Prioritized Traffic Shaping for Low-latency MEC Flows in MEC-enabled Cellular Networks

Po-Hao Huang, Fu-Cheng Hsieh, Wen-Jen Hsieh, Chi-Yu Li, Ying-Dar Lin
Department of Computer Science, National Yang Ming Chiao Tung University, Hsinchu, Taiwan
Email: huangpoh.cs06g@nctu.edu.tw, {fchsieh, hsiehwj, chiyuli, ydlin}@cs.nctu.edu.tw

2022 IEEE 19th Annual Consumer Communications & Networking Conference (CCNC) | 978-1-6654-3161-3/22/$31.00 ©2022 IEEE | DOI: 10.1109/CCNC49033.2022.9700715

***Abstract*—Multi-access edge computing (MEC) has been introduced as an enabler of low-latency performance in 4G/5G cellular networks. For the MEC-enabled cellular networks, several deployment options have been proposed by ETSI. One promising deployment option called Bump-in-the-wire does not require changes on the base station or the core network, so it has the advantage of easy deployment and low cost. However, the unchanged base station connecting to an MEC platform cannot differentiate MEC traffic from Internet traffic or prioritize it; its traffic congestion may thus cause the MEC traffic to suffer from high latency. In this work, we thus design a solution, designated PTS-MEC (Prioritized Traffic Shaping for MEC), to control the forwarding of downlink MEC/Internet traffic at the MEC and prioritize the MEC traffic based on a hierarchical MEC-prioritized fair service model. PTS-MEC alleviates the base station's traffic congestion with a latency-aware service rate adaptor at run time by applying the service curve concept to delaying or/and skipping the Internet traffic. We prototype PTS-MEC on an open source MEC platform and evaluate it with congested cases. The evaluation result confirms the effectiveness of PTS-MEC; it can satisfy latency goals, e.g., 50 ms at the 90th percentile, within 3.70% error for MEC flows while fairly allocating remaining resource to non-MEC UEs.**

***Index Terms*—Multi-access edge computing, MEC, cellular network, low latency**

I. Introduction

Multi-access Edge Computing (MEC) is a concept that deploys a cloud computing system at the edge of any network. Its technology is designed to be collocated with base stations or other edge nodes. It enables application servers close to end devices, thereby eliminating network congestion and reducing end-to-end delay. It is anticipated to benefit many near-future applications with low-latency or/and high-bandwidth demands (e.g., VR, AR, and V2X). These demands push MEC-related technologies to be growing explosively. A research study [1] shows that the global edge computing market size is projected to reach USD 3.24 billion by 2025 with a compound annual growth rate of 41.0% during the forecast period.

For the MEC deployment in 4G/5G cellular networks, there have been several potential options proposed by ETSI [2]. One promising deployment option called Bump-in-the-Wire (BIW) is to deploy an MEC platform to sit on the interface between the base station and the core network. It does not require changes on the cellular components [3], so it has the advantage of easy deployment and low cost. However, the unchanged base station connecting to an MEC platform cannot differentiate between the traffic coming from the MEC and Internet traffic from the core network. It treats them equally so any traffic congestion may cause the MEC traffic to suffer from high latency and offset the low-latency benefit of the MEC deployment.

We conduct an experiment to examine the above high-latency issue of MEC traffic on an MEC-enabled LTE platform [4]. We consider round trip time (RTT) of TCP sessions between user equipment (UE) and the MEC as the latency performance. Take a heavy congestion case as an example, where an MEC UE has a 15 Mbps downlink TCP flow from the MEC platform and three non-MEC UEs have a total of 60 Mbps downlink traffic. The aggregate traffic volume exceeds the base station's capacity. The experimental result shows that the RTT values of the MEC flow are increased from 40.54 ms and 51.85 ms in the case without non-MEC traffic to 90.13 ms and 169.22 ms at the 50th and 90th percentiles, respectively. The heavy congestion increases the latency by 2.22 and 3.26 times, respectively. The result shows that the low-latency gain from the MEC deployment may be offset in such kind of congestion cases.

In this work, we design a solution that does prioritized traffic shaping for MEC traffic, designated PTS-MEC. PTS-MEC controls the forwarding of downlink MEC/Internet traffic at the MEC, which all the traffic flows between the base station and the core network traverse. It then prioritizes the MEC traffic based on a hierarchical MEC-prioritized fair service model. It applies the concept of service curve [5] to alleviating the base station's traffic congestion by delaying or/and skipping the Internet traffic with a latency-aware service rate adaptor at run time. It seeks to trade off minimum service rates of non-MEC traffic for the latency requirement of MEC flows, while keeping fairness among non-MEC UEs.

We build an MEC-enabled LTE platform and prototype PTS-MEC on it for evaluation. The evaluation result shows that PTS-MEC can successfully satisfy latency goals of MEC traffic in congested cases. Specifically, in one heavy congested case, it can satisfy a latency goal, 50 ms at the 90th percentile, within 3.70% error for MEC flows while fairly allocating remaining resources to non-MEC UEs.

The rest of this paper is organized as follows. Section II describes the MEC background and related work. The impact of congested base stations on Low-latency MEC flows is analyzed in Section III. Sections IV, V, and VI design, implement, and evaluate the proposed PTS-MEC solution, respectively. Section VII concludes the paper.

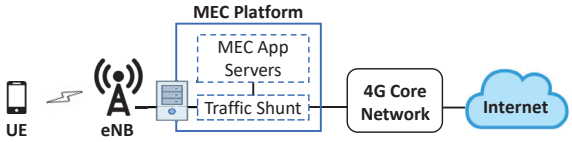


Fig. 1: 4G LTE network architecture with MEC integration

II. Background and Related Work

A. 4G LTE Network Architecture with MEC Integration

The 4G LTE network consists of two major components: core network and radio access network. The core network not only offers control-plane functions including mobility, authentication, and so on, but also forwards user-plane packets between the radio access network and Internet. The radio access network is composed of UE and base station called evolved Node B (eNB). To enable edge computing capability, the 4G LTE network can be integrated with an MEC platform, as shown in Figure 1.

There have been four integration methods introduced by ETSI [2]; in this work, we focus on the BIW approach, which can retain current 4G operations without any need of modifications on the existing network components [3], [6]. The MEC platform is deployed next to the eNB by sitting on the S1 interface connecting the eNB and the core network. As shown in Figure 1, a traffic shunt is required to redirect MEC traffic to MEC application servers but forward the others including Internet and control-plane traffic to the core network. To let the MEC traffic be served, the traffic shunt needs to understand the GPRS tunneling protocol (GTP), which handles user-plane traffic transport on the S1; it strips off GTP encapsulation from uplink GTP packets sent towards MEC application servers, but encapsulates the traffic coming from the servers with GTP and forwards it to the eNB. Note that the BIW method may take overhead to address its security and charging issues.

B. Related Work

Several initial MEC studies propose solutions [3], [7]–[9] to address implementation/deployment issues of MEC in cellular networks. [7] and [8] implement an MEC platform using newly defined interfaces and a software-defined-network architecture, respectively. Another study [9] integrates MEC functions into the eNB. All of these solutions are not standard-compliant since they require modifications on the eNB or/and the core network. [3] implements and deploys an MEC platform in cellular networks using the BIW approach. It is standard-compliant with packet engineering functions that redirect MEC traffic while retaining original functions, and holds MEC application servers using virtualization technologies such as OpenStack. In this work, we demonstrate an interference issue on the latency performance of MEC traffic and evaluate our proposed solution using this MEC platform.

Most of the MEC studies about the latency performance focus on computation offloading problems [10]–[17]. Liu et al. [10] optimize the task scheduling policy by trading off between average delay and power consumption. Chen et al. [11] optimize caching strategy to minimize network latency subject to available resources, whereas Siew et al. [12] employ a dynamic pricing mechanism to achieve resource sharing on the MEC. Another few studies [13], [17] address the computation offloading problems by considering offloading decision, computing resource allocation, and mobility management on the MEC; the others [14], [15] study load-balancing problems for IoT devices connecting to the MEC. However, neither of the existing studies addresses the issue that Internet traffic may cause interference on MEC traffic and hurt its latency performance. In this work, we propose a solution to ensure MEC traffic to achieve low latency under the interference.

III. Impact of Congested Base Stations on Low-latency MEC Flows

We examine the latency of MEC traffic flows in an MEC-enabled LTE network. We mainly consider downlink traffic flows, which take the major portion of application traffic. The MEC and non-MEC traffic flows coexist in the network; in the downlink direction, they are sent from application servers on the MEC platform and on the Internet, respectively, to UE. The MEC traffic can gain low latency from the deployment of application servers on the MEC next to eNB.

However, when the eNB treats the MEC and non-MEC traffic flows as the same traffic type, the latency of the MEC traffic could be hurt by a congested eNB. Each eNB contains a resource scheduler that schedules transmission resource for its connected UEs. The scheduling algorithm usually considers available resources, UE channel conditions, and UE queue statuses while achieving UE fairness. Without any new mechanism along with the MEC deployment on the eNB, the scheduler does not recognize MEC flows or even prioritize them when the eNB is congested.

A. An Illustrative Example

Consider that there is a backlog of MEC and non-MEC traffic packets on an LTE eNB. Since the packets are not differentiated, the eNB schedules them based on its default scheduling algorithm that may consider the order of packet arrivals and a UE fairness policy. Figure 2 shows a simple example that radio resource blocks (RBs), which are the smallest unit of radio resource, are allocated as MEC RBs to one UE for an MEC flow and as non-MEC RBs to two UEs for their non-MEC flows. The left-side figure shows a possible condition that MEC RBs interlaced with non-MEC ones are distributed over time and the scheduled MEC packets may suffer long delays. When the MEC RBs can be prioritized, smaller delays of the MEC packets can be obtained, as shown in the right-side figure. Moreover, when the volume of non-MEC packets is large and then the eNB is congested, the MEC flows without any high priority could be assigned only small amount of RBs and thus experience long queueing delays.

B. An Experimental Case Study

We conduct an experiment to validate the latency issue of MEC flows using an MEC-integrated LTE platform. In

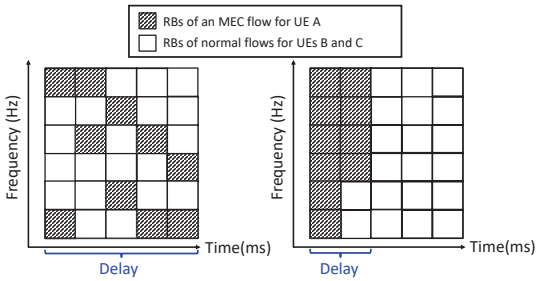


Fig. 2: Illustration of the MEC traffic latency based on allocation of radio bearers on an eNB.

the following, we first introduce experimental platform and methodology, and then present experimental results.

Experimental platform and methodology. Building an MEC-integrated LTE platform mainly requires an LTE network platform, an MEC platform, and UE devices. We use an open source LTE core network, NextEPC, and install it on a PC with Intel Core i7-7700 3.60GHz CPU and 16GB RAM. A commercial LTE small cell, WNC OSQ4G-01E2, is employed as the eNB connecting to the core network. For the MEC platform, we use an open source MEC solution [3], [4] on a commercial server platform, Lanner NCA 6210, and connect it to the LTE network. The UE devices include smartphones with phone model, HTC U11-Eyes, and laptops with model, CJSCOPE SY-250; each laptop is equipped with an LTE dongle, Huawei e3372h, connecting to the eNB.

We generate two types of traffic flows, MEC and non-MEC flows, using `Iperf3` and measure RTT of the MEC traffic under various traffic cases. For each test, there is an MEC flow sent from the MEC to a UE, designated as MEC UE, with TCP traffic. The RTT is calculated as a time period between the time that each TCP data packet is sent and the time that its sequence numbers are acknowledged. To generate background traffic to emulate congestion on the eNB, non-MEC flows are sent from the core network to non-MEC or/and MEC UEs with UDP traffic. Each test has 3 runs with 2 minutes each. Note that since we mainly consider the latency of MEC flows under congestion conditions on the eNB, we minimize the impact from wireless channel variations by putting all the UEs around the eNB with good channel conditions.

In the following, we consider three cases: light, medium, and heavy traffic congestion at the eNB. In each case, we have one MEC UE with a downlink TCP flow bounded by 15 Mbps; we vary the number of non-MEC UEs from 0 to 3 and distribute non-MEC traffic volume to the non-MEC UEs evenly. Take Case I as an example. Given 30 Mbps non-MEC traffic volume, each non-MEC UE has a downlink UDP traffic flow with 30 Mbps, 15 Mbps, or 10 Mbps when the number of the non-MEC UE(s) is 1, 2, or 3, respectively.

Case I: not exceeding capacity with light congestion. We generate 30 Mbps non-MEC traffic from the core network to non-MEC UE(s) and plot RTT's CDF for the MEC flow, as shown in Figure 3a. Since the total traffic amount, 45 Mbps, does not exceed the capacity and causes only light congestion on the eNB, those three congestion cases with 1/2/3 non-MEC UEs have similar RTT distribution of the MEC flow to the case without any non-MEC traffic, noted as '0 UE_{BK}', which stands for background UE. Specifically, all the cases have similar median RTT values between 39.7 ms and 41.2 ms with only 1.5 ms difference; the case '0 UE_{BK}' has only 50.10 ms RTT at the 90th percentile, whereas the other cases have that RTT up to 58.51 ms within only 17% difference.

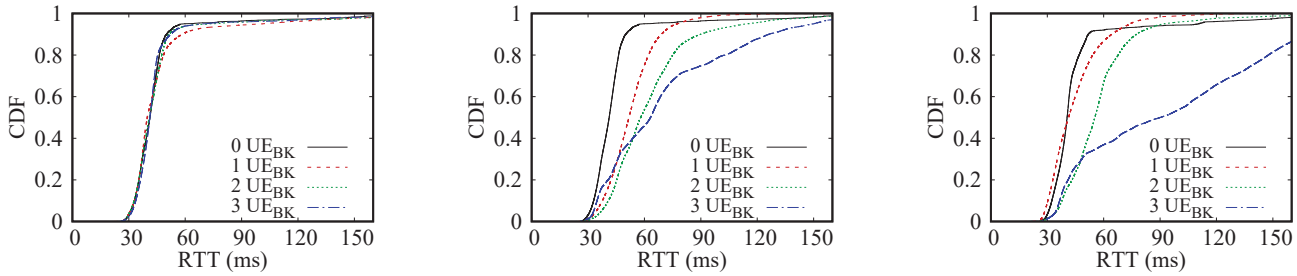
Case II: exceeding capacity with medium congestion. In addition to the 30 Mbps non-MEC traffic for non-MEC UE(s), we generate another 10 Mbps non-MEC UDP flow from the core network to the MEC UE. The total traffic volume, 55 Mbps, exceeds the eNB's capacity, which is about 50 Mbps from nearby UEs. As shown in Figure 3b, the RTT values increase with the number of non-MEC UEs; the median values for the cases with 0/1/2/3 non-MEC UEs are 40.54 ms, 51.74 ms, 58.02 ms, and 63.23 ms, respectively, whereas those at the 90th percentile are 50.10 ms, 68.99 ms, 89.97 ms, and 126.22 ms, respectively. The RTT values of the MEC flow can be increased by up to 1.56 and 2.52 times at the median and 90th percentile, respectively. The reason is that the eNB provides UE fairness and then fairly allocates resources to connected UEs without prioritizing MEC traffic. It not only makes the MEC traffic be delayed but also prevents the MEC flow from growing to the bandwidth bound of 15 Mbps.

Case III: exceeding capacity with heavy congestion. We generate 60 Mbps non-MEC traffic from the core network to non-MEC UE(s) and the RTT result is shown in Figure 3c. The RTT values of the MEC flow are increased from 40.54 ms and 51.85 ms up to 90.13 ms and 169.22 ms at the median and 90th percentile, respectively, from the case with three non-MEC UEs. They achieve 2.22 and 3.26 times, respectively. The larger delay increases come from the greater amount of the non-MEC traffic.

IV. PTS-MEC Design

In this section, we seek to design a solution that can achieve low latency for MEC flows when downlink traffic congests the eNB. By keeping the merit of the BIW deployment solution that no existing cellular components need to be modified, we prevent any changes from the eNB. To this end, we aim to regulate the MEC and Internet traffic flows which pass through the traffic shunt at the MEC. We propose a solution designated PTS-MEC (Prioritized Traffic Shaping for MEC) to prioritize MEC traffic and alleviate the eNB's traffic congestion, if there are any, by delaying/skipping the forwarding of Internet traffic. The MEC traffic flows are given higher priority than the other Internet ones, so their delay requirements are satisfied before the Internet flows are served. Any remaining resources at the eNB to serve the Internet flows are fairly shared among UEs with Internet traffic.

For the prioritized traffic shaping, we apply the concept of the service curve [5]. Consider an example scenario with MEC and non-MEC UEs in Figure 4. Each UE k gets assigned an

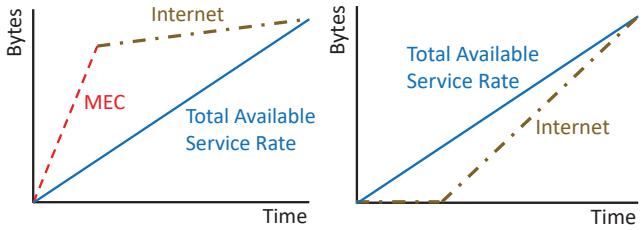


(a) Case I: 30 Mbps non-MEC traffic towards non-MEC UEs.

(b) Case II: 30/10 Mbps non-MEC traffic towards non-MEC/MEC UEs, respectively.

(c) Case III: 60 Mbps non-MEC traffic towards non-MEC UEs.

Fig. 3: CDF of RTT for a 15 Mbps TCP traffic flow towards an MEC UE in various congestion cases with different non-MEC traffic amounts and varying number of non-MEC UEs ($x\ UE_{BK}$ indicates the non-MEC traffic shared by x non-MEC UEs.).



(a) MEC UE with MEC and Internet traffic

(b) non-MEC UE with only Internet traffic

Fig. 4: Service curves for an MEC UE and a non-MEC UE which coexist in an example scenario

available service rate s_k, which is considered as forwarded data bits per time unit from the traffic shunt to the eNB. As shown in Figure 4a, the MEC traffic of the MEC UE is forwarded first and given a higher service rate m_k; its Internet traffic is forwarded with a lower service rate i_k afterwards, if there is any. When the requested MEC traffic rate a_k is greater than s_k, the MEC UE is not allowed to have other Internet traffic to be transmitted. Otherwise, its Internet traffic is transmitted under the constraint of s_k given the MEC traffic has been served. For the non-MEC UE, the traffic forwarding is delayed for the prioritized forwarding of the MEC traffic, as shown in Figure 4b. The transmitted rate does not exceed the available service rate. Note that each UE's available service rate is a fair share from an overall service rate S. We can then adjust S to limit the volume of Internet traffic to reach the eNB so that the eNB's congestion level can be adapted.

We thus propose a hierarchical MEC-prioritized fair service model for the prioritized traffic shaping and design a module called latency-aware service rate adaptation to adapt the eNB's congestion level based on runtime RTT of MEC traffic flows. We elaborate on each of them below.

A. Hierarchical MEC-prioritized Fair Service Model

This hierarchical service model is composed of three levels: overall available service rate (S) at the top, each UE's service rate (s_k) at the medium, and MEC/Internet service rates (m_k/i_k) for each UE at the bottom. Figure 5 shows an example including two MEC UEs with both MEC and Internet traffic flows, and one non-MEC UE. Given n UEs connected to the eNB, the overall service rate is evenly distributed to them, i.e., $s_k = S/n$. Its runtime operation consists of two steps. First, all the MEC traffic flows with higher priority are to be served and satisfied, such as m_1 and m_2 in the example; the aggregate service rate of the MEC flows is constrained by S. Second, if there is any spare bandwidth, it is distributed to the UEs with Internet traffic proportionally based on their available service rates. For each non-MEC UE, the available service rate is s_k. For each MEC UE, it is $r = s_k - m_k$ if $r > 0$; otherwise, it is 0. It means that when an MEC UE has MEC traffic more than its available service rate, its Internet traffic flows are not served. In this case, the MEC traffic reduces the total available service rate of other non-MEC UEs, but it is still evenly shared by the non-MEC UEs.

In this model, we can adjust S to control how much Internet traffic can reach the eNB while MEC flows are prioritized and the fairness of non-MEC UEs is maintained. If the latency requirement of the MEC flows is not satisfied, it means that the eNB is too congested so that S can be reduced. However, S should be as large as possible so that non-MEC UEs can receive maximum service rates. Thus, we design a mechanism of the latency-aware service rate adaptation below.

B. Latency-aware Service Rate Adaptation

We consider that the latency requirement of MEC traffic flows at an MEC platform is given as α ms delay at a certain β percentile. The RTT values of TCP sessions from the MEC traffic are considered as delays. It is monitored based on a moving window with γ RTT samples. Each delay sample, noted as α_s, is collected from the RTT sample at the β percentile within each sampling window. In order to not only make the delay satisfy the latency requirement but also minimize the impact on the non-MEC traffic flows, we seek to keep the delay α_s oscillating between $\alpha - c\mu$ and $\alpha - \mu$, where c is a scale factor and μ is a buffer period. When $\alpha_s < \alpha - c\mu$, where the latency requirement can be easily achieved with spare bandwidth, the overall available service rate S is increased by θ, which is the service rate's

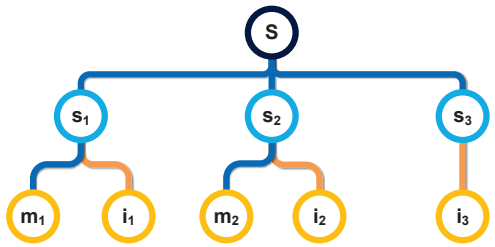


Fig. 5: An example of three UEs in the hierarchical MEC-prioritized fair service model. S represents overall available service rate; s_i, m_i, and n_i represent the available, MEC, and Internet service rates, respectively, for UE i.

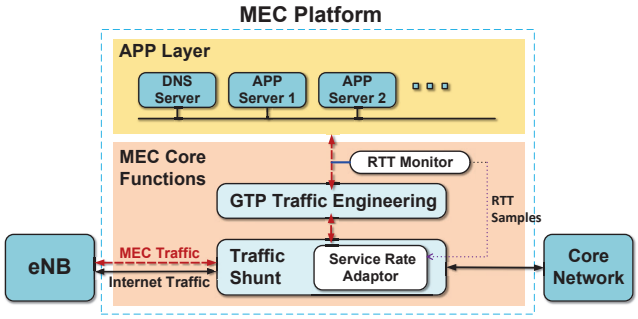


Fig. 6: Our MEC architecture with the PTS-MEC design

tuning granularity; when $\alpha_s > \alpha - \mu$, where the requirement is going to be violated, S is decreased by θ. Note that in order to prevent non-MEC UEs from bandwidth starvation, we set a minimum service rate m to each UE.

V. Implementation

We build an MEC-integrated LTE platform based on an open source solution from [4]. Figure 6 shows the MEC architecture with the PTS-MEC design, and the platform setting is described in Section III-B. We set up the traffic shunt to redirect MEC traffic between the eNB and the MEC app layer while forwarding Internet traffic. The GTP traffic engineering module is deployed to (de)encapsulate GTP packets sent from/to the app layer.

We mainly develop two modules: a RTT monitor and a service rate adaptor. The RTT monitor hooks an interface to monitor TCP packets and measures RTT on TCP DATA and ACK packets with the same sequence number. The service rate adaptor maintains a deficit and per-flow queues for each UE. The deficit is used to log transmitted bits and calculate the UE's service rate over time. It uses the latency-aware service rate adaptation to do packet forwarding. The parameters described in Section IV-B are set as follows: $\gamma = 100$, $c = 2$, $\mu = 2$ ms, $\theta = 0.8$ Kbps, and $m = 5$ Mbps.

Moreover, we use the Intel DPDK (Data Plane Development Kit) to accelerate packet process time. Three cores or threads are used to receive packets, put packets in UE queues, and forward/schedule packets, respectively.

VI. Evaluation

We evaluate the PTS-MEC design based on the prototype shown in Figure 6. Two congested cases are considered: Case III, which is described in Section III-B, and Case IV, where there are two MEC UEs with a 15 Mbps TCP flow each and 1-3 non-MEC UEs with a total of 40 Mbps flows. The experiment methodology is similar to that in Section III-B.

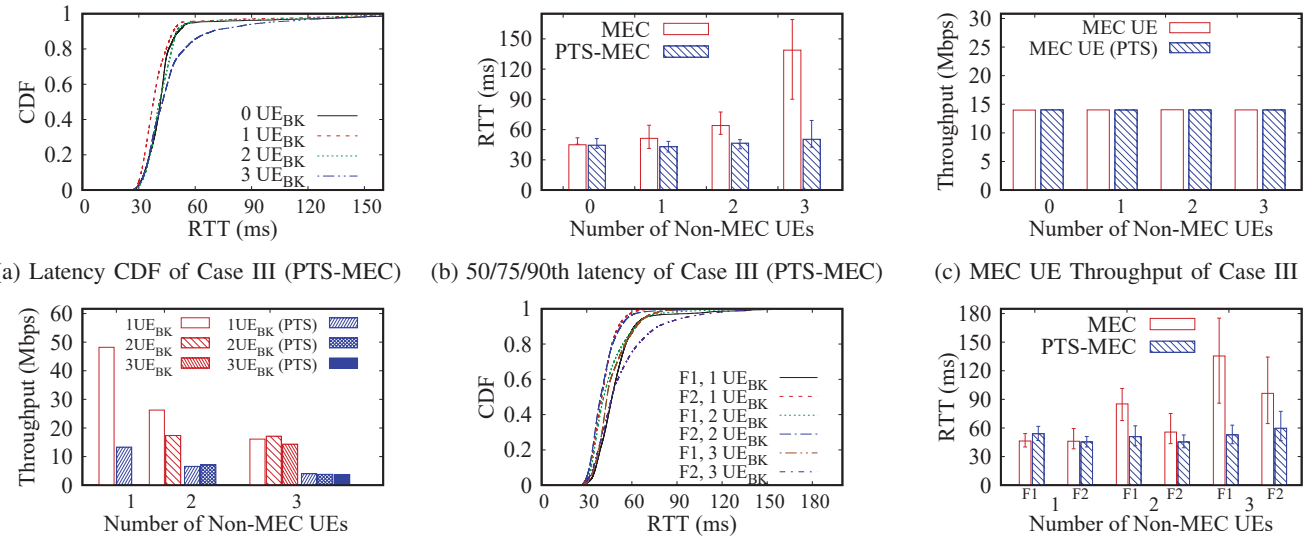
A. Traffic congestion with a single MEC flow

We do the experiment of Case III on the PTS-MEC prototype and compare the result with that of the default MEC in Figure 3c. In this case, we set the latency requirement to be 50 ms RTT, which is the latency requirement for a 4K panoramic VR video [18], at the 90th percentile, i.e., $\alpha = 50$ and $\beta = 90$. The results of latency CDF and 50/75/90th percentile latency are plotted in Figures 7a and 7b, respectively. At the 90th percentile, PTS-MEC can achieve 51.13 ms, 48.33 ms, 50.08 ms, and 69.10 ms RTT in the settings of 0/1/2/3 non-MEC UEs, respectively. It outperforms the default MEC with 51.85 ms, 64.31 ms, 77.46 ms, and 169.22 ms by 1.39%, 24.85%, 35.35%, and 59.17% less latency, respectively. In all the first three settings, PTS-MEC can satisfy the latency goal within 2.26% error. However, in the last setting with three non-MEC UEs, PTS-MEC achieves the 90th RTT latency as high as 69.10 ms; the reason is that it needs to guarantee a minimum service rate 5 Mbps (i.e., $m = 5$) for each UE and the aggregate service rates from those three non-MEC UEs prevent PTS-MEC from achieving the latency goal.

Figures 7c and 7d show the throughput of MEC and non-MEC UEs, respectively, on the PTS-MEC and default MEC platforms. PTS-MEC can satisfy the required traffic amount of the MEC UE, 15 Mbps, with the higher priority. It can be also achieved with the default MEC in all the settings because this amount is still smaller than a fair share amount from the eNB's total bandwidth. As for the non-MEC UEs with the default MEC, the throughput results depend on the eNB's scheduling operation; it does not consider latency so the throughput values are larger than those with PTS-MEC. To alleviate the eNB's congestion by limiting the overall available service rate, PTS-MEC trades off the throughput of non-MEC UEs for the MEC UE's latency requirement. It shows that the available service rate is fairly shared by the non-MEC UEs. For the three non-MEC UE case, the throughput results are suppressed to around the minimum service rate, 5 Mbps.

B. Traffic congestion with two MEC flows

We next examine Case IV for the PTS-MEC and default MEC platforms. The latency requirement of those two MEC flows, F1 and F2, is set to be 60 ms RTT at the 90th percentile, i.e., $\alpha = 60$ and $\beta = 90$. Figures 7e and 7f plot the latency CDF and 50/75/90th percentile latency, respectively. At the 90th percentile, PTS-MEC decreases the RTT latency of MEC flows on the default MEC platform by up to 64.07%. It happens on Flow F1, where PTS-MEC decreases the RTT latency from 101.48 ms and 175.15 ms to 62.22 ms and



(a) Latency CDF of Case III (PTS-MEC) (b) 50/75/90th latency of Case III (PTS-MEC) (c) MEC UE Throughput of Case III

(d) Non-MEC UE Throughput of Case III (e) Latency CDF of Case IV (PTS-MEC) (f) 50/75/90th latency of Case IV (PTS-MEC)

Fig. 7: Improved RTT latency and throughput impact with the PTS-MEC design (Fx represents Flow x).

62.93 ms in the cases of 2 and 3 non-MEC UEs, respectively. It is observed that PTS-MEC can satisfy the latency goal within 3.70% error in the cases of 1 and 2 non-MEC UEs. For the 3 non-MEC UEs, PTS-MEC cannot reach the goal even if the non-MEC UEs are all served with the minimum service rate.

VII. Conclusion

Many MEC platforms are being developed to support low-latency services in cellular networks. Our experimental finding shows that a congested eNB with the BIW method may cause MEC flows to suffer from high latency. The root cause is that the eNB cannot differentiate MEC flows from Internet ones and then prioritize them. We thus propose the PTS-MEC design with prioritized traffic shaping at the MEC. It not only trades off the performance of non-MEC traffic for the latency requirement of MEC flows, but also keeps the eNB from being modified. Our experimental result confirms the effectiveness of PTS-MEC; it can satisfy latency goals within 3.70% error for MEC flows while fairly allocating remaining resource to non-MEC UEs. In the future work, we will consider different latency requirements from MEC flows and another version of the solution built in the eNB that allows to be modified.

References

[1] (2020, Oct.) Cisco annual internet report - Cisco annual internet report (2018–2023) white paper. [Online]. Available: https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html

[2] F. Giust *et al.*, "MEC deployments in 4G and evolution towards 5G," ETSI White Paper No. 24, Feb. 2018.

[3] C.-Y. Li, H.-Y. Liu, P.-H. Huang, H.-T. Chien, G.-H. Tu, P.-Y. Hong, and Y.-D. Lin, "Mobile edge computing platform deployment in 4G LTE networks: A middlebox approach," in *USENIX Workshop on Hot Topics in Edge Computing (HotEdge)*, 2018.

[4] (2020) MEC Middlebox Solution. [Online]. Available: http://nems.cs.nctu.edu.tw/release/

[5] I. Stoica, H. Zhang, and T. S. E. Ng, "A hierarchical fair service curve algorithm for link-sharing, real-time, and priority services," *IEEE/ACM Transactions on Networking*, vol. 8, no. 2, pp. 185–199, 2000.

[6] C.-Y. Li, Y.-D. Lin, Y.-C. Lai, H.-T. Chien, Y.-S. Huang, H. Po-Hao, and H.-Y. Liu, "Transparent AAA security design for low-latency MEC-integrated cellular networks," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 3, pp. 3231–3243, 2020.

[7] C.-Y. Chang, K. Alexandris, N. Nikaein, K. Katsalis, and T. Spyropoulos, "MEC architectural implications for LTE/LTE-A networks," in *ACM Workshop on Mobility in the Evolving Internet Archit. (MobiArch)*, 2016.

[8] A. Huang, N. Nikaein, T. Stenbock, A. Ksentini, and C. Bonnet, "Low latency MEC framework for SDN-based LTE/LTE-A networks," in *IEEE International Conference on Communications (ICC)*, 2017.

[9] S. Huang, Y. Luo, B. Chen, Y. Chung, and J. Chou, "Application-aware traffic redirection: A mobile edge computing implementation toward future 5G networks," in *IEEE 7th International Symposium on Cloud and Service Computing (SC2)*, 2017.

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

[11] M. Chen, Y. Qian, Y. Hao, Y. Li, and J. Song, "Data-driven computing and caching in 5G networks: Architecture and delay analysis," *IEEE Wireless Communications*, vol. 25, no. 1, pp. 70–75, 2018.

[12] M. Siew, D. W. H. Cai, L. Li, and T. Q. S. Quek, "Dynamic pricing for resource-quota sharing in multi-access edge computing," *IEEE Transactions on Network Science and Engineering*, pp. 1–1, 2020.

[13] P. Mach and Z. Becvar, "Mobile edge computing: A survey on architecture and computation offloading," *IEEE Communications Surveys Tutorials*, vol. 19, no. 3, pp. 1628–1656, 2017.

[14] R. Mogi, T. Nakayama, and T. Asaka, "Load balancing method for IoT sensor system using multi-access edge computing," in *Sixth International Symposium on Computing and Networking Workshops (CANDARW)*, 2018.

[15] P. Porambage, J. Okwuibe, M. Liyanage, M. Ylianttila, and T. Taleb, "Survey on multi-access edge computing for internet of things realization," *IEEE Communications Surveys Tutorials*, vol. 20, no. 4, pp. 2961–2991, 2018.

[16] Z. Xu, Y. Zhang, X. Qiao, H. Cao, and L. Yang, "Energy-efficient offloading and resource allocation for multi-access edge computing," in *IEEE International Conference on Consumer Electronics - Taiwan (ICCE-TW)*, 2019.

[17] K. Liu and W. Liao, "Intelligent offloading for multi-access edge computing: A new actor-critic approach," in *IEEE International Conference on Communications (ICC)*, 2020.

[18] GSMA Future Networks, "Cloud AR/VR whitepaper," Apr. 2019.