%. While for the video clip with 500 kbps, the cache successfully reduces the delay by 1.5%. The results of jitter variants demonstrate also an improvement of adopting the local cache in the system. When the video clip has 100 kpbs coding rate, around 2.3% jitters happens during the transmission from the remote server. This number is cut down to 1% when the cache offers the service. In the case of the video with 500 kbps rate, jitter is reduced from 2.7% to 1%.

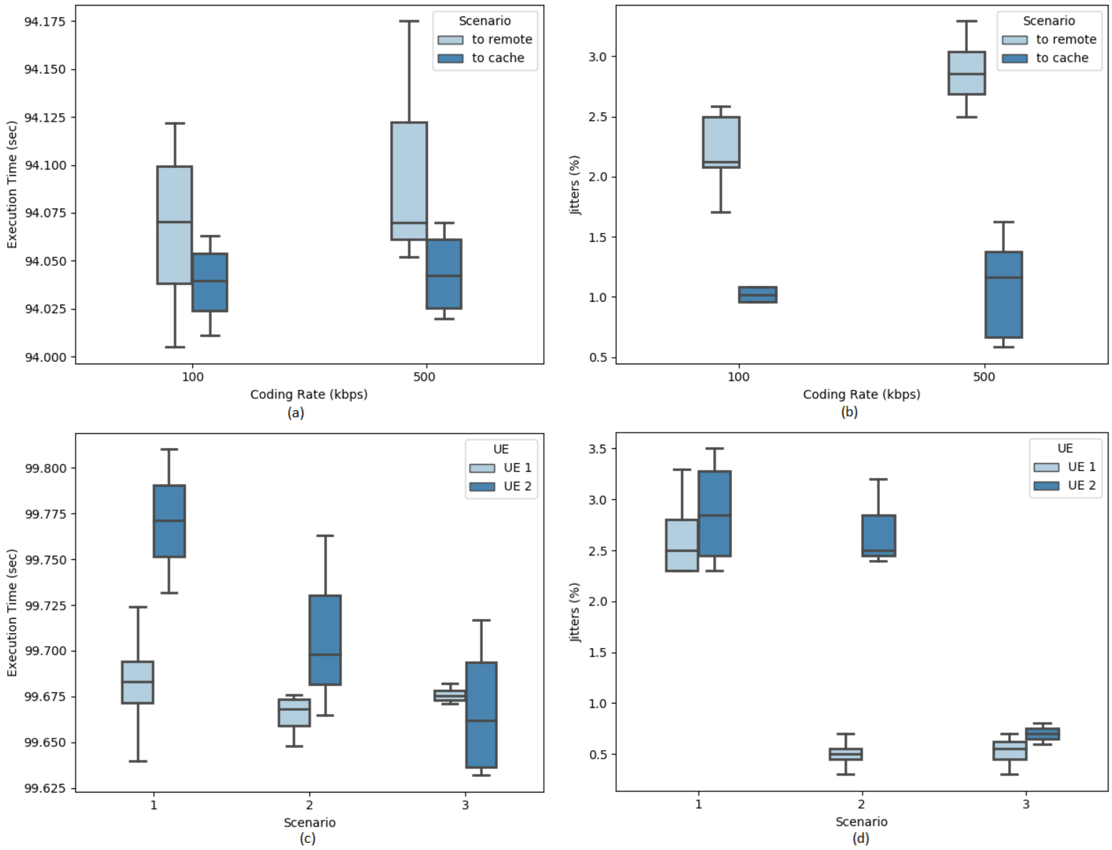

Fig. 4. Testing results (a) delay in case A; (b) jitters in case A; (c) delay in case B and (d) jitters in case B.

Case B Connected UEs: Fig. 4 (c) illustrates the comparison of delay under three scenarios. When both UEs are served directly by cache, the execution time presents a decreasing trend. While this trend is not obvious for UE 1, UE 2 receives a better service with the introduction of the cache. When both UEs are served by the cache, UE 2 plays the video with a reduced delay to 2%. Even when only the UE 1 is served by cache and UE 2 is still kept served by remote server, the delay can be reduced by about 1%. This is also the case for jitters. Fig. 4 (d) demonstrates the comparison of jitters. When UE 1s request is redirected to the cache, the traffic on the backhaul link is offloaded and UE 2 gains a batter service with jitters reduced from about 3% to 2.5%. When both UEs are served by the cache, UE 2 receives a decreasing number of loss frames, which is reduced to 0.5% compared with 3%.

V. Conclusion

This paper has provided a flexible testbed that is capable of determining performance measurements on LTE-based MEC technology. This work also gives a detailed description of its design structure and test configurations. This testbed aims to discover MEC capabilities with multiple connected users under three different test scenarios. However, the configuration possibilities are not limited to our conducted experiments. Taking into account the functional design and setup that has been verified, the proposed in-lab testbed can be used by researchers to further develop advanced MEC algorithms and to integrate with other innovative technologies, such as Software Defined Network (SDN).

Some notable observations can be made based on the test results. Firstly, network performances in terms of delay and jitter are improved by offloading the traffic from the backhaul network to the caches at eNodeB. When users request video streaming service from the local cache, they tend to experience a shorter delay, compared with that from the remote server. Furthermore, it is noticed that the improvement is more distinctive when higher bit rate video is used. Secondly, in the multiple users test scenario, the caching scheme can reduce the burden of the network bandwidth and meantime improve users experience by reducing the loss rate.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

Can Zhang, Mingyuan Zang and Ying Yan conducted the research; Can Zhang and Mingyuan analyzed the data; Can Zhang, Mingyuan and Ying Yan wrote the paper; all authors had approved the final version.

ACKNOWLEDGMENT

We would like to thank Justas Poderys, Matteo Artuso, Claus Michael Oest Lensbøl, Henrik Lehrmann Christiansen and José Soler for their help on OpenAirInterface platform.

REFERENCES

[1] C. Chang, K. Alexandris, N. Nikaein, K. Katsalis, and T. Spyropoulos, "MEC architectural implications for LTE/LTE-A networks," in *Proc. the Workshop on Mobility in the Evolving Internet Architecture*, New York, NY, USA, 2016.

[2] L. Lei, X. Xiong, L. Hou, and K. Zheng, "Collaborative edge caching through service function chaining: Architecture and challenges," *IEEE Wireless Communications*, vol. 25, no. 3, 2018, pp. 94-102.

[3] Amarisoft LTE. [Online]. Available: www.amarisoft.com

[4] K. Jijo George, A. Sivabalan, T. Prabhu, and Anand R. Prasad, "End-to-end mobile communication security testbed using open source applications in virtual environment," *Journal of ICT Standardization*, 2015.

[5] T. Kerdoncuff, T. Galezowski, and X. Lagrange, "Mobile relay for LTE: Proof of concept and performance measurements," in *Proc .2018 IEEE 87th Vehicular Technology Conference (VTC Spring*), Porto, 2018, pp. 1-5.

[6] SrsLTE. [Online]. Available: http://www.softwareradiosystems.com/

[7] Z. Geng, X. Wei, H. Liu, R. Xu, and K. Zheng, "Performance analysis and comparison of GPP-based SDR systems," in *Proc. 2017 7th IEEE International Symposium on Microwave, Antenna, Propagation, and EMC Technologies (MAPE)*, Xi'an, 2017, pp. 124-129.

[8] F. Gringoli, P. Patras, C. Donato, P. Serrano, and Y. Grunenberger, "Performance assessment of open software platforms for 5G prototyping," *IEEE Wireless Communications*, vol. 25, no. 5, pp. 10-15, 2018.

[9] A. Hematian, W. Yu, D. Griffith, and N. Golmie, "Performance assessment of smart meter traffic over LTE network using SDR testbed," in *Proc. 2019 International Conference on Computing, Networking and Communications (ICNC)*, Honolulu, HI, USA, 2019, pp. 408-412.

[10] T. Q. Thanh, D. Vingarzan, Y. Rebahi, and T. Magedanz, "A Diameter based testing system in Next Generation Mobile Network," in *Proc. 2010 14th International Telecommunications Network Strategy and Planning Symposium (NETWORKS)*, Warsaw, 2010, pp. 1-5.

[11] M. Abu-Lebdeh, F. Belqasmi, and R. Glitho, "A 3GPP 4G evolved packet core-based system architecture for QoS-enabled mobile video surveillance applications," in *Proc. 2012 Third International Conference on The Network of the Future (NOF)*, Gammarth, 2012, pp. 1-6.

[12] B. Nguyen, T. Zhang, B. Radunovic, R. Stutsman, T. Karagiannis, J. Kocur, and J. Merwe, "ECHO: A reliable distributed cellular core network for hyper-scale public clouds," in *Proc. the 24th Annual International Conference on Mobile Computing and Networking (MobiCom '18)*, ACM, New York, NY, USA, 2018, pp. 163-178.

[13] M. Arif, "OpenEPC Integration within 5GTN as an NFV proof of concept," 2017.

[14] D. Purohith and K. Sivalingam, "SIFM: A network architecture for seamless flow mobility between LTE and WiFi networks - Analysis and Testbed Implementation," 2017.

[15] openairinterface5g v0.6.1. [Online]. Available: https://gitlab.eurecom.fr/oai/openairinterface5g/tree/v0.6.1

[16] openair-cn v0.4.0. [Online]. Available: https://gitlab.eurecom.fr/oai/openair-cn/tree/v0.4.0

Can Zhang received his B.S. degree in communication engineering from Yangzhou University in 2018. He is currently pursuing his M.S. degree in telecommunication at Technical University of Denmark. His research interests focused on emulation platform in next-generation mobile communication network.

Mingyuan Zang received B.S. degree in electronic engineering from the Beijing University of Technology in 2018. She is currently pursuing the M.S. degree in telecommunication at Technical University of Denmark. Her research is focused on emulation platform in next-generation mobile communication network.

Ying Yan received the B.Eng. degree in electrical engineering from the Beijing University of Technology, China, in 2002. She has received her M.S. degree in electronics engineering in 2004, and the Ph.D. degree in telecommunication engineering in 2010 from Technical University of Denmark. During 2006-2007, she worked as a research scientist at the Department of Com-munication Platforms in the Technical Research Centre of Finland (VTT), Finland. Her research fields include internet of things (IoT) network infrastructure, 5G Mobile IoT network, time sensitive network (TSN), network simulation and emulation, traffic data mining and analysis, and network security.