An Investigation of Application-aware Mobile Edge Computing in 5G Networks

Maxim Domnitskyi
EECS International Graduate Program,
National Chiao Tung University,
Hsinchu, Taiwan
E-mail: maxim.domnitskyi@gmail.com

Chia-Nan Lin
Information and Communication Research Laboratories,
Industrial Technology Research Institute,
Hsinchu, Taiwan
E-mail: chiananlin@itri.org.tw

Abstract— **Mobile Edge Computing (MEC) is one of the key technologies in 5G due to the remarkable increase in perceived user experience by reducing the transmission latency significantly and obtaining higher bandwidth locally. Meanwhile, for the various and large scale of IoT applications, an application-aware MEC server can improve the network efficiency to achieve the requirements of 5G networks. To understand the benefits of the novelty mechanism, an open source solution is used to validate it in the next-generation networks. In this paper, the application-aware MEC server is studied by network simulator and the network performance is analyzed under architectures of mobile edge and cloud computing.**

I. INTRODUCTION

In recent years, the promising fifth-generation (5G) mobile technology has rapidly moved forward due to the urgent necessity of low power consumption and low latency for massive Internet of Thing (IoT) communications. Some key technologies relevant to 5G requirements have been addressed in [1], for example, the distributed computing environment can reduce network latency and increase data rate. In September 2014, European Telecommunications Standards Institute (ETSI) formed a mobile edge computing (MEC) Industry Specifications Group (ISG) to facilitate global market growth for vendors, service providers, and third-parties. Based on the standard, mobile operators can create the sustainable business for revenue generating services in the near future. Since more and more mobile users access their data via the cloud storage servers, the network quality may be degraded with higher and higher loading. In this article, we focus on MEC technique to improve the network performance due to the remarkable increase in perceived user experience by reducing the transmission latency significantly and obtaining higher bandwidth locally. MEC can be installed at the eNB of LTE, the Radio Network Controller (RNC) site of 3G, or the multi-technology cell aggregation site of LTE/3G. Besides, there are six key use cases defined in [2]: 1) active device location tracking, 2) augmented reality (AR) content delivery, 3) video analytics, 4) RAN-aware content optimization, 5) distributed content and DNS caching, and 6) application-aware performance optimization.

There have been many previous works focused on the discussion and improvement for MEC architecture to meet the 5G goals. An ACACIA service abstraction framework [3] is implemented to allow the continuous interactive (CI) applications, such as augmented reality (AR), transmitted between the MEC server and mobile devices via dedicated bearers for reducing the end-to-end application latency. Similarly, in [4], a stochastic computation task scheduling policy for MEC systems is proposed to incorporate different timescales in the task execution process and the channel fading process for reducing average delay in various scenarios. To always ensure high user experience, a Follow Me Edge (FME) architecture is proposed [5] and let the video streaming services move across edge servers according to the user's velocity and direction. By offloading the video encoding effort from mobile devices to the MEC server integrated into the IMS subsystem [6], the power consumption of mobile devices can be reduced by approximately 13% from measurement results. A context-aware collaboration platform embedded in the MEC server of middleware is proposed for the examples of road accident scenario and remote robotic telesurgery [7]. Moreover, under the latency constraints, an energy-efficient computation offloading (EECO) scheme [8] is designed to obtain the minimal energy consumption by jointly optimizing computation offloading decisions and radio resource allocation strategies.

With the growth of IoT services in B4G/5G networks, how to manage the network performance is getting more and more important. Therefore, there are several algorithms to improve the downlink QoS that we focused for IoT applications. To reduce the collision rate and access delay of dense IoT devices in a heterogeneous network architecture, a novelty framework and algorithm is proposed [9] by scheduling the radio resources of macro eNB and mini base stations associated with UEs and IoT devices, respectively. Another scheduling algorithm called DP-VT-MLWDF [10] can adopt the size of flow buffer, the average throughput of past time, and the CQI to make resource allocation determination. Thus, not only the performance of real-time (RT) services can be optimized, but that of non-real-time (NRT) services should be remained within an acceptable level. A scheme called context-aware dynamic resource allocation (CADRA) mechanism for cellular M2M communications [11] can improve the random access performance by a two-phase

978-1-5386-1542-3@2017 APSIPA

method: Phase I for the estimation of random access attempts, and Phase II for resource allocation. A packet scheduling algorithm called Decoupled-level with QoS-Aware (DLQA) was proposed by Trabelsi *et al.* [12] for LTE downlink to keep reasonable values of fairness and user throughput according to GBR/Non-GBR bearers and time/frequency domains. Mehmood *et al.* [13] proposed a data aggregation scheme for downlink M2M traffic in LTE-A to utilize the radio resources of the Un interface efficiently. Similarly, a traffic flow-based and dynamic grouping-enabled resource allocation algorithm for LTE-D2D vehicular networks is proposed [14] to improve the utilization of spectrum resources by the vehicle speed and two link types which are Traffic & Safety information transmission and Recreation & Entertainment information interaction.

The rest of this paper is organized as follows. Section II introduces the application-aware MEC. The simulation environment is described in Section III. Section IV shows the simulation results along with discussions. Finally, Section V contains our concluding remarks and future work.

II. APPLICATION-AWARE MOBILE EDGE COMPUTING

ETSI ISG has standardized a mobile edge computing framework [15] as show in Fig. 1, which consists of four entities, i.e. mobile edge host, mobile edge system level management, mobile edge host level management, and network level entities. Furthermore, the Mobile edge host level is composed of Mobile edge host and the corresponding Mobile edge host-level management. Mobile edge host is further divided to the Mobile edge platform, the Mobile edge applications which are deployed on virtual machines (VMs) and adjustable by the characteristics of applications, and the virtualization infrastructure. The networks level consists of related external entities that are the 3GPP cellular network, the local networks, and the external networks. On top of the Mobile edge system level management is responsible for the overall view of the mobile edge system by the important functionality of the Mobile edge orchestrator.

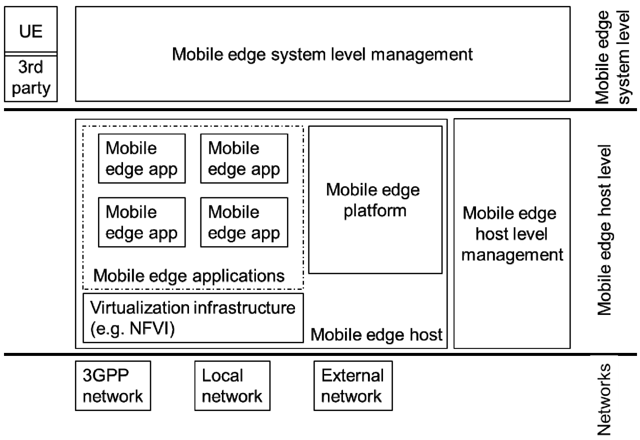

Fig. 1 Mobile Edge Computing framework.

ISG MEC has published the technical requirements on each entity for use cases to enable a large number of new features. E.g., mobile edge system shall reuse the Network Functions Virtualization (NFV) infrastructure and the management functionality. Mobile edge management shall 1) identify which mobile edge services that a mobile edge application requires to run, 2) deploy mobile edge applications on different mobile edge hosts, and 3) maintain connectivity between a UE and an application instance when the UE performs a handover to another cell. Mobile edge platform shall 1) use available radio network information to optimize the mobility procedures, 2) be capable of providing mobile edge services that can be consumed by authorized mobile edge applications, 3) allow authorized mobile edge applications on the same mobile edge host to communicate with each other, 4) provide access to persistent storage space to an authorized mobile edge application, 5) allow authorized mobile edge applications to send/receive user plane traffic to UEs, and 6) provide a capability of supplying Coordinated Universal Time (UTC) time of day information to the authorized mobile edge applications, etc. MEC can serve an important role in improving wireless system performance, reducing the cost of operation, and promoting the creation of new applications to telecommunication operators by the following use case categorizations: 1) consumer-oriented services, including gaming, remote desktop applications, augmented reality, cognitive assistance, etc., 2) operator and third party services, including active device location tracking, big data, security/safety, enterprise services, etc., and 3) network performance and QoE improvements, including content/DNS caching, performance optimization, video optimization, etc.

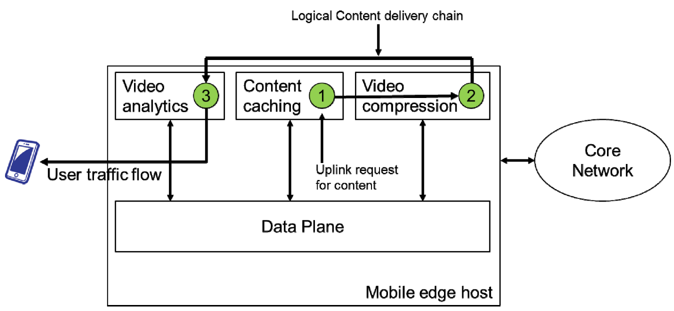

Fig. 2 Video content delivery.

Furthermore, for the use case of video optimization, if the uplink request arrives at the mobile edge platform for video streaming, it is routed to the content caching ① application in order to retrieve the content as shown in Fig. 2. Once the content is identified, the user traffic needs to be passed to the video compression ② and video analytics ③ application before delivered to the end user. The platform needs to support this scenario, whereby it will classify the traffic and then steer the traffic through multiple applications. By means of the logical content delivery chain, the video users can get better user experience by comparing with video server in mobile cloud computing (MCC) environment. Some use cases for IoT on MEC platform can increase performance, safety, and efficiency [16]. For example, to satisfy the tight latency requirement of connected car communications, MEC can be implemented in the highly distributed mobile network environment. Accordingly, an example of different types of UEs residing in LTE network with a MEC server is depicted in Fig. 3. Video and FTP UEs represent the applications of RT and NRT traffic, respectively. Under the varied number of IoT UEs, the network performance is investigated while a MEC server is deployed at or near the eNB. Then, the downlink traffic flow is focused to observe the comparison between MEC and mobile cloud computing (MCC).

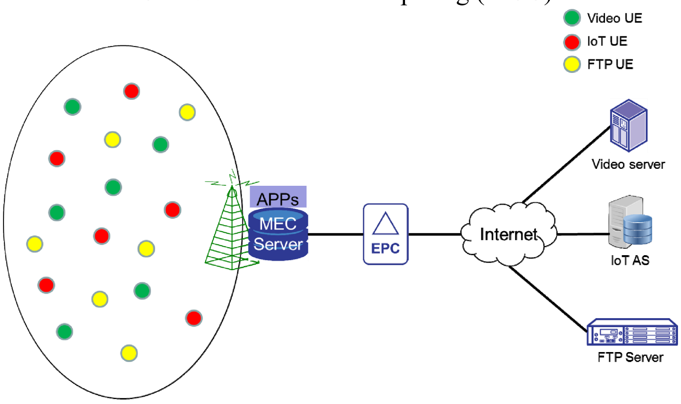

Fig. 3 An example of MEC architecture.

III. SIMULATION ENVIRONMENT

TABLE I PARAMETERS

Parameters	Values
Coverage of an eNB (m)	1000
Number of Video UEs	5, 10
Number of IoT UEs	5 to 100
Number of FTP UEs	5, 10
Packet size of a video UE (Bytes)	1500
Packet size of an IoT UE (Bytes)	512
Packet size of an FTP UE (Bytes)	1500
Data rate of video application (bps)	7.5M, 12M
Data rate of IoT application (bps)	500k
Data rate of FTP application (bps)	30M
Simulation time (sec)	30

Network Simulator 3 (ns-3) is an open source of simulation platform published under the GNU GPLv2 license for computer networking research and education, and has extensive supported for LTE modelling and comprehensive statistics collection tool which is critical for such an experiment. The experiment simulated three types of traffic originating from remote or MEC servers and bound to its own type of UEs as shown in Fig. 3. The traffic types were as follows: Video Streaming traffic that was created with UDP packet stream with a packet size of 1500 bytes, FTP traffic that was created with TCP packet stream with the identical packet size of 1500 bytes, and IoT traffic that was done with UDP packet stream with smaller packet size of 512 bytes. We varied the amount of UEs used for IoT traffic, thus changing the network load, and observed traffic characteristics such as packet interval in two types of scenarios: when remote or MEC servers were used for traffic transmission. The physical architecture that we used consists of one eNB that was located in the middle of the circle with radius 1000m. All three types of UEs were placed inside this circle using uniform random function for UE placement. eNB was connected to single EPC, which in turn was connected with all remote servers and a MEC server using Point-to-Point connection with different connection characteristics for remote servers and MEC server. Based on the attributes of Point-to-Point model in ns-3, we have configured the delay of all three remote servers (namely Video server, IoT application server (AS) and FTP server) which are twice more than that of MEC server.

All other parameters and values were listed in Table I. Using this approach, we simulated a MEC server that was physically closer to UEs (with means of delay). The network layer was done using IPv4 and static routing installed on each node of the simulation for the minimal overheads regarding routing. The traffic generation was done using `PacketSinkHelper` with `UdpClientHelper` modules for UDP traffic and `PacketSinkHelper` with `OnOffHelper` for TCP traffic. The traffic statistics was calculated using `flowMonitor` module that calculates traffic information on per-flow basis and outputs data in XML format. This information was then parsed using python scripts to first extract data from XML file and then calculate mean values of traffic parameters and output them in CSV file, ready for analysis.

IV. SIMULATION RESULTS

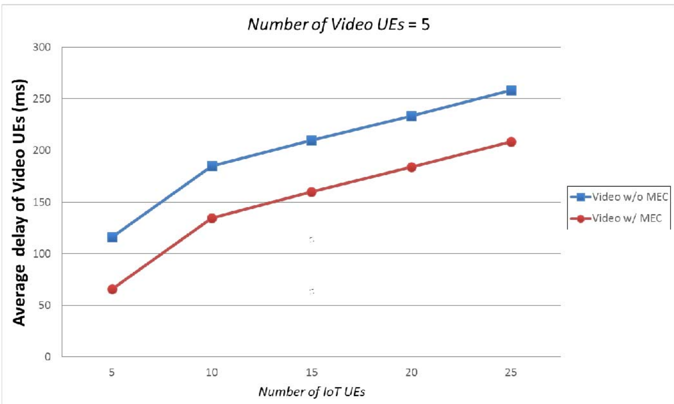

Fig. 4 Average delay of video UEs.

To investigate the impact of deploying an MEC server in the networks, we first compared the video content downloaded from the local and cloud respectively while other applications (i.e., IoT and FTP) are communicated with cloud servers. Fig. 4 shows the average delay of video UEs as the number of IoT UEs increase s from 5 to 25. It is interesting to notice that the average delay of video UEs can be significantly reduced by 23.9% as the number of IoT UEs is 15 while downloading video from MEC server. The reason is because that the mobile edge host for computing equipment is installed near base station to decrease the transmission delay.

Unlike centralized cloud servers, MEC server can be managed locally by the telecommunication operators.

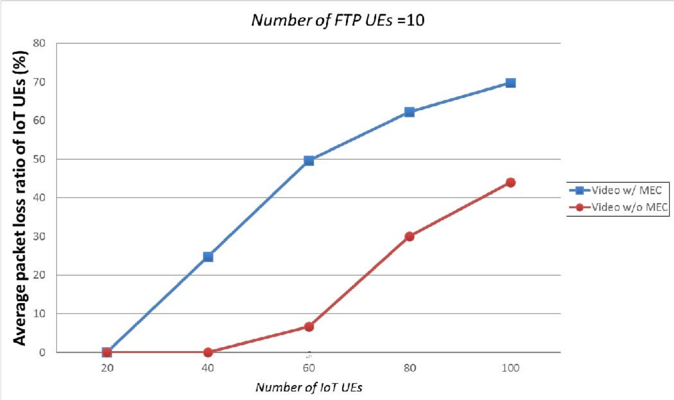

Fig. 5 Average packet loss ratio of IoT UEs.

Next, let us investigate the impact of downloading the video with or without the MEC server for video UEs while the IoT UEs are always communicated with the MEC server. Fig. 5 shows the average packet loss ratio (PLR) of IoT UEs is increased as the number of IoT UEs increases from 20 to 100. In the figure, it can be also observed that the PLR of IoT UEs as video with MEC server is higher than that without MEC server. This result reveals that higher number of UEs communicated with the MEC server have higher possibility to invoke the packet loss because of the limited resource. It is probable that the lost packets may cause damage to the individual for some time-sensitive environments, e.g., connected vehicles. Therefore, how to manage the traffic transmitted via cloud or local servers becomes an important issue in the next-generation networks.

V. Conclusions and Future Work

This paper has presented the impact of application-aware MEC server deploying in the network with various traffic types and analyzed by open source network simulator. The simulation results show that the average delay of UEs can be decreased significantly when the MEC server is deployed in the network. However, the network performance is degraded because of increasing the number of UEs which connect to the same MEC server. Consequently, not only the traffic type of applications is considered, but also performance constraints such as delay or packet loss ratio have to be taken into account. Although the MEC server has been implemented in LTE networks, it still needs to be completed the full functionalities, including the resource consumption of deploying VMs for the upcoming 5G technologies and the scheduling mechanism for various applications.

References

[1] I. F. Akyildiz, S. Nie, S.-C. Lin, and M. Chandrasekaran, “5G roadmap: 10 key enabling technologies,” Computer Networks, Vol. 106, Page(s): 17-48, Jun. 2016.

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

[3] J. Cho, K. Sundaresan, R. Mahindra, J. V. d. Merwe, and S. Rangarajan, “ACACIA: Context-aware Edge Computing for Continuous Interactive Applications over Mobile Networks,” CoNEXT '16 Proceedings of the 12th International on Conference on emerging Networking EXperiments and Technologies, Page(s): 375-389, Dec. 12-15, 2016.

[4] J. Liu, Y. Mao, J. Zhang, and K. B. Letaief, “Delay-optimal computation task scheduling for mobile-edge computing systems,” 2016 IEEE International Symposium on Information Theory (ISIT), Jul. 10-15, 2016.

[5] T. Taleb, S. Dutta, A. Ksentini, M. Iqbal, and H. Flinck, “Mobile Edge Computing Potential in Making Cities Smarter,” IEEE Communications Magazine, Vol. 55, Issue 3, Page(s): 38–43, Mar. 2017.

[6] M. T. Beck, S. Feld, A. Fichtner, and C. L.-Popien, “ME-VoLTE: Network functions for energy-efficient video transcoding at the mobile edge,” 2015 18th International Conference on Intelligence in Next Generation Networks (ICIN), Feb. 17-19, 2015.

[7] S. Nunna, A. Kousaridas, M. Ibrahim, M. Dillinger, C. Thuemmler, H. Feussner, and A. Schneider, “Enabling Real-Time Context-Aware Collaboration through 5G and Mobile Edge Computing,” 2015 12th International Conference on Information Technology - New Generations (ITNG), Apr. 13-15, 2015.

[8] K. Zhang, Y. Mao, S. Leng, Q. Zhao, L. Li, X. Peng, L. Pan, S. Maharjan, and Y. Zhang, “Energy-Efficient Offloading for Mobile Edge Computing in 5G Heterogeneous Networks,” IEEE Access, Vol. 4, Page(s): 5896–5907, Aug. 2016.

[9] P. K. Wali and D. Das, “A novel access scheme for IoT communications in LTE-Advanced network,” 2014 IEEE International Conference on Advanced Networks and Telecommuncations Systems (ANTS), Dec. 14-17, 2014.

[10] Y.-P. Li, B.-J. Hu, H. Zhu, Z.-H. Wei, and W. Gao, “A delay priority scheduling algorithm for downlink real-time traffic in LTE networks,” 2016 IEEE Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), May 20-22 2016.

[11] Y.-C. Pang, G.-Y. Lin, and H.-Y. Wei, “Context-Aware Dynamic Resource Allocation for Cellular M2M Communications,” IEEE Internet of Things Journal, Vol. 3, Issue 3, Jun. 2016.

[12] S. Trabelsi, A. Belghith, F. Zarai, and M. S. Obaidat, “Performance Evaluation of a Decoupled-Level with QoS-Aware Downlink Scheduling Algorithm for LTE Networks,” 2015 IEEE International Conference on Data Science and Data Intensive Systems (DSDIS), Dec. 11-13, 2015.

[13] Y. Mehmood, C. Görg, and A. T.-Giel, “A radio resource sharing scheme for IoT/M2M communication in LTE-A downlink,” 2016 IEEE International Conference on Communications Workshops (ICC), May 23-27, 2016.

[14] G. Li, Z. Yang, S. Chen, Y. Li, and P. Yuan, “A traffic flow-based and dynamic grouping-enabled resource allocation algorithm for LTE-D2D vehicular networks,” 2016 IEEE/CIC International Conference on Communications in China (ICCC), Jul. 27-29, 2016.

[15] “Mobile Edge Computing (MEC); Framework and Reference Architecture,” ETSI GS MEC 003 V1.1.1 (2016-03).

[16] “Mobile Edge Computing A key technology towards 5G,” ETSI White Paper No. 11, Sep. 2015.