This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2021.3139026, IEEE Transactions on Vehicular Technology

# Combined Communication and Computing Resource Scheduling in Sliced 5G Multi-access Edge Computing Systems

Winston K.G. Seah
*Engineering and Computer Science*
*Victoria University of Wellington*
Wellington, New Zealand
winston.seah@ecs.vuw.ac.nz

Chung-Hau Lee Ying-Dar Lin
*Computer Science*
*National Chiao-Tung University*
Hsinchu, Taiwan
ydlin@cs.nctu.edu.sg

Yuan-Cheng Lai
*Information Management*
*National Taiwan University of Sci. and Tech.*
Taipei, Taiwan
laiyc@cs.ntust.edu.tw

**The fifth generation (5G) cellular networks aim to deliver data with low/ultra-low latency to users and support diverse services which require two main types of resources: communication and computing resources. These motivated the development of the Multi-access Edge Computing (MEC) paradigm because it can execute tasks' computation closer to users to reduce latency. To satisfy quality of services (QoS) requirements of different applications, both communication and computing resources in 5G MEC systems can be hard partitioned or soft partitioned among multiple network slices where each slice supports one or more services. A typical soft partition approach is using scheduling. Furthermore, packets in the multi-resource system can either be scheduled individually for each resource using a discrete resource scheduling approach or scheduled collectively only once by combined resource scheduling. In this paper, we propose a combined resource scheduler, called Extended Weighted Fair Queueing with Latency Constraint (EWFQ/LC), to schedule packets among network slices subject to system fairness and their latency constraints. By exploiting the virtual finish time feature inherent in weight fair queueing, EWFQ/LC schedules packets according to the fairness when their latency constraints can be met while according to these latency constraints when they are possibly violated. Based on simulations of realistic heterogeneous traffic scenarios, we show that EWFQ/LC reduces latency by as much as 73% compared to scheduling each resource discretely and maintains latency satisfaction ratio above 90%. More importantly, EWFQ/LC is able to accomplish these improvements with greater fairness in resource consumption especially under heavy traffic conditions, i.e., traffic with stricter latency constraints do not excessively over-consume resources of other traffic classes.**

***Index Terms*—5G networks, multi-access edge computing, network slicing, packet scheduling**

## I. Introduction

The explosive increase in mobile phones and laptops has driven the usage of mobile device applications, leading to the drastic growth in traffic volume and their computing demands. Mobile applications such as high resolution video streaming, interactive gaming and augmented reality are emerging and attracting great attention. These applications often require low/ultra-low latency, high network bandwidth and intensive computing resources. Traditional mobile network architectures cannot meet such strict demands due to mainly the long propagation delay. To satisfy these demands, Mobile or Multi-access Edge Computing (MEC) systems have been proposed for the fifth generation (5G) mobile networks [1][2].

5G networks can support diverse services which have different traffic characteristics and quality of services (QoS) requirements, viz., ultra-reliable low-latency communications (URLLC), enhanced Mobile Broadband (eMBB) and massive Machine Type Communications (mMTC) [3]. Mission-critical applications such as vehicle-to-vehicle (V2V) communication has appeared as a typical use case in URLLC. As latency and reliability are the main concerns in such an application, service providers must provide sufficient communication and computing resources to minimize the overall delay between the vehicles. eMBB applications with high-bandwidth demands (e.g., Virtual Reality (VR) and Augmented Reality (AR),) have also pushed mobile computing performance to the next level of real-time responsiveness. As a continuing evolution of 4G Long Term Evolution (LTE) services, eMBB is expected to dominate user demands and network traffic. Apart from applications which require either high communication or high computing resources, 5G is expected to support mMTC-type applications in the Internet of Things (IoT) where each IoT device needs low resources but there are massive number of IoT devices.

In order to meet diverse QoS requirements demanded by different services, network administrators can adopt network slicing techniques, which can be hard partitioning or soft partitioning of resources for different slices [4][5]. The hard-partition approach will divide the network communication and computing resources among network slices to isolate their resource usage. Thus, a physical network architecture is regarded as being divided into several logical networks. Isolated resources guarantee that one slice cannot interfere with another slice on resource utilization, i.e., only the services belonging to a network slice can use dedicated communication and computing resources of that network slice.

In the soft-partition category, a typical approach is using scheduling. Multi-resource scheduling strategies have been proposed that strive to maintain a fair share across the different resources. If we can improve resource utilization, more requests can be processed concurrently, and average task completion time can be reduced. Therefore, besides fairness, maximizing system throughput and minimizing average task completion time are also important performance metrics for multi-resource schedulers. Tasks in MEC come with diverse service requirements because some tasks are communication-intensive while other tasks are computing-intensive. Given such diversity, the goal of this paper is to propose a scheduler that jointly considers communication and computing resources for the sliced MEC system.

The main contributions of this work are summarized as follows: (1) We investigate the importance of joint resource scheduling in MEC systems, which jointly considers both communication and computing requirements of packets. (2) We propose a combined scheduling approach, *Extended Weighted Fair Queueing with Latency Constraint (EWFQ/LC)*, that aims to maximize fairness in resource utilization while satisfying the latency constraints of different service types. When resources are enough, the scheduler operates like Weighted Fair Queueing (WFQ) to jointly allocate both communication and computing resources fairly. However, when some packets' latency constraints cannot be fulfilled, EWFQ/LC determines an optimal packet transmission order that strives to satisfy the latency constraint for each service type. (3) We show that EWFQ/LC can effectively reduce the negative effect of the non-work conserving nature of discrete multi-resource scheduling approaches, and hence provide better QoS.

The rest of this paper is structured as follows. Section II describes the related research in multi-resource scheduling, followed by our problem statement in Section III. In Section IV, we first present our novel combined scheduler, EWFQ/LC, and, for the purpose of comparison, discuss discrete resource scheduling which schedules tasks for each resource discretely. The validation of our approach is presented in Section V, followed by the conclusions in Section VI.

## II. Related Work

Packet or flow scheduling can be regarded as allocating network bandwidth in the time domain. Traditional fair queueing algorithms are designed to schedule packets according to shared bandwidth in a fair manner. However, in the evolution of network appliances or "middleboxes", together with the emergence of software defined networking and network function virtualization, communication resource is not the only shared resource in modern networks. Traffic with diverse QoS requirements are competing for multiple resources [6].

Thus, researchers have proposed fair multi-resource allocation schemes based on Dominant Resource Fairness (DRF) [7]. DRF is further combined with fair queueing to allocate resource which is proven to be fair and strategy-proof [8]. Wang *et al.* [9] proposed a low complexity multi-resource round robin aimed to decrease the time complexity using the elastic round robin (ERR) algorithm [10] while maintaining fairness. Though previous methods such as DRF are able to achieve resource isolation of multiple resources, they failed to maximize system utilization which leads to waste of resource.

In a multiple resource environment, under-utilization may also lead to poor throughput. If we can improve resource utilization, more requests can be processed concurrently, and average task completion time can be reduced. Besides fairness, maximizing system throughput and minimizing average task completion time should also be the performance metrics for multi-resource schedulers. The Tetris multi-resource cluster scheduler [11] packs tasks with similar resource demand and preferentially serves tasks that have less remaining work. Doing so can avoid resource fragmentation and resource over-provisioning, which are the drawbacks of current schedulers. Another approach, Group Multi-Resource Round Robin (GMR$^3$) [12], utilizes a two-level scheduler to minimize average task waiting time while maintaining fairness.

Mobile devices offloading computation tasks to nearby systems with more computing resources can be regarded as early examples of MEC [16]. Computation offloading mechanism in mobile edge cloud that satisfies QoS constraints of traffic flows are studied in [17]. Jointly considering both communication and computing resources in order to achieve optimal MEC resource allocation have also been studied, e.g., [15], [18] and [19]. While multiple resources are considered by the Maximum Task Product (MTP) approach [15], their target environment is a network with multiple access points where the shared bandwidth is a constraint, not an allocatable resource; we focus on a 5G network where the base station allocates radio resources based on users' requests using our proposed algorithm. Another study focuses on power saving [18], which is significantly different from our work. The other is communication technology specific [19] but our approach is physical layer technology agnostic.

Similarly, scheduling of both bandwidth and computing resources have been considered in task offloading in MEC systems [20] and it was noted that resource allocation in MEC should also focus on the optimization of QoS perceived by the different users in terms of the average latency. This work allows trade-off between QoS requirement and energy saving, so there is thus a risk that the QoS requirements of some requests may not be met. Maximizing revenue for MEC service providers through efficient resource allocation has also be studied as there is a cost incurred in providing these resources [13]. Finally, packet scheduling among multiple classes of users with diverse QoS requirements in MEC systems is discussed in [14]. However, this work considers only communication resources.

Depending on the system's goals, the objectives of multi-resource scheduling research include maximizing fairness [7][8][9][12], minimizing time complexity [12] and/or maximizing resource utilization [11]. However, scheduling schemes in these works are designed for a single system (e.g., cloud or middlebox) where multiple resources are located in a place. Thus, they do not consider how to control and schedule the resources that are shared among more than one system where multiple resources are located in different locations (e.g., 5G MEC). Related research on MEC has focused on the impact of both communication and computing resource, such as traffic offloading [17][20][21] and MEC resource allocation [13][18]; the packet scheduling problem in MEC systems has also been addressed [14]. Though all these works emphasize the impact of multiple resources, they have not specifically addressed the latency constraint which is critical to the 5G cellular network.

Table I summarizes and compares the related research to our proposed scheduler; not included in the table are research that focus solely on offloading jobs to MEC, rather than MEC scheduling [17][18][20][21], as well as those on communication-specific technology [19].

TABLE I
COMPARISON OF MULTI-RESOURCE SCHEDULING APPROACHES

| Papers | Algorithm | Scenario | Objective | Resources |
|---|---|---|---|---|
| Dominant Resource Fairness (DRF) [7] | Generalised Max-Min Fairness | Cloud (datacenters) | maximize fairness | computation, memory |
| Dominant Resource Fair Queueing (DRFQ) [8] | Generalised Virtual Time (Fair Queueing) | Middlebox (routers) | maximize fairness | computation, egress* bandwidth |
| Multi-Resource Round Robin (MR$^3$) [9] | Elastic Round Robin [10] | Middlebox (routers, firewalls, etc.) | maximize fairness; minimize complexity | computation, egress bandwidth |
| Tetris - multi-resource cluster scheduler [11] | Multi-Resource Packing | Cloud | maximize throughput; minimize task completion time | computation, memory, disk, bandwidth, etc. |
| Group Multi-Resource Round Robin (GMR$^3$) [12] | Multi-Resource Fair Queueing | Middlebox | maximize fairness; minimize complexity; bounded delay | computation, bandwidth |
| Resource Modeling and Scheduling for MEC [13] | Queueing network based Optimal Resource Deployment | MEC | maximise revenue from services; minimize total capital expenditure | computation, uplink & downlink bandwidth |
| Downlink Slicing for Softwarized MEC [14] | Hierarchical Fair Service Curve (HFSC) | MEC | network slice isolation for low latency applications | downlink bandwidth |
| Fair Multi-Resource Allocation in MEC [15] | Maximum Task Product (MTP) | MEC with multiple APs | maximize fairness given shared bandwidth constraints | computation, memory |
| **This paper** | Extended Weighted Fair Queueing with Latency Constraint | MEC | maximize fairness; maximize latency satisfactory ratio | computation, uplink & downlink bandwidth |

*egress – output link of a middlebox (e.g., router) that is similar to the downlink of a MEC system.
fairness – each network slice gets at least its guaranteed resource capacity specified in the service-level agreement.
latency satisfaction ratio – ratio of the number of packets meeting their latency constraints over the number of total packets.

## III. SYSTEM MODEL AND PROBLEM FORMULATION

In this section, we first provide an overview of the MEC system architecture. We then formulate the scheduling problem which considers communication and computing resources together, and give a concise description of the WFQ scheduling algorithm that forms the basis of our approach.

### A. System architecture

The architecture of the MEC system consists of base stations and MEC server as shown in Figure 1. The base station (BS) serves as the radio access technology which connects the user equipment and the MEC server. For an uplink resource request, a UE sends a scheduling request to the base station via the physical uplink control channel (PUCCH) prior to offloading its task to the MEC. If the BS accepts this offloading request, it replies to the UE via the physical downlink control channel (PDCCH). At a rate of once every transmission time interval (TTI), the medium access control (MAC) scheduler distributes available 5G New Radio resource blocks (RBs) to the UEs awaiting for uplink resource. By focusing on RBs, we abstract the physical layer technology, making our approach technology agnostic. The task is transmitted to the BS via the physical uplink shared channel (PUSCH) then forwarded to the MEC server. The MEC server is responsible for processing the tasks carried in the packets.

### B. Problem Description

A MEC system is equipped with communication capacity, viz., uplink $C^U$ and downlink $C^D$, and computing capacity $C^P$. The unit of the communication resource is the RB. Computing capacity, $C^P$, is in the unit of instructions per second. The $j^{th}$ packet of the $i^{th}$ network slice, denoted as $P_i^j$, has the packet length $\mu_i^C$, the number of required processing instructions $\mu_i^P$, and the latency constraint $LC_i^j$. Note herein, for simplicity, we assume the packet length and the number of required processing instructions of all packets belonging to a slice are the same. Therefore, $\mu_i^C$ and $\mu_i^P$ do not have the index $j$.

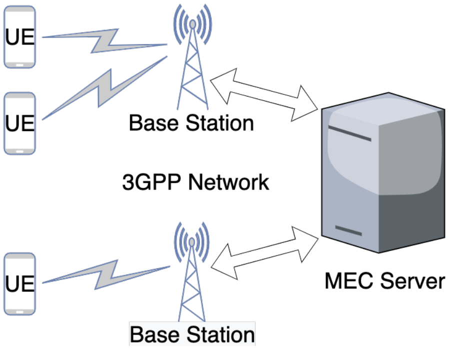


Fig. 1. System architecture

Given $C^U$, $C^D$ and $C^P$ of the MEC system, we shall determine the order in which packets of various independent network slices are forwarded to maximize fairness, of each service, while subject to their latency constraints, i.e. $LC_i^j$.

Scheduling in 4G/5G has attracted lots of attention over the years. While current schedulers in 4G/5G tend to focus on communication resource only [22][23][24], we focus on scheduling packets by considering both communication and computing resources based on the requirements of service types in 5G in order to satisfy its QoS constraints in a sliced mobile network. We leverage WFQ to virtually achieve resource isolation among different slices. At the same time, we extend weighted fair queueing with consideration of the latency constraint by exploiting its virtual finish time concept. Our approach maximizes resource utilization and minimizes average task completion time simultaneously while maintaining fairness subject to satisfying latency constraints.

### C. Weighted Fair Queueing (WFQ) Basics

WFQ is a packet-based variant of generalized processor sharing [25] which partitions system resources and then shares the total system resources among multiple queues. If all queues are backlogged, system resources are assigned to queues in proportion to their weights, However, due to the burstiness of traffic, not all users may have requests in a certain interval. The work-conserving characteristic of WFQ ensures that unused system resources at a scheduling interval are then distributed among all other backlogged flows according to their weights. Hence, the resources are shared among flows in a fair and efficient way, and overall system throughput is increased as a result. WFQ is an excellent solution in a scenario where we want to provide users with a bounded delay while maintaining fairness given a shared resource. WFQ is also a means to prioritise the flows without causing starvation.

WFQ is built on the concept of virtual time. The virtual finish time of $P_i^j$ arriving in the system, denoted by $F(P_i^j)$, is calculated as [25]

$$F(P_i^j) = S(P_i^j) + \frac{LEN(P_i^j)}{C_i} \tag{1}$$

where $LEN(P_i^j)$ is the length of the packet $P_i^j$ and $C_i$ is the proportion of the total link capacity $C$ allocated to $i^{th}$ service type based on its weight, $w_i$; $C_i$ is computed as

$$C_i = \frac{w_i}{\sum_{\forall i} w_i} C.$$

Its virtual start time, $S(P_i^j)$, is given by [25]

$$S(P_i^j) = \max\left\{A(P_i^j), F(P_i^{j-1})\right\} \tag{2}$$

where $A(P_i^j)$ is the arrival time of $P_i^j$ in the system. For a backlogged flow, the virtual start time of $P_i^j$ will be the virtual finish time of the previous packet in the flow, i.e., $F(P_i^{j-1})$.

Furthermore, 5G traffic types like URLLC have stringent latency constraints that must be met. To satisfy the latency constraint of each packet, we assign every service type with the appropriate share of communication and computing resources according to the amount of the resources needed by each service type and its latency constraint. Based on those information, we determine the packet transmission order among flows based on each packet's computed virtual finish time.

## IV. MEC Multi-Resource Scheduling

In MEC systems, each 5G service type can be viewed as an individual network slice that consumes both the communication and computing resources. It is common to support different service types concurrently and to differentiate offloaded tasks belonging to individual service types, each network slice is assumed to be a first-in-first-out (FIFO) queue. Multiple queues represent multiple service types that exist in both communication and computing resource schedulers. Each offloaded task is modelled as a packet which requires both communication and computing resources and it has to wait in the queue if any required resource is not available at the moment.

We propose a new scheduling algorithm that extends the WFQ scheduling algorithm to allocate both communication and computing resources simultaneously, that also accounts for the latency constraints of the different slices. The algorithm aims to satisfy the QoS requirements of each request while maintaining fairness, and most importantly, ensures that latency constraints are met. We refer to our proposed scheduler as *Extended Weighted Fair Queueing with Latency Constraint (EWFQ/LC)* scheduling. For comparison, we also discuss WFQ scheduling of requests discretely/separately, i.e., uplink, computing and downlink, in such a way as to meet the QoS requirements of each slice while maintaining fairness.

### *A. Combined Resource Scheduling*

Our proposed MEC system includes a combined communication and computing resource scheduler, multiple uplink queues, and FIFO queues for the downlink and computing server, as shown in Figure 2. FIFO queues are appropriate for the downlink and computing server as the requests are already scheduled in optimal order to maximize both communication and computing resources. The number of uplink queues is equal to the number of slices in the MEC system. The combined communication and computing resource scheduler resides in the MAC layer of the 5G gNodeB protocol stack where the allocation of RBs to the UEs is done. For UEs offloading computing tasks to the MEC server, UEs send a scheduling request to the MEC server beforehand through the PUCCH. The scheduling request carries Uplink Channel Information (UCI) which includes information, such as, the service type of the packet as well as Channel Quality Indicator (CQI) reports. The combined communication and computing resource scheduler decides the packet order among multiple uplink queues according to packets' required communication and computing resources. Finally, the scheduler sends Downlink Channel Information (DCI) through the PDCCH that carries the scheduling decisions to the UE.

*a) Extended Weighted Fair Queueing (EWFQ):* The EWFQ algorithm is designed to make scheduling decisions of tasks offloaded by MEC users among multiple service types by considering both communication and computing resources. The virtual finish time of $P_i^j$ is then calculated as

$$F(P_i^j) = S(P_i^j) + T_{ij}^U + T_{ij}^D + T_{ij}^P. \tag{3}$$

$T_{ij}^U$ is the virtual uplink service time of the packet $P_i^j$, which can be calculated as

$$T_{ij}^U = \frac{\mu_i^C}{\phi_i^C * C^U} \tag{4}$$

where $\mu_i^C$ is its packet length, $\phi_i^C$ is the weight of uplink resource for the $i^{th}$ slice, and $C^U$ is the uplink resource capacity. $T_{ij}^D$ is the virtual downlink service time of the packet $P_i^j$, which can calculated as

$$T_{ij}^D = \frac{\mu_i^C}{\phi_i^C * C^D} \tag{5}$$

where $C^D$ is the downlink resource capacity. Last, but not least, $T_{ij}^P$ is the virtual service time of the packet $P_i^j$ at the computing server, which can be calculated as

$$T_{ij}^P = \frac{\mu_i^P}{\phi_i^P * C^P} \tag{6}$$

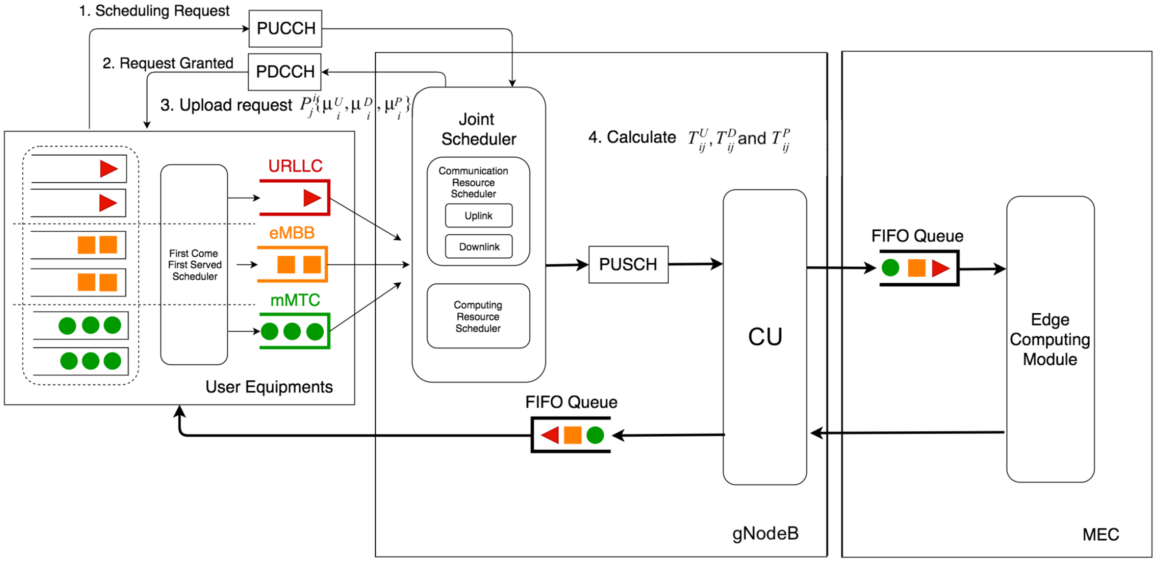

1. Scheduling Request
PUCCH
2. Request Granted
PDCCH
3. Upload request $P_j^i\{\mu_i^U, \mu_i^D, \mu_i^P\}$
Joint Scheduler
Communication Resource Scheduler
Uplink
Downlink
Computing Resource Scheduler
4. Calculate $T_{ij}^U, T_{ij}^D$ and $T_{ij}^P$
URLLC
eMBB
mMTC
First Come First Served Scheduler
User Equipments
PUSCH
CU
FIFO Queue
FIFO Queue
Edge Computing Module
gNodeB
MEC

Fig. 2. Combined resource scheduling.

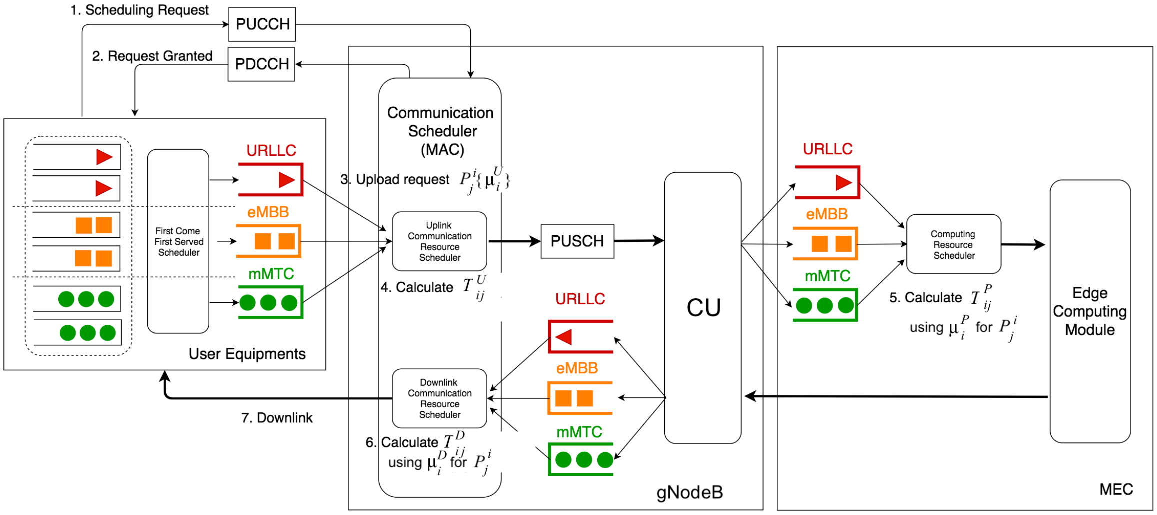

1. Scheduling Request
PUCCH
2. Request Granted
PDCCH
Communication Scheduler (MAC)
3. Upload request $P_j^i\{\mu_i^U\}$
URLLC
eMBB
mMTC
First Come First Served Scheduler
User Equipments
Uplink Communication Resource Scheduler
4. Calculate $T_{ij}^U$
PUSCH
CU
URLLC
eMBB
mMTC
Downlink Communication Resource Scheduler
6. Calculate $T_{ij}^D$ using $\mu_i^D$ for $P_j^i$
7. Downlink
URLLC
eMBB
mMTC
Computing Resource Scheduler
5. Calculate $T_{ij}^P$ using $\mu_i^P$ for $P_j^i$
Edge Computing Module
gNodeB
MEC

Fig. 3. Discrete resource scheduling.

where $\mu_i^P$ is its required computing resource, $\phi_i^P$ is the weight of the computing resource for the $i^{th}$ slice, and $C^P$ is the computing resource capacity.

Packets are then scheduled in an increasing order of their virtual finish times.

*b) Extended Weighted Fair Queueing with Latency Constraint (EWFQ/LC):* To limit the violation of the latency constraint, the MEC server should not only schedule a request according to its virtual finish time but also satisfy the latency constraint, $LC_i^j$, of each $P_i^j$ to maximize the probability, if not guarantee, that each packet can be processed before its deadline. For example, for connections with the stringent QoS requirement, its virtual finish time calculated using Eqn. (3) may exceed its $LC_i^j$ value. In contrast, a connection which has loose latency constraint can be delayed until its latency constraint so that the resource can be released to serve those with tight latency constraints. Thus, if the calculated virtual finish time exceeds its latency constraint, then the packet's virtual finish time is based on the latency constraint. This effectively switches the scheduling of affected packets from fairness to latency dependent. In this case, the virtual finish time of $P_i^j$ is then changed to

$$F(P_i^j) = S(P_i^j) + \min(LC_i^j, T_{ij}^{Total}) \tag{7}$$

where $T_{ij}^{Total} = T_{ij}^U + T_{ij}^D + T_{ij}^P$.

To recap, by exploiting the use of virtual finish time that is an integral component of the WFQ algorithm, we are able to switch seamlessly between scheduling according to fairness and scheduling according to latency by comparing a request's computed virtual finish time with its latency constraint. If the computed virtual finish time is earlier than the latency constraint, the request will be scheduled according to fairness, based on the computed virtual finish time, as per the original WFQ design. On the other hand, if its computed virtual finish time violates its latency constraint, it will be scheduled based on latency, as shown in Eqn. (7).

### B. Discrete Resource Scheduling

In the traditional approach of scheduling resources separately, which we refer to as the *discrete resource scheduling approach*, multiple queues exist separately in the uplink MAC scheduler, downlink MAC scheduler and the computing resource scheduler. Packets exiting the base station are then classified into queues in the computing server according to slices, as shown in Figure 3. Each scheduler in this model performs resource scheduling discretely without the global view of total system resources.

Scheduling of each packet can be divided into three parts. The first part is the uplink scheduler in the base station. The virtual finish time of $P_i^j$ in the uplink queue is calculated as

$$F^U(P_i^j) = S(P_i^j) + T_{ij}^U. \tag{8}$$

The second part is the computing scheduler in the MEC system. The packet arrives in the computing server after exiting the uplink at virtual time $F^U(P_i^j)$. Therefore, the virtual finish time of the packet in the computing queue is calculated as

$$F^P(P_i^j) = \max\left\{F^U(P_i^j), F^P(P_i^{j-1})\right\} + T_{ij}^P. \tag{9}$$

Finally, the last part is the downlink scheduler in the base station. The virtual finish time of the packet in the downlink queue is calculated as

$$F^D(P_i^j) = \max\left\{F^P(P_i^j), F^D(P_i^{j-1})\right\} + T_{ij}^D. \tag{10}$$

In each scheduler, packets are scheduled based on virtual finish time. Once the amount of requested resources exceed the communication or computing capability, a network slice which has a relative tight latency constraint may fail to satisfy its QoS. To fulfill the packets' latency constraints, these packets can be scheduled based on their latency constraints only instead of considering the virtual finish time obtained above. However, the overall virtual finish time of a packet in a MEC server cannot be pre-computed when discrete resource scheduling is adopted as the virtual finish time of a packet at each server can only be computed upon the packet's arrival at the particular server.

As a result of the limitation, the virtual finish time of the incoming packet, $P_i^j$ in the downlink scheduler is calculated as

$$F^D(P_i^j) = S(P_i^j) + LC_i^j. \tag{11}$$

The discrete scheduling using Eqn. (8), Eqn. (9), and Eqn. (10), is denoted as WFQ in this paper because each scheduler adopts WFQ. On the other hand, the discrete scheduling using Eqn. (8), Eqn. (9), and Eqn. (11) is denoted as WFQ/LC because the downlink scheduler considers the latency constraint, rather than using WFQ.

## V. Simulations

In this section, we first describe our simulation setup and scenarios used to validate the performance of the proposed scheduling framework. A detailed analysis based on end-to-end latency, satisfaction ratio of the latency constraint and fairness index is presented, as well as how resources are consumed by the different service types.

TABLE II
Key System Simulation Parameters

| | Parameter | Setting |
|---|---|---|
| Overall | Number of service types | 3 |
| | Number of MECs | 1 |
| | Traffic arrival distribution | Poisson |
| | Simulation time | 60 seconds |
| MEC | Uplink bandwidth | 25 resource blocks |
| | Downlink bandwidth | 25 resource blocks |
| | Computing capacity | 10,000 MIPS |

### A. Simulation Setup

This work is implemented using the ns-3 simulator (version 3.26) with LTE module. ns-3's LTE module provides us with the functionality for algorithm design and performance evaluation of downlink and uplink schedulers, as well as Radio Resource Management algorithms. As the current MEC model is not supported yet, we simulate a MEC server by combining the BS with a computing node and the propagation delay between them is set as zero.

To ensure that packet scheduling is done across network slices, we implemented our scheduling algorithm in the MAC layer of the LTE protocol stack provided by ns-3, where other scheduling algorithms are implemented, e.g., Round Robin (RR), Proportional Fair (PF), Priority Set Scheduler, etc. (https://www.nsnam.org/docs/models/html/lte-design.html#mac.) A computing server application is installed in the MEC to perform scheduling decisions as well as simulate the computation task scheduling. UEs are positioned near the BS and non-mobile, as the purpose of the validation is to assess the efficiency of our combined resource scheduling algorithm. The performance of our approach is measured using Flow Monitor, a network monitoring framework for ns-3 (https://www.nsnam.org/docs/models/html/flow-monitor.html), which measures well known network performance metrics by probing the network nodes to track packets exchanged by the nodes. The key system simulation parameters, which are adopted from a similar network scenario [24], are listed in Table II.

For functional testing and debugging, we first considered a homogenous traffic scenario where packet size, consumption of both communication and computing resources, and the resource allocation weights are the same between slices, as shown in Table III. In this test scenario, our key purpose is to assess the relationship between the resource share and the resource demand, with the latency constraint being the key differentiating parameter.

To realistically simulate traffic of different service types in the 5G mobile network, traffic from UEs are classified into URLLC which has the most stringent latency constraint, eMBB which has normal latency constraint and mMTC which has the loosest latency constraint. The traffic type characteristics and resource requirements for this heterogeneous traffic scenario are summarized in Table IV.

### B. Simulation Results

The performance of our proposed combined scheduler is assessed based on the following metrics: latency, satisfaction ratio and fairness. For each set of parameters, multiple runs

TABLE III
UE Traffic Types and Homogeneous Resource Requirements

| Requirements | mMTC | eMBB | URLLC |
|---|---|---|---|
| Maximum Packet Size (byte) | 1024 Bytes | | |
| Communication resource weight | 1/3 | 1/3 | 1/3 |
| Required Instructions(MIPS) | 2000 | 2000 | 2000 |
| Computing resource weight | 1/3 | 1/3 | 1/3 |
| Latency Constraint (ms) | 100 [26] | 50 [27] | 20 [28] |

TABLE IV
UE Traffic Types and Heterogeneous Resource Requirements

| Requirements | mMTC | eMBB | URLLC |
|---|---|---|---|
| Maximum Packet Size (byte) | 40 Bytes | 1100 Bytes | 40 Bytes |
| Communication resource weight | 1/5 | 3/5 | 1/5 |
| Required Instructions(MIPS) | 200 | 400 | 700 |
| Computing resource weight | 1/4 | 1/4 | 1/2 |
| Latency Constraint (ms) | 100 [26] | 50 [27] | 20 [28] |

were executed, results averaged and checked to ensure that we achieved a stable representation of the performance [29].

The latency metric measures the total time required for an arbitrary packet to be sent from the UE to the network, processed at the MEC server and outcome returned to the UE via the downlink. This includes the time spent waiting in the queues of the UE, the MEC server and downlink of the BS.

*Satisfaction ratio* measures the percentage of packets whose overall latency fall within the latency constraint of their service type. We first analyze the latency performance of our proposed combined resource scheduling compared to discrete resource scheduling. Then, we assess the ability of our joint resource scheduler in satisfying the latency constraints and the difference in resource consumption ratio among different service types with different latency constraints (cf: Table IV).

To evaluate fairness of the proposed scheduling algorithm, we extended Jain's fairness index to calculate the overall system fairness. Jain's fairness index [30] measures fairness by looking at each flow's ratio of its bandwidth usage and its guaranteed bandwidth in a system. We selected Jain's fairness index as it is the most widely use quantitative metric that provides a real number representation which can be used for performance comparison, unlike qualitative measures such as max-min and proportional fairness [31].

Typically, fairness is measured by assuming each flow competes for its fair share in a backlogged link. However, in the WFQ-based scheduler, not all flows are able to fully utilize its resource, so the remaining resource of a certain flow will be distributed to the flows which demand more resource. In this case, the equal rate fairness concept inherent in the original Jain's fairness index fail to reflect the correct system fairness. We solve this limitation of fairness measurement by taking the demand of each flow into account. Then, the extended Jain's fairness index $F$, which indicates the overall system fairness, is calculated as [30]

$$F = \frac{(\sum_{i=1}^{n} G_i)^2}{n * (\sum_{i=1}^{n} G_i^2)} \tag{12}$$

where the resource gain ratio of the $i^{th}$ slice, denoted as $G_i$, is defined as

$$G_i = \frac{Gained\ throughput_i}{Min(Guaranteed\ throughput_i, Demand\ throughput_i)}$$

and $G_i$ ranges between 0 and 1.

### 1) Combined vs Discrete Resource Scheduling

Our proposed combined resource scheduler EWFQ/LC is able to perform scheduling with the global view of the entire system. It improves latency performance by alleviating the negative effect of non-work-conserving phenomenon induced in multi-resource scheduling where the computing resource is idle while waiting for tasks to be sent from UEs to the MEC server via the BS.

Under homogeneous traffic conditions, Figure 4(a) shows that combined resource scheduling using EWFQ/LC can improve end-to-end latency up to 10% compared to discrete resource scheduling using WFQ/LC. In the case of heterogeneous traffic, the improvement is even more significant, as shown in Figure 4(b). The combined resource scheduling using EWFQ/LC can improve end-to-end latency by 36% up to 73% compared to discrete resource scheduling using WFQ/LC especially at traffic arrival rates of more than 600 packets/second. At higher rates, while striving to satisfy the latency constraints, it is expected that eMBB traffic experiences longer average latency than mMTC because of the larger packet size as compared to both URLLC and mMTC, and also longer computation time than mMTC.

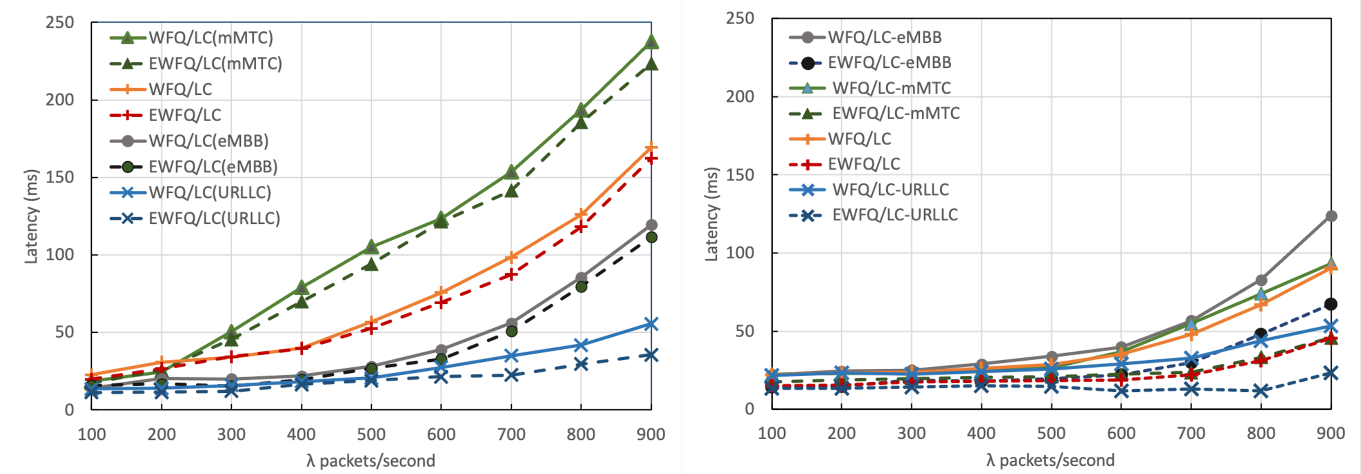
With regard to the ability of EWFQ/LC in satisfying latency constraints, Figures 5(a) and 5(b) indicate that EWFQ/LC is able to keep the overall system's latency satisfaction ratio above 77% at a loss of 27% in fairness under the homogenous traffic scenario. (Note: WFQ without any latency constraints achieves 100% fairness.) In contrast, the WFQ/LC can only keep the overall system's latency satisfaction ratio above 48% but sacrifices up to 60% in fairness. Figures 6(a) and 6(b) show that EWFQ/LC continues to perform well in maintaining overall latency satisfaction ratio above 90% with an 11% loss in fairness. This performance still exceeds that of WFQ/LC, though less significant than in the homogeneous traffic scenario, but EWFQ/LC has much better overall latency performance than WFQ/LC as previously noted above.

A shortcoming of the discrete scheduling approach is that each scheduling decision can only be made when task requests are presented to the respective schedulers. A new task request arriving at the UE is first scheduled based on available bandwidth resources without knowledge of whether the MEC server has resources to accommodate its computing requirements. Computing resources can only be scheduled after the task has been sent to the BS and presented to the MEC server. Conversely, the computing server may be left idle while long packets are being sent from the UE. Had smaller packets, albeit of lower priority traffic, been transmitted first, the computing server can be used to process them while waiting for the larger packets to arrive. Our EWFQ approach mitigates this limitation by calculating the overall packet virtual finish time, with the consideration of both communication and computing resources, to determine a packet transmission order at the point of entry into the network.

### 2) Resource Consumption Ratio among Service Types

In order to satisfy the latency constraints, it is inevitable that traffic with tighter latency constraints, e.g., URLLC, consumes more resources than their allocated share. We define Resource

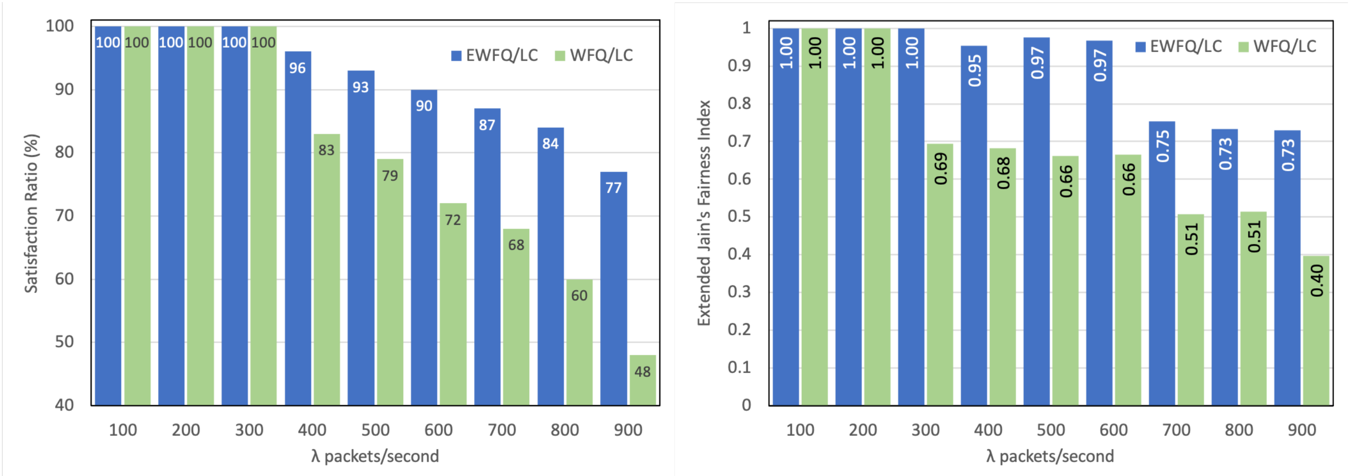
This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2021.3139026, IEEE Transactions on Vehicular Technology



(a) Homogeneous Traffic

(b) Heterogeneous Traffic

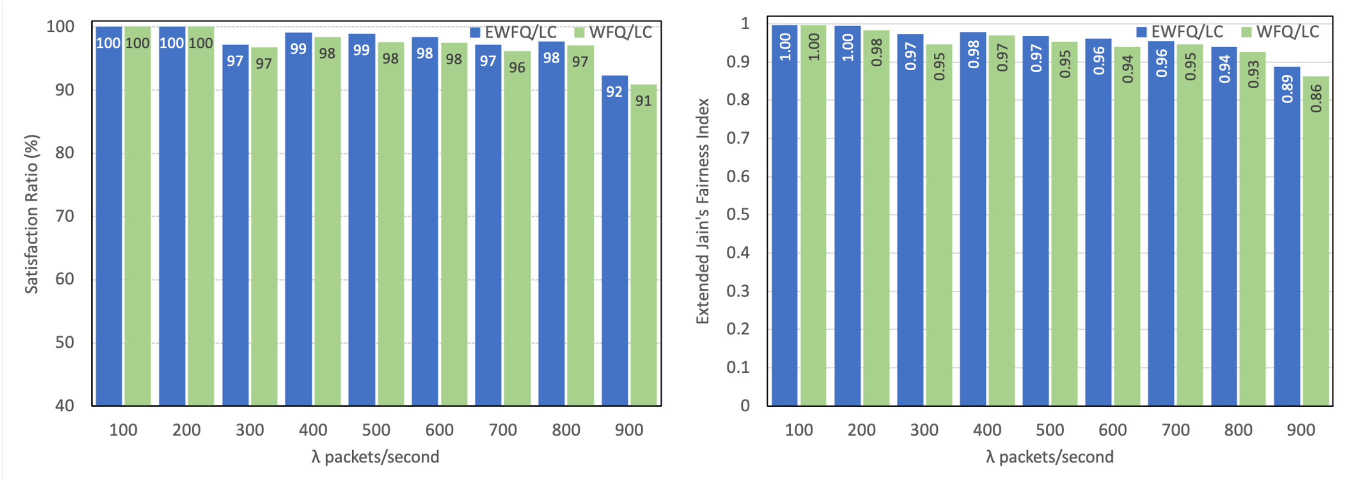
Fig. 4. Latency performance of combined vs. discrete resource scheduling



(a) Satisfaction Ratio

(b) Fairness

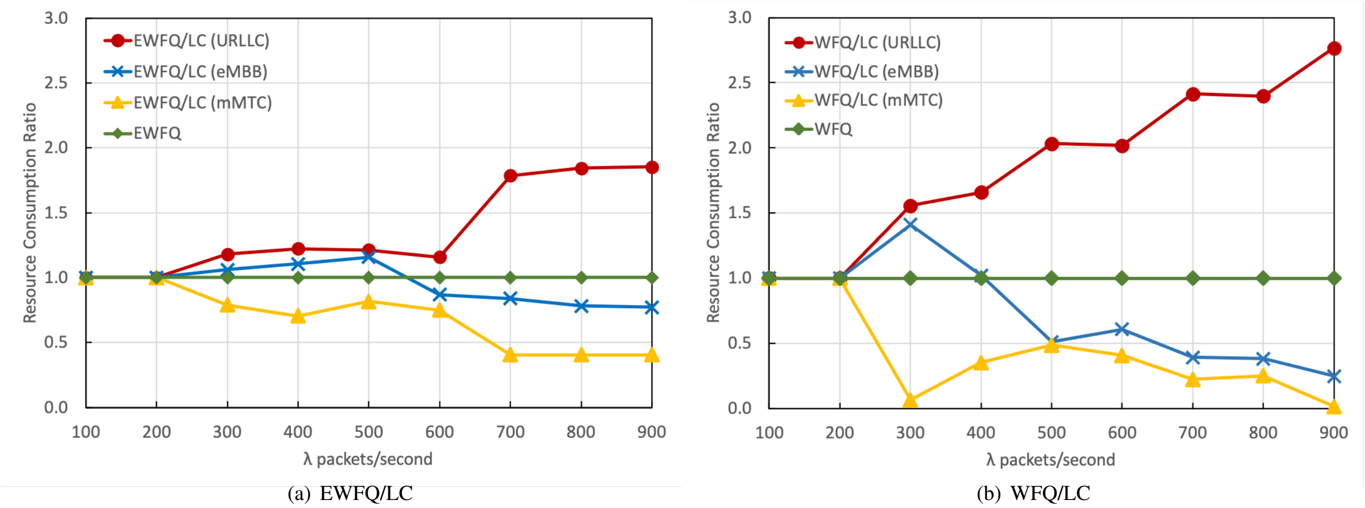
Fig. 5. Satisfaction Ratio & Fairness in Homogeneous Traffic



(a) Satisfaction Ratio

(b) Fairness

Fig. 6. Satisfaction Ratio & Fairness in Heterogeneous Traffic



(a) EWFQ/LC

(b) WFQ/LC

Fig. 7. Resource Consumption Ratio among traffic types under heterogeneous traffic scenario

Consumption Ratio ($RCR_i$) of a service type $i$ as the ratio of the actual amount of resources consumed by type $i$ traffic against the resources allocated to type $i$ traffic, computed as follows:

$$RCR_i = \frac{Amount\ of\ resources\ consumed\ by\ type\ i}{Resources\ allocated\ to\ type\ i} \tag{13}$$

where $i$ = {URLLC, eMBB, mMTC}. Figures 7(a) and 7(b) show resource consumption ratios achieved by EWFQ/LC and WFQ/LC, respectively, for the URLLC, eMBB and mMTC services under heterogeneous traffic; a value of 1.0 shows that the amount of resources consumed is the amount that has been allocated.

When there are sufficient resources at low traffic loads of up to 200 packets/second, EWQF/LC ensures fair sharing of resources like WFQ. However, as the traffic load increases and resources become scarcer, packets with relatively looser latency constraints yield their resources to those with tighter latency constraints. This unfair resource consumption must be minimized. URLLC and eMBB do not over-consume resources at the expense of mMTC under EWQF/LC despite increasing traffic loads up to 600 packets/second, beyond which URLLC traffic has priority over resources though not excessively.

On the other hand, WFQ/LC is no longer able to ensure fair resource consumption once traffic arrival rate exceeds 200 packets/second. mMTC traffic suffers most significantly but manages to reclaim some share of the resources from eMBB traffic at 500 packets/second load. However, as traffic load further increases, both eMBB and mMTC traffic collectively yield resources to URLLC traffic.

The results show that URLLC service type over-consumes less resources under EWFQ/LC as compared to WFQ/LC, while eMBB service type gives up less resources in using EWFQ/LC, as compared to using WFQ/LC, yet is able to satisfy its latency constraints. Thus, using EWFQ/LC, eMBB and mMTC are able to keep their resource consumption ratios above 77% and 40% respectively.

## VI. Conclusions

This paper proposed the Extended Weighted Fair Queueing with Latency Constraint (EWFQ/LC) scheduling, which jointly considers communication and computing resources to perform one-time scheduling in sliced 5G MEC systems. This scheduling approach fits well into the distributed Evolved Packet Core deployment option proposed by ETSI [32]. EWFQ/LC aims to optimize resource fairness among service types in 5G while simultaneously satisfying their QoS requirements. EWFQ/LC first examines the relationship between the required resources of an incoming request and the resource capacity of its services. Then, it calculates the request's finish time. If the request will violate its latency constraint, EWFQ/LC adjusts the request transmission order to find an optimal solution to satisfy the prescribed service latency requirement.

The simulation results show that our EWFQ/LC scheduler for sliced MEC system is able to decrease the overall end-to-end latency of the incoming requests. When subjected to latency constraints, EWFQ/LC ensures that URLLC traffic with a stringent latency constraint of $20ms$ has the lowest average end-to-end latency while mMTC (latency constraint of $100ms$) has the highest average end-to-end latency. The analysis on resource consumption ratio shows that more resources are needed for the network slice with a tighter latency constraint. The traffic with a loose latency constraint yields resources to the traffic with a tighter latency constraint.

In the discrete resource scheduling approach, the lack of global view of resources leads to the occurrence of the non-work-conserving phenomenon, as noted in Section IV-B. URLLC traffic consumes much more resources compared to the combined resource scheduling to meet its end-to-end latency constraint. The discrete resource scheduling leaves little resource to other service types which leads to low satisfaction of latency constraint as the amount of traffic spikes up. Although both approaches perform comparably in the latency satisfaction ratio under heterogeneous traffic, the better resource consumption ratio of EWFQ/LC will contribute to better performance as traffic intensifies.

Our future work includes the following. Firstly, a rigorous mathematical analysis needs to be done to determine the performance bounds. From the design perspective, we plan to add an admission control mechanism in order to satisfy latency constraint when the incoming traffic exceeds resource capacity for a sustained period. Lastly, we will study an adaptive weight determination method to dynamically assign weights to the slices in order to make EWFQ/LC more reliable. With the envisaged increase in the number and heterogeneity of end devices, together with diverse traffic types, the reliance on machine learning techniques to solve resource allocation challenges in MEC is inevitable [33].

## References

[1] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A Survey on Mobile Edge Computing: The Communication Perspective," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2322–2358, 2017.

[2] A. Filali, A. Abouaomar, S. Cherkaoui, A. Kobbane, and M. Guizani, "Multi-Access Edge Computing: A Survey," *IEEE Access*, vol. 8, 2020.

[3] 3GPP, "Study on Scenarios and Requirements for Next Generation Access Technologies," ETSI, Technical Report TR 38.913 version 14.2.0 Release 14, 2017.

[4] L. U. Khan, I. Yaqoob, N. H. Tran, Z. Han, and C. S. Hong, "Network Slicing: Recent Advances, Taxonomy, Requirements, and Open Research Challenges," *IEEE Access*, vol. 8, pp. 36 009–36 028, 2020.

[5] H. Basilier, J. Lemark, A. Centonza, and T. Åsberg, "Applied network slicing scenarios in 5G," *Ericsson Technology Review*, 2021.

[6] Q. Ye, J. Li, K. Qu, W. Zhuang, X. S. Shen, and X. Li, "End-to-End Quality of Service in 5G Networks: Examining the Effectiveness of a Network Slicing Framework," *IEEE Vehicular Technology Magazine*, vol. 13, no. 2, pp. 65–74, 2018.

[7] A. Ghodsi *et al.*, "Dominant Resource Fairness: Fair Allocation of Multiple Resource Types," in *Proc. of the 8th USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, Boston, MA, USA, 2011.

[8] A. Ghodsi, V. Sekar, M. Zaharia, and I. Stoica, "Multi-Resource Fair Queueing for Packet Processing," in *Proc. of the ACM SIGCOMM*, Helsinki, Finland, 2012.

[9] W. Wang, B. Li, and B. Liang, "Multi-Resource Round Robin: A Low Complexity Packet Scheduler with Dominant Resource Fairness," in *Proc. of the 21st IEEE International Conference on Network Protocols (ICNP)*, Goettingen, Germany, 2013, pp. 1–10.

[10] S. S. Kanhere, H. Sethu, and A. B. Parekh, "Fair and Efficient Packet Scheduling Using Elastic Round Robin," *IEEE Transactions on Parallel and Distributed Systems*, vol. 13, no. 3, pp. 324–336, 2002.

[11] R. Grandl, G. Ananthanarayanan, S. Kandula, S. Rao, and A. Akella, "Multi-Resource Packing for Cluster Schedulers," *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 4, 2014.

[12] W. Wang, B. Liang, and B. Li, "Low Complexity Multi-Resource Fair Queueing with Bounded Delay," in *Proceeding of IEEE INFOCOM*, Toronto, ON, Canada, 2014, pp. 1914–1922, ISBN: 978-1-4799-3360-0.

[13] S. Guo, D. Wu, H. Zhang, and D. Yuan, "Resource Modeling and Scheduling for Mobile Edge Computing: A Service Provider's Perspective," *IEEE Access*, vol. 6, 2018.

[14] K. Amemiya, Y. Akiyama, K. Kobayashi, Y. Inoue, S. Yamamoto, and A. Nakao, "On-Site Evaluation of a Software Cellular Based MEC System with Downlink Slicing Technology," in *Proc. of the IEEE 7th International Conference on Cloud Networking (CloudNet)*, Tokyo, Japan, 2018, pp. 1–7.

[15] E. Meskar and B. Liang, "Fair Multi-Resource Allocation in Mobile Edge Computing with Multiple Access Points," in *Proc. of the 21st International Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing (MobiHoc)*, Boston, MA, USA, 2020, pp. 11–20.

[16] B. Wang, C. Wang, W. Huang, Y. Song, and X. Qin, "A Survey and Taxonomy on Task Offloading for Edge-Cloud Computing," *IEEE Access*, vol. 8, 2020.

[17] X. Chen, W. Li, S. Lu, Z. Zhou, and X. Fu, "Efficient Resource Allocation for On-Demand Mobile-Edge Cloud Computing," in *IEEE Transactions on Vehicular Technology*, vol. 67, 2018, pp. 8769–8780.

[18] Y. Yu, J. Zhang, and K. B. Letaief, "Joint Subcarrier and CPU Time Allocation for Mobile Edge Computing," in *Proc. of the IEEE Global Communications Conference (GLOBECOM)*, Washington, DC, USA, 2016, pp. 1–6.

[19] C. Zhao, Y. Cai, A. Liu, M. Zhao, and L. Hanzo, "Mobile Edge Computing Meets mmWave Communications: Joint Beamforming and Resource Allocation for System Delay Minimization," *IEEE Transactions on Wireless Communications*, vol. 19, no. 4, pp. 2382–2396, 2020.

[20] M. Molina, O. Muñoz, A. Pascual-Iserte, and J. Vidal, "Joint Scheduling of Communication and Computation Resources in Multiuser Wireless Application Offloading," in *Proc. of the 25th IEEE Annual International Symposium on Personal, Indoor, and Mobile Radio Communication (PIMRC)*, Washington, DC, USA, 2014, pp. 1093–1098.

[21] T. K. Rodrigues, J. Liu, and N. Kato, "Offloading Decision for Mobile Multi-Access Edge Computing in a Multi-Tiered 6G Network," *IEEE Transactions on Emerging Topics in Computing*, pp. 1–15, 2021.

[22] N. Abu-Ali, A. M. Taha, M. Salah, and H. Hassanein, "Uplink Scheduling in LTE and LTE-Advanced: Tutorial, Survey and Evaluation Framework," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 3, pp. 1239–1265, 2014.

[23] A. Anand, G. De Veciana, and S. Shakkottai, "Joint Scheduling of URLLC and eMBB Traffic in 5G Wireless Networks," in *Proc. of the IEEE INFOCOM*, Honolulu, HI, USA, 2018, pp. 1970–1978.

[24] A. Ksentini, P. A. Frangoudis, A. PC, and N. Nikaein, “Providing Low Latency Guarantees for Slicing-Ready 5G Systems via Two-Level MAC Scheduling,” *IEEE Network*, vol. 32, no. 6, pp. 116–123, 2018.

[25] K. A. Parekh and R. G. Gallager, “A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks - The Single Node Case,” *IEEE/ACM Transactions on Networking*, vol. 1, no. 3, pp. 344–357, Jun. 1993.

[26] P. Schulz *et al.*, “Latency Critical IoT Applications in 5G: Perspective on the Design of Radio Interface and Network Architecture,” *IEEE Communications Magazine*, vol. 55, no. 2, pp. 70–78, 2017.

[27] M. Keltsch *et al.*, “Remote Production and Mobile Contribution Over 5G Networks: Scenarios, Requirements and Approaches for Broadcast Quality Media Streaming,” in *Proc. of the IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*, Valencia, Spain, 2018, pp. 1–7.

[28] M. Boban, A. Kousaridas, K. Manolakis, J. Eichinger, and W. Xu, “Connected Roads of the Future: Use Cases, Requirements, and Design Considerations for Vehicle-to-Everything Communications,” *IEEE Vehicular Technology Magazine*, vol. 13, no. 3, pp. 110–123, 2018.

[29] F. E. Ritter, M. J. Schoelles, K. S. Quigley, and L. C. Klein, “Determining the number of model runs: Treating cognitive models as theories by not sampling their behavior,” in *Human-in-the-loop simulations: Methods and practice*, S. Narayanan and L. Rothrock, Eds., Springer-Verlag, 2011, pp. 97–116.

[30] R. Jain, D Chiu, and W Hawe, “A Quantitative Measure Of Fairness And Discrimination For Resource Allocation In Shared Computer Systems,” DEC, Research Report TR-301, 1984.

[31] H. Shi, R. V. Prasad, E. Onur, and I. Niemegeers, “Fairness in Wireless Networks: Issues, Measures and Challenges,” *IEEE Communications Surveys & Tutorials*, vol. 16, no. 1, pp. 5–24, 2014.

[32] F. Giust *et al.*, “MEC Deployments in 4G and Evolution Towards 5G,” ETSI, White Paper, 2018.

[33] T. K. Rodrigues, K. Suto, H. Nishiyama, J. Liu, and N. Kato, “Machine Learning Meets Computation and Communication Control in Evolving Edge and Cloud: Challenges and Future Perspective,” *IEEE Communications Surveys & Tutorials*, vol. 22, no. 1, pp. 38–67, 2020.

**Winston K.G. Seah** received the Dr.Eng. degree from Kyoto University, Kyoto, Japan, in 1997, and ME and BSc degrees from the National University of Singapore in 1993 and 1987 respectively. In 2009, he joined Victoria University of Wellington, New Zealand as Professor of Network Engineering. Prior to this, he has worked for more than 16 years in mission-oriented industrial research, taking ideas from theory to prototypes, most recently, as a Senior Scientist (Networking Protocols) in the Institute for Infocomm Research (I$^2$R), Singapore. Being actively involved in research in the areas of mobile ad hoc and sensor networks, he co-developed one of the first Quality of Service (QoS) models for mobile ad hoc networks and was one of the first to study networking protocols for wireless sensor networks powered solely by ambient energy harvesting. His latest research focuses on networking protocols and the use of artificial intelligence and machine learning techniques to address the needs of 5G/6G networks, Internet of Things (IoT) and other machine-type communications (MTC), encompassing both short range technologies (e.g. IEEE802.15.4, 6LoWPAN, and RPL) as well as, long-range communications (such as, LTE-A, NB-IoT, LoRaWAN, and IEEE802.11ah.) He is a Senior Member of the IEEE and Professional Member of the ACM.

**Chung-Hau Lee** received the B.S. degree in engineering science from National Cheng Kung University (NCKU), Tainan, Taiwan, in 2017, and the M.S. degree in computer science from National Yang-Ming Chiao-Tung University (NYCU), Hsinchu, Taiwan, in 2019. His research interests include wireless communications, mobile network protocol designs and machine learning algorithms for edge devices.

**Ying-Dar Lin** is a Chair Professor of computer science at National Yang-Ming Chiao-Tung University (NYCU), Taiwan. He received his Ph.D. in computer science from the University of California at Los Angeles (UCLA) in 1993. He was a visiting scholar at Cisco Systems in San Jose during 2007–2008, CEO at Telecom Technology Center, Taiwan, during 2010-2011, and Vice President of National Applied Research Labs (NARLabs), Taiwan, during 2017-2018. He was the founder and director of Network Benchmarking Lab (NBL) in 2002-2018, which reviewed network products with real traffic and automated tools. He also cofounded L7 Networks Inc. in 2002, later acquired by D-Link Corp, and O’Prueba Inc. in 2018, a spin-off from NBL. His research interests include cybersecurity, wireless communications, network softwarization, and machine learning for communications. His work on multi-hop cellular was the first along this line, and has been cited over 1000 times and standardized into IEEE 802.11s, IEEE 802.15.5, IEEE 802.16j, and 3GPP LTE-Advanced. He is an IEEE Fellow (class of 2013), IEEE Distinguished Lecturer (2014–2017), ONF (Open Networking Foundation) Research Associate (2014-2018), and received K. T. Li Breakthrough Award in 2017 and Research Excellence Award in 2017 and 2020. He has served or is serving on the editorial boards of several IEEE journals and magazines, including Editor-in-Chief of IEEE Communications Surveys and Tutorials (COMST, 2017-2020). He published a textbook, Computer Networks: An Open Source Approach, with Ren-Hung Hwang and Fred Baker (McGraw-Hill, 2011).

**Yuan-Cheng Lai** received his Ph.D. degree in the Department of Computer and Information Science from National Chiao Tung University in 1997. He joined the faculty of the Department of Information Management at National Taiwan University of Science and Technology in August 2001 and has been a distinguished professor since June 2012. His research interests include performance analysis, software-defined networking, wireless networks, and IoT security.