

# Dynamic Task Offloading and Resource Allocation for Ultra-Reliable Low-Latency Edge Computing

Chen-Feng Liu, *Student Member, IEEE*, Mehdi Bennis, *Senior Member, IEEE*,  
 Mérouane Debbah, *Fellow, IEEE*, and H. Vincent Poor, *Fellow, IEEE*  
*(Invited Paper)*

## Abstract

To overcome devices' limitations in performing computation-intensive applications, mobile edge computing (MEC) enables users to offload tasks to proximal MEC servers for faster task computation. However, current MEC system design is based on average-based metrics, which fails to account for the ultra-reliable low-latency requirements in mission-critical applications. To tackle this, this paper proposes a new system design, where probabilistic and statistical constraints are imposed on task queue lengths, by applying *extreme value theory*. The aim is to minimize users' power consumption while trading off the allocated resources for local computation and task offloading. Due to wireless channel dynamics, users are re-associated to MEC servers in order to offload tasks using higher rates or accessing proximal servers. In this regard, a user-server association policy is proposed, taking into account the channel quality as well as the servers' computation capabilities and workloads. By marrying tools from Lyapunov optimization and matching theory, a two-timescale mechanism is proposed, where a user-server association is solved in the long timescale while a dynamic task offloading and resource

This work was supported in part by the U.S. National Science Foundation under Grant CNS-1702808. This paper was presented in part at the IEEE Global Communications Conference Workshops, Singapore, December 2017 [1].

C.-F. Liu and M. Bennis are with the Centre for Wireless Communications, University of Oulu, 90014 Oulu, Finland (e-mail: chen-feng.liu@oulu.fi; mehdi.bennis@oulu.fi).

M. Debbah is with the Large Networks and System Group, CentraleSupélec, Université Paris-Saclay, 91192 Gif-sur-Yvette, France, and also with the Mathematical and Algorithmic Sciences Laboratory, Huawei France Research and Development, 92100 Paris, France (e-mail: merouane.debbah@huawei.com).

H. V. Poor is with the Department of Electrical Engineering, Princeton University, Princeton, NJ 08544, USA (e-mail: poor@princeton.edu).

allocation policy is executed in the short timescale. Simulation results corroborate the effectiveness of the proposed approach by guaranteeing highly-reliable task computation and lower delay performance, compared to baselines.

### Index Terms

5G and beyond, mobile edge computing (MEC), fog networking and computing, ultra-reliable low latency communications (URLLC), extreme value theory.

## I. INTRODUCTION

Motivated by the surging traffic demands spurred by online video and Internet-of-things (IoT) applications, including machine type and mission-critical communication (e.g., augmented/virtual reality (AR/VR) and drones), mobile edge computing (MEC)/fog computing are emerging technologies that distribute computations, communication, control, and storage at the network edge [2]–[6]. When executing the computation-intensive applications at mobile devices, the performance and user’s quality of experience are significantly affected by the device’s limited computation capability. Additionally, intensive computations are energy-consuming which severely shortens the lifetime of battery-limited devices. To address the computation and energy issues, mobile devices can wirelessly offload their tasks to proximal MEC servers. On the other hand, offloading tasks incurs additional latency which cannot be overlooked and should be taken into account in the system design. Hence, the energy-delay tradeoff has received significant attention and has been studied in various MEC systems [7]–[22].

### A. Related Work

In [7], Kwak *et al.* focused on an energy minimization problem for local computation and task offloading in a single-user MEC system. The authors further studied a multi-user system, which takes into account both the energy cost and monetary cost of task offloading [8]. Therein, the cost-delay tradeoff was investigated in terms of competition and cooperation among users and offloading service provider. Additionally, the work [9] considered the single-user system and assumed that the mobile device is endowed with a multi-core central process unit (CPU) to compute different applications simultaneously. In order to stabilize all task queues at the mobile device and MEC server, the dynamic task offloading and resource allocation policies were proposed by utilizing Lyapunov stochastic optimization in [7]–[9]. Assuming that the MEC

server is equipped with multiple CPU cores to compute different users' offloaded tasks in parallel, Mao *et al.* [10] studied a multi-user task offloading and bandwidth allocation problem. Subject to the stability of task queues, the energy-delay tradeoff was investigated using the Lyapunov framework. Extending the problem of [10], the authors further took into account the server's power consumption and resource allocation in the system analysis [11]. In [12], a wireless powered MEC network was considered in which multiple users, without fixed energy supply, are wirelessly powered by a power beacon to carry out local computation and task offloading. Taking into account the causality of the harvested energy, this work [12] aimed at maximizing energy efficiency subject to the stability of users' task queues. Therein, the tradeoff between energy efficiency and average execution delay was analyzed by stochastic optimization. Xu *et al.* studied another energy harvesting MEC scenario, in which the edge servers are mainly powered by solar or wind energy, whereas the cloud server has a constant grid power [13]. Aiming at minimizing the long-term expected cost which incorporates the end-to-end delay and operational cost, the authors proposed a reinforcement learning-based resource provisioning and workload offloading (to the cloud) to edge servers. Besides the transmission and computation delays, the work [14] took into account the cost (in terms of delay) of handover and computation migration, due to user mobility, in an ultra-dense network. Taking into the long-term available energy constraint, an online energy-aware base station association and handover algorithm was proposed to minimize the average end-to-end delay by incorporating Lyapunov optimization and multi-armed bandit theory [14]. Ko *et al.* [15] analyzed the average latency performance, including communication delay and computation delay, of a large-scale spatially random MEC network. Furthermore, an upper and a lower bound [15] on the average computation delay were derived by applying stochastic geometry and queuing theory. A hybrid cloud-fog architecture was considered in [16]. The delay-tolerable computation workloads, requested by the end users, are dispatched from the fog devices to the cloud servers when delay-sensitive workloads are computed at the fog devices. The studied problem was cast as a network-wide power minimization subject to an average delay requirement [16]. Focusing on the cloud-fog architecture, Lee *et al.* [17] studied a scenario in which a fog node distributes the offloaded tasks to the connected fog nodes and a remote cloud server for cooperative computation. To address the uncertainty of the arrival of neighboring fog nodes, an online fog network formation algorithm was proposed such that the maximal average latency among different computation nodes is minimized [17]. Considering a hierarchical cloudlet architecture, Fan and Ansari [18] proposed a workload allocation (among

different cloudlet tiers) and computational resource allocation approach in order to minimize the average response time of a task request. The authors further focused on an edge computing-based IoT network in which each user equipment (UE) can run several IoT applications [19]. Therein, the objective was to minimize the average response time subject to the delay requirements of different applications. In [20], a distributed workload balancing scheme was proposed for fog computing-empowered IoT networks. Based on the broadcast information of fog nodes' estimated traffic and computation loads, each IoT device locally chooses the associated fog node in order to reduce the average latency of its data flow. In addition to the task uploading and computation phases, the work [21] also accounted for the delay in the downlink phase, where the computed tasks are fed back to the users. The objective was to minimize a cost function of the estimated average delays of the three phases. The authors in [22] studied a software-defined fog network, where the data service subscribers (DSSs) purchase the fog nodes' computation resources via the data service operators. Modeling the average latency using queuing theory in the DSS's utility, a Stackelberg game and a many-to-many matching game were incorporated to allocate fog nodes' resources to the DSSs [22].

### *B. Our Contribution*

While conventional communication networks were engineered to boost network capacity, little attention has been paid to reliability and latency performance. Indeed, ultra-reliable and low latency communication (URLLC) is one of the pillars for enabling 5G and is currently receiving significant attention in both academia and industry [23]–[25]. Regarding the existing MEC literature, the vast majority considers the average delay as a performance metric or the quality-of-service requirement [13]–[22]. In other words, these system designs focus on latency through the lens of the average. In the works addressing the stochastic nature of the task arrival process [7]–[12], their prime concern is how to maintain the mean rate stability of task queues, i.e., ensuring a finite average queue length as time evolves [26]. However, merely focusing on the average-based performance is not sufficient to guarantee URLLC for mission-critical applications, which mandates a further examination in terms of bound violation probability, high-order statistics, characterization of the extreme events with very low occurrence probabilities, and so forth [24].

The main contribution of this work is to propose a URLLC-centric task offloading and resource allocation framework, by taking into account the statistics of extreme queue length events. We consider a multi-user MEC architecture with multiple servers having heterogeneous computation

resources. Due to the UE's limited computation capability and the additional incurred latency during task offloading, the UEs need to smartly allocate resources for local computation and the amount of tasks to offload via wireless transmission if the executed applications are latency-sensitive or mission-critical. Since the queue value is implicitly related to delay, we treat the former as a delay metric in this work. Motivated by the aforementioned drawbacks of average-based designs, we set a threshold for the queue length and impose a probabilistic requirement on the threshold deviation as a URLLC constraint. In order to model the event of threshold deviation, we characterize its statistics by invoking *extreme value theory* [27] and impose another URLLC constraint in terms of higher-order statistics. The problem is cast as a network-wide power minimization problem for task computation and offloading, subject to statistical URLLC constraints on the threshold deviation and extreme queue length events. Furthermore, we incorporate the UEs' mobility feature and propose a two-timescale UE-server association and task computation framework. In this regard, taking into account task queue state information, servers' computation capabilities and workloads, co-channel interference, and URLLC constraints, we associate the UEs with the MEC servers, in a long timescale, by utilizing matching theory [28]. Then, given the associated MEC server, task offloading and resource allocation are performed in the short timescale. To this end, we leverage Lyapunov stochastic optimization [26] to deal with the randomness of task arrivals, wireless channels, and task queue values. Simulation results show that considering the statistics of the extreme queue length as a reliability measure, the studied partially-offloading scheme includes more reliable task execution than the scheme without MEC servers and the fully-offloading scheme. In contrast with the received signal strength (RSS)-based baseline, our proposed UE-server association approach achieves better delay performance for heterogeneous MEC server architectures. The performance enhancement is more remarkable in denser networks.

The remainder of this paper is organized as follows. The system model is first specified in Section II. Subsequently, we formulate the latency requirements, reliability constraints, and the studied optimization problem in Section III. In Section IV, we detailedly specify the proposed UE-server association mechanism as well as the latency and reliability-aware task offloading and resource allocation framework. The network performance is evaluated numerically and discussed in Section V which is followed by Section VI for conclusions. Furthermore, for the sake of readability, we list all notations in Table II shown in Appendix A. The meaning of the notations will be detailedly defined in the following sections.

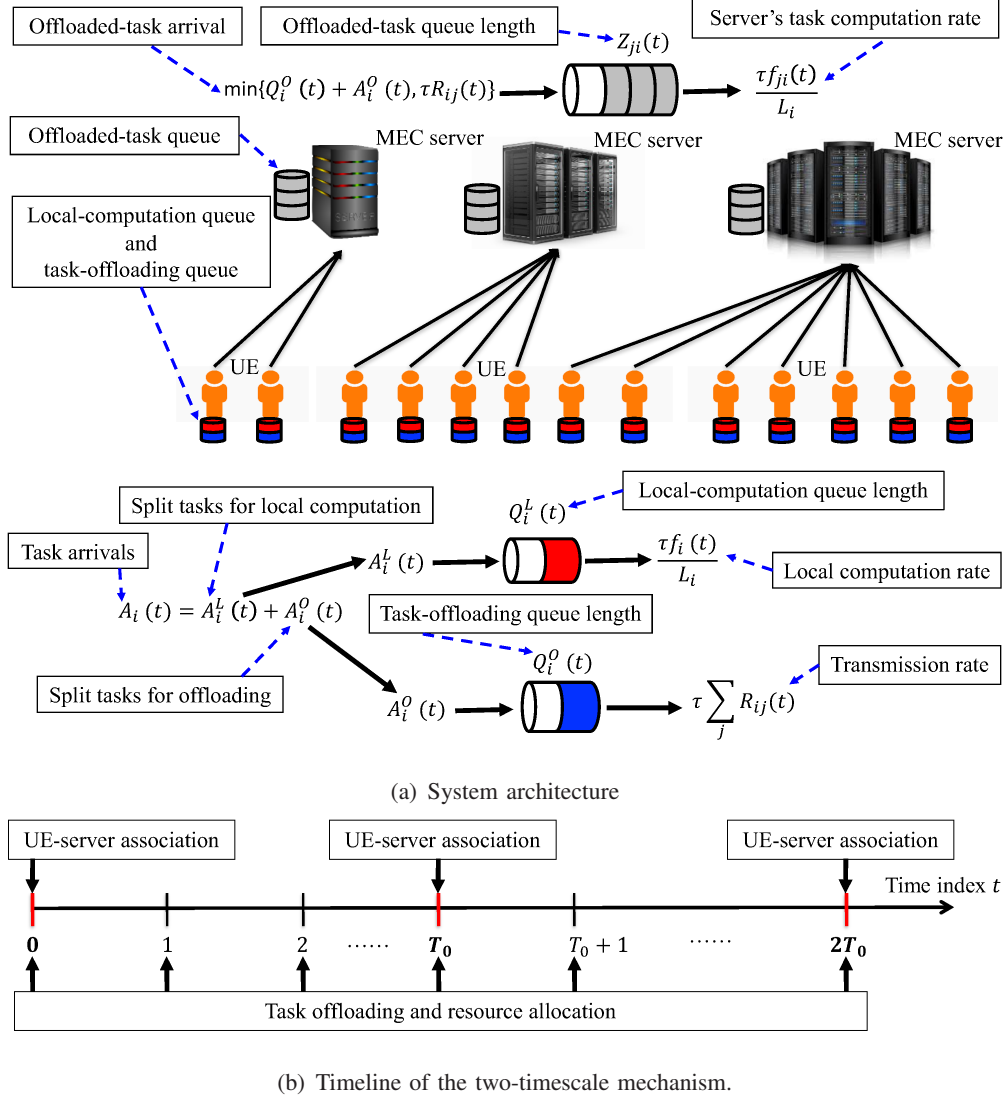


Figure 1. System model and timeline of the considered MEC network.

## II. SYSTEM MODEL

The considered MEC network consists of a set  $\mathcal{U}$  of  $U$  UEs and a set  $\mathcal{S}$  of  $S$  MEC servers. UEs have computation capabilities to execute their own tasks locally. However, due to the limited computation capabilities to execute computation-intensive applications, UEs can wirelessly offload their tasks to the MEC servers with an additional cost of communication latency. The MEC servers are equipped with multi-core CPUs such that different UEs' offloaded tasks can be computed in parallel. Additionally, the computation and communication timeline is slotted and indexed by  $t \in \mathbb{N}$  in which each time slot, with the slot length  $\tau$ , is consistent with the coherence block of the wireless channel. We further assume that UEs are randomly distributed and moves

continuously in the network, whereas the MEC servers are located in fixed positions. Since the UE's geographic location keeps changing, the UE is incentivized to offload its tasks to a different server which is closer to the UE, provides a stronger computation capability, or has the lower workload than the currently associated one. In this regard, we consider a two-timescale UE-server association and task-offloading mechanism. Specifically, we group every successive  $T_0$  time slots as a time frame, which is indexed by  $n \in \mathbb{Z}^+$  and denoted by  $\mathcal{T}(n) = [(n-1)T_0, \dots, nT_0 - 1]$ . In the beginning of each time frame (i.e., the long/slow timescale), each UE is associated with an MEC server. Let  $\eta_{ij}(n) \in \{0, 1\}$  represent the UE-server association indicator in the  $n$ th time frame, in which  $\eta_{ij}(n) = 1$  indicates that UE  $i$  can offload its tasks to server  $j$  during time frame  $n$ . Otherwise,  $\eta_{ij}(n) = 0$ . We also assume that each UE can only offload its tasks to one MEC server at a time. The UE-server association rule can be formulated as

$$\begin{cases} \eta_{ij}(n) \in \{0, 1\}, & \forall i \in \mathcal{U}, j \in \mathcal{S}, \\ \sum_{j \in \mathcal{S}} \eta_{ij}(n) = 1, & \forall i \in \mathcal{U}. \end{cases} \quad (1)$$

Subsequently in each time slot, i.e., the short/fast timescale, within the  $n$ th frame, each UE dynamically offloads part of the tasks to the associated MEC server and computes the remaining tasks locally. The network architecture and timeline of the considered MEC network are shown in Fig. 1.

#### A. Traffic Model at the UE Side

The UE uses one application in which tasks arrive in a stochastic manner. Following the data-partition model [5], we assume that each task can be computed locally, i.e., at the UE, or remotely, i.e., at the server. Different tasks are independent and can be computed in parallel. Thus, having the task arrivals  $A_i(t)$  in time slot  $t$ , each UE  $i$  divides its arrival into two disjoint parts in which one part  $A_i^L(t)$  is executed locally when the remaining tasks  $A_i^O(t)$  will be offloaded to the server. Task splitting at UE  $i \in \mathcal{U}$  can be expressed as

$$\begin{cases} A_i(t) = A_i^L(t) + A_i^O(t), \\ A_i^L(t), A_i^O(t) \in \{0, A_{\text{unit}}, 2A_{\text{unit}}, \dots\}. \end{cases} \quad (2)$$

Here,  $A_{\text{unit}}$  represents the unit task which cannot be further split. Moreover, we assume that task arrivals are independent and identically distributed (*i.i.d.*) over time with the average arrival rate  $\lambda_i = \mathbb{E}[A_i]/\tau$ .



Each UE has two queue buffers to store the split tasks for local computation and offloading. For the UE  $i$ 's local-computation queue, the queue length (in the unit of bits) in time slot  $t$  is denoted by  $Q_i^L(t)$  which evolves as

$$Q_i^L(t+1) = \max \left\{ Q_i^L(t) + A_i^L(t) - \frac{\tau f_i(t)}{L_i}, 0 \right\}, \quad \forall i \in \mathcal{U}. \quad (3)$$

Here,  $f_i(t)$  (in the unit of cycle/sec) is the UE  $i$ 's allocated CPU-cycle frequency to execute tasks when  $L_i$  accounts for the required CPU cycles per bit for computation, i.e., the processing density. The magnitude of the processing density depends on the performed application.<sup>1</sup> Furthermore, given a CPU-cycle frequency  $f_i(t)$ , the UE consumes the amount  $\kappa[f_i(t)]^3$  of power for computation.  $\kappa$  is a parameter affected by the device's hardware implementation [10], [29]. For UE  $i$ 's task-offloading queue, we denote the queue length (in the unit of bits) in time slot  $t$  as  $Q_i^O(t)$ . Analogously, the task-offloading queue dynamics is given by

$$Q_i^O(t+1) = \max \left\{ Q_i^O(t) + A_i^O(t) - \sum_{j \in \mathcal{S}} \tau R_{ij}(t), 0 \right\}, \quad \forall i \in \mathcal{U}, \quad (4)$$

in which

$$R_{ij}(t) = W \log_2 \left( 1 + \frac{\eta_{ij}(n) P_i(t) h_{ij}(t)}{N_0 W + \sum_{i' \in \mathcal{U} \setminus i} \eta_{i'j}(n) P_{i'}(t) h_{i'j}(t)} \right), \quad \forall i \in \mathcal{U}, j \in \mathcal{S}, \quad (5)$$

is UE  $i$ 's transmission rate<sup>2</sup> to offload tasks to the associated MEC server  $j$  in time slot  $t \in \mathcal{T}(n)$ .  $P_i(t)$  and  $N_0$  are UE  $i$ 's transmit power and the power spectral density of the additive white Gaussian noise (AWGN), respectively.  $W$  is the bandwidth dedicated to each server and shared by its associated UEs. Additionally,  $h_{ij}$  is the wireless channel gain between UE  $i \in \mathcal{U}$  and server  $j \in \mathcal{S}$ , including path loss and channel fading. We also assume that all channels experience block fading. In this work, we mainly consider the uplink, i.e., offloading tasks from the UE to the MEC server, and neglect the downlink, i.e., downloading the computed tasks from the server. The rationale is that compared with the offloaded tasks before computation, the computation results typically have smaller sizes [15], [30], [31]. Hence, the overheads in the downlink can be neglected.

In order to minimize the total power consumption of resource allocation for local computation and task offloading, the UE adopts the dynamic voltage and frequency scaling (DVFS) capability

<sup>1</sup>For example, the six-queen puzzle, 400-frame video game, seven-queen puzzle, face recognition, and virus scanning require the processing densities of 1760 cycle/bit, 2640 cycle/bit, 8250 cycle/bit, 31680 cycle/bit, and 36992 cycle/bit, respectively [7].

<sup>2</sup>All transmissions are encoded based on a Gaussian distribution.



to adaptively adjust its CPU-cycle frequency [5], [29]. Thus, to allocate the CPU-cycle frequency and transmit power, we impose the following constraints at each UE  $i \in \mathcal{U}$ , i.e.,

$$\begin{cases} \kappa[f_i(t)]^3 + P_i(t) \leq P_i^{\max}, \\ f_i(t) \geq 0, \\ P_i(t) \geq 0, \end{cases} \quad (6)$$

where  $P_i^{\max}$  is UE  $i$ 's power budget.

### B. Traffic Model at the Server Side

We assume that each MEC server has distinct queue buffers to store different UEs' offloaded tasks, where the queue length (in bits) of the UE  $i$ 's offloaded tasks at server  $j$  in time slot  $t$  is denoted by  $Z_{ji}(t)$ . The offloaded-task queue length evolves as

$$Z_{ji}(t+1) = \max \left\{ Z_{ji}(t) + \min \{ Q_i^O(t) + A_i^O(t), \tau R_{ij}(t) \} - \frac{\tau f_{ji}(t)}{L_i}, 0 \right\} \quad (7)$$

$$\leq \max \left\{ Z_{ji}(t) + \tau R_{ij}(t) - \frac{\tau f_{ji}(t)}{L_i}, 0 \right\}, \quad \forall i \in \mathcal{U}, j \in \mathcal{S}. \quad (8)$$

Here,  $f_{ji}(t)$  is the server  $j$ 's allocated CPU-cycle frequency to process UE  $i$ 's offloaded tasks. Note that the MEC server is deployed to provide a faster computation capability for the UE. Thus, we consider the scenario in which each CPU core of the MEC server is dedicated to at most one UE (i.e., its offloaded tasks) in each time slot, and a UE's offloaded tasks at each server can only be computed by one CPU core at a time [9], [10]. The considered computational resource scheduling mechanism at the MEC server is mathematically formulated as

$$\begin{cases} \sum_{i \in \mathcal{U}} \mathbb{1}_{\{f_{ji}(t) > 0\}} \leq N_j, \quad \forall j \in \mathcal{S}, \\ f_{ji}(t) \in \{0, f_j^{\max}\}, \quad \forall i \in \mathcal{U}, j \in \mathcal{S}, \end{cases} \quad (9)$$

where  $N_j$  denotes the total CPU-core number of server  $j$ ,  $f_j^{\max}$  is server  $j$ 's computation capability of one CPU core, and  $\mathbb{1}_{\{\cdot\}}$  is the indicator function. In (9), we account for the allocated CPU-cycle frequencies to all UEs even though some UEs are not associated with this server in the current time frame. The rationale will be detailedly explained in Section IV-D after formulating the concerned optimization problem. Additionally, in order to illustrate the relationship between the offloaded-task queue length and the transmission rate, we introduce inequality (8) which will be further used to formulate the latency and reliability requirements of the considered MEC system and derive the solution of the studied optimization problem.

### III. LATENCY REQUIREMENTS, RELIABILITY CONSTRAINTS, AND PROBLEM FORMULATION

In this work, the end-to-end delays experienced by the locally-computed tasks  $A_i^L(t)$  and offloaded tasks  $A_i^O(t)$  consist of different components. When the task is computed locally, it experiences the queuing delay (for computation) and computation delay at the UE. If the task is offloaded to the MEC server, the end-to-end delay includes: 1) queuing delay (for offloading) at the UE, 2) wireless transmission delay while offloading, 3) queuing delay (for computation) at the server, and 4) computation delay at the server. From Little's law, we know that the average queuing delay is proportional to the average queue length [32]. However, without taking the tail distribution of the queue length into account, solely focusing on the average queue length fails to account for the low-latency and reliability requirement [24]. To tackle this, we focus on the statistics of the task queue and impose probabilistic constraints on the local-computation and task-offloading queue lengths as follows:

$$\lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=1}^T \Pr(Q_i^L(t) > d_i^L) \leq \epsilon_i^L, \quad \forall i \in \mathcal{U}, \quad (10)$$

$$\lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=1}^T \Pr(Q_i^O(t) > d_i^O) \leq \epsilon_i^O, \quad \forall i \in \mathcal{U}. \quad (11)$$

Here,  $d_i^L$  and  $d_i^O$  are the queue length bounds when  $\epsilon_i^L \ll 1$  and  $\epsilon_i^O \ll 1$  are the tolerable bound violation probabilities. Furthermore, the queue length bound violation also undermines the reliability issue of task computation. For example, if a finite-size queue buffer is over-loaded, the incoming tasks will be dropped.

In addition to the bound violation probability, let us look at the complementary cumulative distribution function (CCDF) of the UE's local-computation queue length, i.e.,  $\bar{F}_{Q_i^L}(q) = \Pr(Q_i^L > q)$ , which reflects the queue length profile. If the monotonically decreasing CCDF decays faster while increasing  $q$ , the probability of having an extreme queue length is lower. Since the prime concern in this work lies in the extreme-case events with very low occurrence probabilities, i.e.,  $\Pr(Q_i^L(t) > d_i) \ll 1$ , we resort to principles of extreme value theory<sup>3</sup> to characterize the statistics and tail distribution of the extreme event  $Q_i^L(t) > d_i$ . To this end, we first introduce the *Pickands–Balkema–de Haan theorem* [27].

<sup>3</sup>Extreme value theory is a powerful and robust framework to study the tail behavior of a distribution. Extreme value theory also provides statistical models for the computation of extreme risk measures.

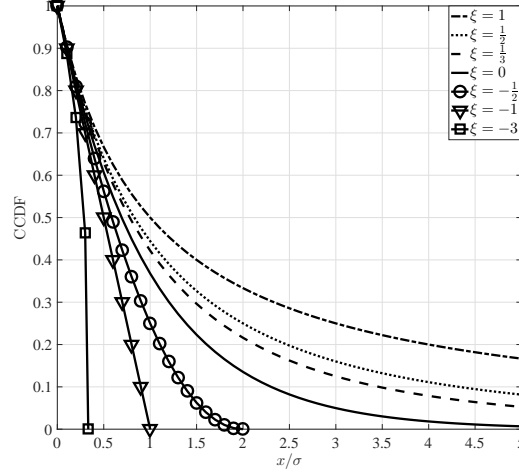


Figure 2. CCDFs of the GPDs for various shape parameters  $\xi$ .

**Theorem 1 (Pickands–Balkema–de Haan theorem).** Consider a random variable  $Q$ , with the cumulative distribution function (CDF)  $F_Q(q)$ , and a threshold value  $d$ . As the threshold  $d$  closely approaches  $F_Q^{-1}(1)$ , i.e.,  $d \rightarrow \sup\{q: F_Q(q) < 1\}$ , the conditional CCDF of the excess value  $X|_{Q>d} = Q - d > 0$ , i.e.,  $\bar{F}_{X|Q>d}(x) = \Pr(Q - d > x | Q > d)$ , can be approximated by a generalized Pareto distribution (GPD)  $G(x; \sigma, \xi)$ , i.e.,

$$\bar{F}_{X|Q>d}(x) \approx G(x; \sigma, \xi) = \begin{cases} \left(1 + \frac{\xi x}{\sigma}\right)^{-1/\xi}, & \text{where } x \geq 0 \text{ and } \xi > 0, \\ e^{-x/\sigma}, & \text{where } x \geq 0 \text{ and } \xi = 0, \\ \left(1 + \frac{\xi x}{\sigma}\right)^{-1/\xi}, & \text{where } 0 \leq x \leq -\sigma/\xi \text{ and } \xi < 0, \end{cases} \quad (12a)$$

$$\bar{F}_{X|Q>d}(x) \approx G(x; \sigma, \xi) = \begin{cases} e^{-x/\sigma}, & \text{where } x \geq 0 \text{ and } \xi = 0, \end{cases} \quad (12b)$$

$$\bar{F}_{X|Q>d}(x) \approx G(x; \sigma, \xi) = \begin{cases} \left(1 + \frac{\xi x}{\sigma}\right)^{-1/\xi}, & \text{where } 0 \leq x \leq -\sigma/\xi \text{ and } \xi < 0, \end{cases} \quad (12c)$$

which is characterized by a scale parameter  $\sigma > 0$  and a shape parameter  $\xi \in \mathbb{R}$ .

In other words, the conditional CCDF of the excess value  $X|_{Q>d}$  converges to a GPD as  $d \rightarrow \infty$ . However, from the proof [27] for Theorem 1, we know that the GPD provides a good approximation when  $F_Q(d)$  is close to 1, e.g.,  $F_Q(d) = 0.99$ . That is, depending on the CDF of  $Q$ , imposing a very large  $d$  might not be necessary for obtaining the approximated GPD. Moreover, for a GPD  $G(x; \sigma, \xi)$ , its mean  $\sigma/(1 - \xi)$  and other higher-order statistics such as variance  $\frac{\sigma^2}{(1-\xi)^2(1-2\xi)}$  and skewness exist if  $\xi < 1$ ,  $\xi < \frac{1}{2}$ , and  $\xi < \frac{1}{3}$ , respectively. Note that the scale parameter  $\sigma$  and the domain  $x$  of  $G(x; \sigma, \xi)$  are in the same order. In this regard, we can see that  $G(\sigma; \sigma, 0) = e^{-1} = 0.37$  at  $x = \sigma$  and  $G(3\sigma; \sigma, 0) = e^{-3} = 0.05$  at  $x = 3\sigma$  in (12b). We also show the CCDFs of the GPDs for various shape parameters  $\xi$  in Fig. 2, where the x-axis

is indexed with respect to the normalized value  $x/\sigma$ . As shown in Fig. 2, the decay speed of the CCDF increases as  $\xi$  decreases. In contrast with the curves with  $\xi \geq 0$ , we can see that the CCDF decays rather sharply when  $\xi \leq -3$ .

Now, let us denote the excess value (with respect to the threshold  $d_i^L$  in (10)) of the local-computation queue of each UE  $i \in \mathcal{U}$  in time slot  $t$  as  $X_i^L(t)|_{Q_i^L(t) > d_i^L} = Q_i^L(t) - d_i^L > 0$ . By applying Theorem 1, the excess queue value can be approximated by a GPD  $G(x_i; \sigma_i^L, \xi_i^L)$  whose mean and variance are

$$\mathbb{E}[X_i^L(t)|Q_i^L(t) > d_i^L] \approx \frac{\sigma_i^L}{1 - \xi_i^L}, \quad (13)$$

$$\text{Var}(X_i^L(t)|Q_i^L(t) > d_i^L) \approx \frac{(\sigma_i^L)^2}{(1 - \xi_i^L)^2(1 - 2\xi_i^L)}, \quad (14)$$

with the corresponding scale parameter  $\sigma_i^L$  and shape parameter  $\xi_i^L$ . In (13) and (14), we can find that the smaller  $\sigma_i^L$  and  $\xi_i^L$  are, the smaller the mean value and variance. Since the approximated GPD is just characterized by the scale and shape parameters as mentioned previously, therefore, we impose thresholds on these two parameters, i.e.,  $\sigma_i^L \leq \sigma_i^{L,\text{th}}$  and  $\xi_i^L \leq \xi_i^{L,\text{th}}$ . The selection of threshold values can be referred to the above discussions about the GPD, Fig. 2, and the magnitude of the interested metric's values. Subsequently, applying the two parameter thresholds and  $\text{Var}(X_i^L) = \mathbb{E}[(X_i^L)^2] - \mathbb{E}[X_i^L]^2$  to (13) and (14), we consider the constraints on the long-term time-averaged conditional mean and second moment of the excess value of each UE's local-computation queue length, i.e.,

$$\bar{X}_i^L = \lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=1}^T \mathbb{E}[X_i^L(t)|Q_i^L(t) > d_i^L] \leq \frac{\sigma_i^{L,\text{th}}}{1 - \xi_i^{L,\text{th}}}, \quad \forall i \in \mathcal{U}, \quad (15)$$

$$\bar{Y}_i^L = \lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=1}^T \mathbb{E}[Y_i^L(t)|Q_i^L(t) > d_i^L] \leq \frac{2(\sigma_i^{L,\text{th}})^2}{(1 - \xi_i^{L,\text{th}})(1 - 2\xi_i^{L,\text{th}})}, \quad \forall i \in \mathcal{U}, \quad (16)$$

with  $Y_i^L(t) = [X_i^L(t)]^2$ . Analogously, denoting the excess value, with respect to the threshold  $d_i^O$ , of UE  $i$ 's task-offloading queue length in time slot  $t$  as  $X_i^O(t)|_{Q_i^O(t) > d_i^O} = Q_i^O(t) - d_i^O > 0$ , we have the constraints on the long-term time-averaged conditional mean and second moment

$$\bar{X}_i^O = \lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=1}^T \mathbb{E}[X_i^O(t)|Q_i^O(t) > d_i^O] \leq \frac{\sigma_i^{O,\text{th}}}{1 - \xi_i^{O,\text{th}}}, \quad \forall i \in \mathcal{U}, \quad (17)$$

$$\bar{Y}_i^O = \lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=1}^T \mathbb{E}[Y_i^O(t)|Q_i^O(t) > d_i^O] \leq \frac{2(\sigma_i^{O,\text{th}})^2}{(1 - \xi_i^{O,\text{th}})(1 - 2\xi_i^{O,\text{th}})}, \quad \forall i \in \mathcal{U}, \quad (18)$$

in which  $\sigma_i^{O,\text{th}}$  and  $\xi_i^{O,\text{th}}$  are the thresholds for the characteristic parameters of the approximated GPD, and  $Y_i^O(t) = [X_i^O(t)]^2$ .

Likewise, the average queuing delay at the server is proportional to the ratio of the average queue length to the average transmission rate. Referring to (8), we consider the probabilistic constraint as follows:

$$\lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=1}^T \Pr \left( \frac{Z_{ji}(t)}{\tilde{R}_{ij}(t-1)} > d_{ji} \right) \leq \epsilon_{ji}, \quad \forall i \in \mathcal{U}, j \in \mathcal{S}, \quad (19)$$

with the threshold  $d_{ji}$  and tolerable violation probability  $\epsilon_{ji} \ll 1$ , on the offloaded-task queue length at the MEC server.  $\tilde{R}_{ij}(t-1) = \frac{1}{t} \sum_{\tau=0}^{t-1} R_{ij}(\tau)$  is the moving time-averaged transmission rate. Similar to the task queue lengths at the UE side, we further denote the excess value, with respect to the threshold  $\tilde{R}_{ij}(t-1)d_{ji}$ , in time slot  $t$  as  $X_{ji}(t)|_{Z_{ji}(t) > \tilde{R}_{ij}(t-1)d_{ji}} = Z_{ji}(t) - \tilde{R}_{ij}(t-1)d_{ji} > 0$  of the offloaded-task queue length of UE  $i \in \mathcal{U}$  at server  $j \in \mathcal{S}$  and impose the constraints as follows:

$$\bar{X}_{ji} = \lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=1}^T \mathbb{E}[X_{ji}(t)|Z_{ji}(t) > \tilde{R}_{ij}(t-1)d_{ji}] \leq \frac{\sigma_{ji}^{\text{th}}}{1 - \xi_{ji}^{\text{th}}}, \quad (20)$$

$$\bar{Y}_{ji} = \lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=1}^T \mathbb{E}[Y_{ji}(t)|Z_{ji}(t) > \tilde{R}_{ij}(t-1)d_{ji}] \leq \frac{2(\sigma_{ji}^{\text{th}})^2}{(1 - \xi_{ji}^{\text{th}})(1 - 2\xi_{ji}^{\text{th}})}, \quad (21)$$

with  $Y_{ji}(t) = [X_{ji}(t)]^2$ . Here,  $\sigma_{ji}^{\text{th}}$  and  $\xi_{ji}^{\text{th}}$  are the thresholds for the characteristic parameters of the approximated GPD.

We note that the local computation delay at the UE and the transmission delay while offloading are inversely proportional to the computation speed  $f_i(t)/L_i$  and the transmission rate  $\sum_{j \in \mathcal{S}} R_{ij}(t)$  as per (3) and (4), respectively. To decrease the local computation and transmission delays, the UE should allocate a higher local CPU-cycle frequency and more transmit power, which, on the other hand, incurs energy shortage. Since allocating a higher CPU-cycle frequency and more transmit power can also further decrease the queue length, both (local computation and transmission) delays are implicitly taken into account in the queue length constraints (10), (11), and (15)–(18). At the server side, the remote computation delay can be neglected because one CPU core with the better computation capability is dedicated to one UE's offloaded tasks at a time. On the other hand, the server needs to schedule its computational resources, i.e., multiple CPU cores, when the associated UEs are more than the CPU cores.

Incorporating the aforementioned latency requirements and reliability constraints, the studied

optimization problem is formulated as follows:

$$\begin{aligned}
\mathbf{MP}: \quad & \underset{\boldsymbol{\eta}(n), \mathbf{f}(t), \mathbf{P}(t)}{\text{minimize}} \quad \sum_{i \in \mathcal{U}} (\bar{P}_i^C + \bar{P}_i^T) \\
& \text{subject to} \quad (1) \text{ for UE-server association,} \\
& \quad (2) \text{ for task splitting,} \\
& \quad (6) \text{ and (9) for resource allocation,} \\
& \quad (10), (11), \text{ and (19) for queue length bound violation,} \\
& \quad (15)–(18), (20), \text{ and (21) for the GPDs,}
\end{aligned}$$

where  $\bar{P}_i^C = \lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=0}^{T-1} \kappa[f_i(t)]^3$  and  $\bar{P}_i^T = \lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=0}^{T-1} P_i(t)$  are the UE  $i$ 's long-term time-averaged power consumptions for local computation and task offloading, respectively.  $\boldsymbol{\eta}(n) = (\eta_{ij}(n) : i \in \mathcal{U}, j \in \mathcal{S})$  and  $\mathbf{P}(t) = (P_i(t) : i \in \mathcal{U})$  denote the network-wide UE-server association and transmit power allocation vectors, respectively. In addition,  $\mathbf{f}(t) = (f_i(t), \mathbf{f}_j(t) : i \in \mathcal{U}, j \in \mathcal{S})$  denotes the network-wide computational resource allocation vector in which  $\mathbf{f}_j(t) = (f_{ji}(t) : i \in \mathcal{U}, j \in \mathcal{S})$  is the computational resource allocation vector of server  $j$ . To solve problem **MP**, we utilize techniques from Lyapunov stochastic optimization and propose a dynamic task offloading and resource allocation policy in the next section.

#### IV. LATENCY AND RELIABILITY-AWARE TASK OFFLOADING AND RESOURCE ALLOCATION

Let us give an overview of the proposed task offloading and resource allocation approach before specifying the details. In the beginning of each time frame, i.e., every  $T_0$  slots, we carry out a UE-server association, taking into account the wireless link strength, the UEs' and servers' computation capabilities, their historical workloads, and URLLC constraints (11) and (17)–(21). To this end, a many-to-one matching algorithm is utilized to associate each server with multiple UEs. Afterwards, we focus on task offloading and resource allocation by solving three decomposed optimization problems, via Lyapunov optimization, in each time slot. At the UE side, each UE splits its instantaneous task arrivals into two parts, which will be computed locally and offloaded respectively, while allocating the local computation CPU-cycle frequency and transmit power for offloading. At the server side, each MEC server schedules its CPU cores to execute the UEs' offloaded tasks. In the procedures (of task splitting and offloading, resource allocation, and CPU-core scheduling), the related URLLC constraints out of (10), (11), and (15)–

(21) are considered. The details of our proposed approach will be illustrated in the remainder of this section.

#### A. Lyapunov Optimization Framework

We first introduce a virtual queue  $Q_i^{L,(X)}$  for the long-term time-averaged constraint (15) with the queue evolution as follows:

$$Q_i^{L,(X)}(t+1) = \max \left\{ Q_i^{L,(X)}(t) + \left( X_i^L(t+1) - \frac{\sigma_i^{L,\text{th}}}{1 - \xi_i^{L,\text{th}}} \right) \times \mathbb{1}_{\{Q_i^L(t+1) > d_i^L\}}, 0 \right\}, \quad \forall i \in \mathcal{U}, \quad (22)$$

in which the incoming traffic amount  $X_i^L(t+1) \times \mathbb{1}_{\{Q_i^L(t+1) > d_i^L\}}$  and outgoing traffic amount  $\frac{\sigma_i^{L,\text{th}}}{1 - \xi_i^{L,\text{th}}} \times \mathbb{1}_{\{Q_i^L(t+1) > d_i^L\}}$  correspond to the left-hand side and right-hand side of the inequality (15), respectively. Note that [26] ascertains that the introduced virtual queue is *mean rate stable*, i.e.,  $\lim_{t \rightarrow \infty} \frac{\mathbb{E}[|Q_i^{L,(X)}(t)|]}{t} \rightarrow 0$ , is equivalent to satisfying the long-term time-averaged constraint (15). Analogously, for the constraints (16)–(18), (20), and (21), we respectively introduce the virtual queues as follows:

$$Q_i^{L,(Y)}(t+1) = \max \left\{ Q_i^{L,(Y)}(t) + \left( Y_i^L(t+1) - \frac{2(\sigma_i^{L,\text{th}})^2}{(1 - \xi_i^{L,\text{th}})(1 - 2\xi_i^{L,\text{th}})} \right) \times \mathbb{1}_{\{Q_i^L(t+1) > d_i^L\}}, 0 \right\}, \quad \forall i \in \mathcal{U}, \quad (23)$$

$$Q_i^{O,(X)}(t+1) = \max \left\{ Q_i^{O,(X)}(t) + \left( X_i^O(t+1) - \frac{\sigma_i^{O,\text{th}}}{1 - \xi_i^{O,\text{th}}} \right) \times \mathbb{1}_{\{Q_i^O(t+1) > d_i^O\}}, 0 \right\}, \quad \forall i \in \mathcal{U}, \quad (24)$$

$$Q_i^{O,(Y)}(t+1) = \max \left\{ Q_i^{O,(Y)}(t) + \left( Y_i^O(t+1) - \frac{2(\sigma_i^{O,\text{th}})^2}{(1 - \xi_i^{O,\text{th}})(1 - 2\xi_i^{O,\text{th}})} \right) \times \mathbb{1}_{\{Q_i^O(t+1) > d_i^O\}}, 0 \right\}, \quad \forall i \in \mathcal{U}, \quad (25)$$

$$Q_{ji}^{(X)}(t+1) = \max \left\{ Q_{ji}^{(X)}(t) + \left( X_{ji}(t+1) - \frac{\sigma_{ji}^{\text{th}}}{1 - \xi_{ji}^{\text{th}}} \right) \times \mathbb{1}_{\{Z_{ji}(t+1) > \bar{R}_{ij}(t)d_{ji}\}}, 0 \right\}, \quad \forall i \in \mathcal{U}, j \in \mathcal{S}. \quad (26)$$

$$Q_{ji}^{(Y)}(t+1) = \max \left\{ Q_{ji}^{(Y)}(t) + \left( Y_{ji}^k(t+1) - \frac{2(\sigma_{ji}^{\text{th}})^2}{(1 - \xi_{ji}^{\text{th}})(1 - 2\xi_{ji}^{\text{th}})} \right) \times \mathbb{1}_{\{Z_{ji}(t+1) > \bar{R}_{ij}(t)d_{ji}\}}, 0 \right\}, \quad \forall i \in \mathcal{U}, j \in \mathcal{S}. \quad (27)$$



Additionally, given an event  $B$  and the set of all possible outcomes  $\Omega$ , we can derive  $\mathbb{E}[\mathbb{1}_{\{B\}}] = 1 \cdot \Pr(B) + 0 \cdot \Pr(\Omega \setminus B) = \Pr(B)$ . By applying  $\mathbb{E}[\mathbb{1}_{\{B\}}] = \Pr(B)$ , constraints (10), (11), and (19) can be equivalently rewritten as

$$\lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=1}^T \mathbb{E}[\mathbb{1}_{\{Q_i^L(t) > d_i^L\}}] \leq \epsilon_i^L, \quad \forall i \in \mathcal{U}, \quad (28)$$

$$\lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=1}^T \mathbb{E}[\mathbb{1}_{\{Q_i^O(t) > d_i^O\}}] \leq \epsilon_i^O, \quad \forall i \in \mathcal{U}, \quad (29)$$

$$\lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=1}^T \mathbb{E}[\mathbb{1}_{\{Z_{ji}(t) > \tilde{R}_{ij}(t-1)d_{ji}\}}] \leq \epsilon_{ji}, \quad \forall i \in \mathcal{U}, j \in \mathcal{S}. \quad (30)$$

Then let us follow the above steps. The corresponding virtual queues of (28)–(30) are expressed as

$$Q_i^{L,(Q)}(t+1) = \max \left\{ Q_i^{L,(Q)}(t) + \mathbb{1}_{\{Q_i^L(t+1) > d_i^L\}} - \epsilon_i^L, 0 \right\}, \quad \forall i \in \mathcal{U}, \quad (31)$$

$$Q_i^{O,(Q)}(t+1) = \max \left\{ Q_i^{O,(Q)}(t) + \mathbb{1}_{\{Q_i^O(t+1) > d_i^O\}} - \epsilon_i^O, 0 \right\}, \quad \forall i \in \mathcal{U}, \quad (32)$$

$$Q_{ji}^{(Z)}(t+1) = \max \left\{ Q_{ji}^{(Z)}(t) + \mathbb{1}_{\{Z_{ji}(t+1) > \tilde{R}_{ij}(t)d_{ji}\}} - \epsilon_{ji}, 0 \right\}, \quad \forall i \in \mathcal{U}, j \in \mathcal{S}. \quad (33)$$

Now problem **MP** is equivalently transferred to [26]

$$\begin{aligned} \mathbf{MP'}: \quad & \underset{\boldsymbol{\eta}(n), \mathbf{f}(t), \mathbf{P}(t)}{\text{minimize}} \quad \sum_{i \in \mathcal{U}} (\bar{P}_i^C + \bar{P}_i^T) \\ & \text{subject to} \quad (1), (2), (6), \text{ and } (9), \end{aligned}$$

Stability of (22)–(27) and (31)–(33).

To solve problem **MP'**, we let  $\mathbf{Q}(t) = (Q_i^{L,(X)}(t), Q_i^{L,(Y)}(t), Q_i^{O,(X)}(t), Q_i^{O,(Y)}(t), Q_{ji}^{(X)}(t), Q_{ji}^{(Y)}(t), Q_i^{L,(Q)}(t), Q_i^{O,(Q)}(t), Q_{ji}^{(Z)}(t) : i \in \mathcal{U}, j \in \mathcal{S})$  denote the combined queue vector for notational simplicity and express the conditional Lyapunov drift-plus-penalty for slot  $t$  as

$$\mathbb{E} \left[ \mathcal{L}(\mathbf{Q}(t+1)) - \mathcal{L}(\mathbf{Q}(t)) + \sum_{i \in \mathcal{U}} V(\kappa[f_i(t)]^3 + P_i(t)) \middle| \mathbf{Q}(t) \right], \quad (34)$$

where

$$\begin{aligned} \mathcal{L}(\mathbf{Q}(t)) = & \frac{1}{2} \sum_{i \in \mathcal{U}} \left( [Q_i^{L,(X)}(t)]^2 + [Q_i^{L,(Y)}(t)]^2 + [Q_i^{O,(X)}(t)]^2 + [Q_i^{O,(Y)}(t)]^2 \right. \\ & \left. + [Q_i^{L,(Q)}(t)]^2 + [Q_i^{O,(Q)}(t)]^2 \right) + \frac{1}{2} \sum_{i \in \mathcal{U}} \sum_{j \in \mathcal{S}} \left( [Q_{ji}^{(X)}(t)]^2 + [Q_{ji}^{(Y)}(t)]^2 + [Q_{ji}^{(Z)}(t)]^2 \right) \end{aligned}$$

is the Lyapunov function. The term  $V \geq 0$  is a parameter which trades off objective optimality and queue length reduction. Subsequently, plugging the inequality  $(\max\{x, 0\})^2 \leq x^2$ , all physical and virtual queue dynamics, and (8) into (34), we can derive

$$\begin{aligned}
(34) \leq & C + \mathbb{E} \left[ \sum_{i \in \mathcal{U}} \left[ \left( Q_i^{L,(X)}(t) + Q_i^L(t) + 2Q_i^{L,(Y)}(t)Q_i^L(t) + 2[Q_i^L(t)]^3 \right) \left( A_i^L(t) - \frac{\tau f_i(t)}{L_i} \right) \right. \right. \\
& \times \mathbb{1}_{\{Q_i^L(t) + A_i(t) > d_i^L\}} + Q_i^{L,(Q)}(t) \times \mathbb{1}_{\{\max\{Q_i^L(t) + A_i^L(t) - \tau f_i(t)/L_i, 0\} > d_i^L\}} \left. \right] \\
& + \sum_{i \in \mathcal{U}} \left[ \left( Q_i^{O,(X)}(t) + Q_i^O(t) + 2Q_i^{O,(Y)}(t)Q_i^O(t) + 2[Q_i^O(t)]^3 \right) \left( A_i^O(t) - \sum_{j \in \mathcal{S}} \tau R_{ij}(t) \right) \right. \\
& \times \mathbb{1}_{\{Q_i^O(t) + A_i(t) > d_i^O\}} + Q_i^{O,(Q)}(t) \times \mathbb{1}_{\{\max\{Q_i^O(t) + A_i^O(t) - \sum_{j \in \mathcal{S}} \tau R_{ij}(t), 0\} > d_i^O\}} \left. \right] \\
& + \sum_{i \in \mathcal{U}} \sum_{j \in \mathcal{S}} \left[ \left( Q_{ji}^{(X)}(t) + Z_{ji}(t) + 2Q_{ji}^{(Y)}(t)Z_{ji}(t) + 2[Z_{ji}(t)]^3 \right) \left( \tau R_{ij}(t) - \frac{\tau f_{ji}(t)}{L_i} \right) \right. \\
& \times \mathbb{1}_{\{Z_{ji}(t) + \tau R_{ij}^{\max}(t) > \tilde{R}_{ij}(t-1)d_{ji}\}} + Q_{ji}^{(Z)}(t) \times \mathbb{1}_{\{\max\{Z_{ji}(t) + \tau R_{ij}(t) - \tau f_{ji}(t)/L_i, 0\} > \tilde{R}_{ij}(t-1)d_{ji}\}} \left. \right] \\
& + \sum_{i \in \mathcal{U}} V(\kappa[f_i(t)]^3 + P_i(t)) \Big| \mathbf{Q}(t) \Big]. \tag{35}
\end{aligned}$$

Here,  $R_{ij}^{\max}(t) = W \log_2 \left( 1 + \frac{P_i^{\max} h_{ij}(t)}{N_0 W} \right)$  is UE  $i$ 's maximum offloading rate. Since the constant  $C$  does not affect the system performance in Lyapunov optimization, we omit its details in (35) for expression simplicity. Note that a solution to problem **MP'** can be obtained by minimizing the upper bound (35) in each time slot  $t$ , in which the optimality of **MP'** is asymptotically approached by increasing  $V$  [26]. To minimize (35), we have three decomposed optimization problems **P1**, **P2**, and **P3** which are detailed and solved in the following parts.

The first decomposed problem, which jointly associates UEs with MEC servers and allocates UEs' computational and communication resources, is given by

$$\begin{aligned}
\mathbf{P1}: \quad & \underset{\boldsymbol{\eta}(n), \mathbf{f}(t), \mathbf{P}(t)} \text{ minimize}}{\sum_{i \in \mathcal{U}} \sum_{j \in \mathcal{S}} (\beta_{ji}(t) - \beta_i^O(t)) \tau W \log_2 \left( 1 + \frac{\eta_{ij}(n) P_i(t) h_{ij}(t)}{N_0 W + \sum_{i' \in \mathcal{U} \setminus i} \eta_{i'j}(n) P_{i'}(t) h_{i'j}(t)} \right)} \\
& - \sum_{i \in \mathcal{U}} \frac{\beta_i^L(t) \tau f_i(t)}{L_i} + \sum_{i \in \mathcal{U}} V(\kappa[f_i(t)]^3 + P_i(t)) \\
& \text{subject to} \quad (1) \text{ and } (6),
\end{aligned}$$

with

$$\begin{aligned}
\beta_{ji}(t) = & \left( Q_{ji}^{(X)}(t) + Z_{ji}(t) + 2Q_{ji}^{(Y)}(t)Z_{ji}(t) + 2[Z_{ji}(t)]^3 \right) \\
& \times \mathbb{1}_{\{Z_{ji}(t) + \tau R_{ij}^{\max}(t) > \tilde{R}_{ij}(t-1)d_{ji}\}} + Q_{ji}^{(Z)}(t) + Z_{ji}(t), \tag{36}
\end{aligned}$$

$$\begin{aligned} \beta_i^O(t) = & \left( Q_i^{O,(X)}(t) + Q_i^O(t) + 2Q_i^{O,(Y)}(t)Q_i^O(t) + 2[Q_i^O(t)]^3 \right) \\ & \times \mathbb{1}_{\{Q_i^O(t)+A_i(t)>d_i^O\}} + Q_i^{O,(Q)}(t) + Q_i^O(t), \end{aligned} \quad (37)$$

$$\begin{aligned} \beta_i^L(t) = & \left( Q_i^{L,(X)}(t) + Q_i^L(t) + 2Q_i^{L,(Y)}(t)Q_i^L(t) + 2[Q_i^L(t)]^3 \right) \\ & \times \mathbb{1}_{\{Q_i^L(t)+A_i(t)>d_i^L\}} + Q_i^{L,(Q)}(t) + Q_i^L(t). \end{aligned} \quad (38)$$

Note that in **P1**, the UE's allocated transmit power is coupled with the local CPU-cycle frequency. The transmit power also depends on the wireless channel strength to the associated server and the weight  $\beta_{ji}(t)$  of the corresponding offloaded-task queue, in which the former depends on the distance between the UE and server when the latter is related to the MEC server's computation capability and the number of associated UEs. Therefore, the UEs' geographic configuration and the servers' computation capabilities should be taken into account while we associate the UEs with the servers. Moreover, UE-server association, i.e.,  $\eta(n)$ , and resource allocation, i.e.,  $\mathbf{f}(t)$  and  $\mathbf{P}(t)$ , are performed in two different timescales, i.e., in the beginning of each time frame and every time slot afterwards. We solve **P1** in two steps, in which the UE-server association is firstly decided. Then, given the association results, UEs' CPU-cycle frequencies and transmit powers are allocated.

### B. UE-Server Association using Many-to-One Matching with Externalities

To associate UEs to the MEC servers, let us focus on the wireless transmission part of **P1** and, thus, fix  $P_i(t) = P_i^{\max}$  and  $f_i(t) = 0, \forall i \in \mathcal{U}$ , at this stage. The wireless channel gain  $h_{ij}(t)$  and the weight factors  $\beta_i^O(t)$  and  $\beta_{ji}(t)$  dynamically change in each time slot, whereas the UEs are re-associated with the servers in every  $T_0$  slots. In order to take the impacts of  $h_{ij}(t)$ ,  $\beta_i^O(t)$ , and  $\beta_{ji}(t)$  into account, we consider the average strength for the channel gain, i.e., letting  $h_{ij}(t) = \mathbb{E}[h_{ij}]$ ,  $\forall i \in \mathcal{U}, j \in \mathcal{S}$ , and the empirical average, i.e.,

$$\begin{aligned} \tilde{\beta}_i^O(n) &= \frac{1}{(n-1)T_0} \sum_{t=0}^{(n-1)T_0-1} \beta_i^O(t), \quad \forall i \in \mathcal{U}, \\ \tilde{\beta}_j(n) &= \frac{1}{(n-1)T_0} \sum_{k=1}^{n-1} \sum_{t=(k-1)T_0}^{kT_0-1} \sum_{i \in \mathcal{U}} \frac{\eta_{ij}(k)\beta_{ji}(t)}{\sum_{i \in \mathcal{U}} \eta_{ij}(k)}, \quad \forall j \in \mathcal{S}, \end{aligned}$$

as the estimations of the weight factors  $\beta_i^O(t)$  and  $\beta_{ji}(t)$ . Here,  $\tilde{\beta}_j(n)$  represents the estimated average weight factor of a single UE's offloaded-task queue (at server  $j$ ) since we also take the

average over all the associated UEs in each time frame. Incorporating the above assumptions, the UE-server association problem of **P1** can be considered as

$$\begin{aligned} \mathbf{P1-1:} \quad & \underset{\eta(n)}{\text{maximize}} \quad \sum_{i \in \mathcal{U}} \sum_{j \in \mathcal{S}} (\tilde{\beta}_i^O(n) - \tilde{\beta}_j(n)) \log_2 \left( 1 + \frac{\eta_{ij}(n) P_i^{\max} \mathbb{E}[h_{ij}]}{N_0 W + \sum_{i' \in \mathcal{U} \setminus i} \eta_{i'j}(n) P_{i'}^{\max} \mathbb{E}[h_{i'j}]} \right) \\ & \text{subject to} \quad (1). \end{aligned}$$

However, due to the binary nature of optimization variables and the non-convexity of the objective, **P1-1** is an NP-hard nonlinear integer programming problem [33]. To tackle this, we invoke matching theory which provides an efficient and low-complexity way to solve integer programming problems [34], [35] and has been utilized in various wireless communication systems [22], [35]–[38].

Defined in matching theory, *matching* is a bilateral assignment between two sets of players, i.e., the sets of UEs and MEC servers, which matches each player in one set to the player(s) in the opposite set [28]. Referring to the structure of problem **P1-1**, we consider a many-to-one matching model [39] in which each UE  $i \in \mathcal{U}$  is assigned to a server when each server  $j \in \mathcal{S}$  can be assigned to multiple UEs. Moreover, UE  $i$  is assigned to server  $j$  if server  $j$  is assigned to UE  $i$ , and vice versa. The considered many-to-one matching is formally defined as follows.

**Definition 1.** *The considered many-to-one matching game consists of two sets of players, i.e.,  $\mathcal{U}$  and  $\mathcal{S}$ , and the outcome of the many-to-one matching is a function  $\eta$  from  $\mathcal{U} \times \mathcal{S}$  to the set of all subsets of  $\mathcal{U} \times \mathcal{S}$  with*

- 1)  $|\eta(i)| = 1, \forall i \in \mathcal{U}$ ,
- 2)  $|\eta(j)| \leq U, \forall j \in \mathcal{S}$ ,
- 3)  $j = \eta(i) \Leftrightarrow i \in \eta(j)$ .

Having a matching  $\eta$ , the UE-server association indicator can be specified as per,  $\forall i \in \mathcal{U}, j \in \mathcal{S}$ ,

$$\begin{cases} \eta_{ij}(n) = 1, & \text{if } j = \eta(i), \\ \eta_{ij}(n) = 0, & \text{otherwise.} \end{cases} \quad (39)$$

Note that **P1-1** is equivalent to a weighted sum rate maximization problem in which  $(\tilde{\beta}_i^O(n) - \tilde{\beta}_j(n)) \log_2 \left( 1 + \frac{P_i^{\max} \mathbb{E}[h_{ij}]}{N_0 W + \sum_{i' \in \mathcal{U} \setminus i} \eta_{i'j}(n) P_{i'}^{\max} \mathbb{E}[h_{i'j}]} \right)$  can be treated as UE  $i$ 's weighted transmission rate provided that UE  $i$  is matched to server  $j$ . Thus, the UE's matching preference over the servers can be chosen based on the weighted rates in the descending order. Since the rate is affected by

the other UEs (i.e., the players in the same set of the matching game) via interference if they are matched to the same server, the UE's preference also depends on the matching state of the other UEs. The interdependency between the players' preferences is called *externalities* [35] in which the player's preference dynamically changes with the matching state of the other players in the same set. Thus, the UE's preference over matching states should be adopted. To this end, given a matching  $\eta$  with  $j = \eta(i)$ , we define UE  $i$ 's utility as

$$\Psi_i(\eta) = (\tilde{\beta}_i^O - \tilde{\beta}_j) \log_2 \left( 1 + \frac{P_i^{\max} \mathbb{E}[h_{ij}]}{N_0 W + \sum_{i' \in \mathcal{U} \setminus i} \mathbb{1}_{\{\eta(i')=j\}} P_{i'}^{\max} \mathbb{E}[h_{i'j}]} \right), \quad \forall i \in \mathcal{U}, \quad (40)$$

and server  $j$ 's utility as

$$\Psi_j(\eta) = \sum_{i \in \mathcal{U}} (\tilde{\beta}_i^O - \tilde{\beta}_j) \log_2 \left( 1 + \frac{\eta_{ij} P_i^{\max} \mathbb{E}[h_{ij}]}{N_0 W + \sum_{i' \in \mathcal{U} \setminus i} \mathbb{1}_{\{\eta(i')=j\}} P_{i'}^{\max} \mathbb{E}[h_{i'j}]} \right), \quad \forall j \in \mathcal{S}. \quad (41)$$

The UE's and server's matching preferences are based on their own utilities in a descending order. For notational simplicity, we remove the time index  $n$  in (40) and (41). Subsequently, we consider the notion of *swap matching* to deal with externalities [39].

**Definition 2.** Given a many-to-one matching  $\eta$ , a pair of UEs  $(i, i')$ , and a pair of servers  $(j, j')$  with  $j = \eta(i)$  and  $j' = \eta(i')$ , a swap matching  $\eta_{ij}^{i'j'}$  is

$$\eta_{ij}^{i'j'} = \{\eta \setminus \{(i, j), (i', j')\}\} \cup \{(i, j'), (i', j)\}.$$

In other words, we have  $j' = \eta_{ij}^{i'j'}(i)$  and  $j = \eta_{ij}^{i'j'}(i')$  in the swap matching  $\eta_{ij}^{i'j'}$  for the UE pair  $(i, i')$  and server pair  $(j, j')$ , whereas the matching state of the other UEs and servers remains identical in both  $\eta$  and  $\eta_{ij}^{i'j'}$ . Furthermore, in Definition 2, one of the UE pair in the swap operation, e.g.,  $i'$ , can be an open spot of server  $j'$  in  $\eta$  with  $|\eta(j')| < U$ . In this situation, we have  $|\eta_{ij}^{i'j'}(j')| - |\eta(j')| = 1$  and  $|\eta(j)| - |\eta_{ij}^{i'j'}(j)| = 1$ . Moreover,  $\Psi_{i'}(\eta) = 0$  and  $\Psi_{i'}(\eta_{ij}^{i'j'}) = 0$ .

**Definition 3.** For the matching  $\eta$ ,  $(i, i')$  is a swap-blocking pair [35] if and only if

- 1)  $\forall u \in \{i, i', j, j'\}, \Psi_u(\eta_{ij}^{i'j'}) \geq \Psi_u(\eta)$ ,
- 2)  $\exists u \in \{i, i', j, j'\}$  such that  $\Psi_u(\eta_{ij}^{i'j'}) > \Psi_u(\eta)$ .

Therefore, provided that the matching state of the remaining UEs and servers is fixed, two UEs  $(i, i')$  exchange their respectively matched servers if both UEs' and both servers' utilities will not be worse off, and at least one's utility is better off, after the swap. The same criteria are

---

**Algorithm 1** UE-Server Association by Many-to-One Matching with Externalities
 

---

- 1: Initialize  $\eta(i) = \operatorname{argmax}_{j \in \mathcal{S}} \{\mathbb{E}[h_{ij}]\}, \forall i \in \mathcal{U}$ .
  - 2: Calculate (40) and (41).
  - 3: **repeat**
  - 4:     Select a pair of UEs  $(i, i')$  or a UE  $i$  with an open spot  $i'$  of server  $j'$ .
  - 5:     **if**  $(i, i')$  is a swap-blocking pair of the current matching  $\eta$  **then**
  - 6:         Update  $\eta \leftarrow \eta_{ij}^{i'j'}$ .
  - 7:         Calculate (40) and (41).
  - 8:     **end if**
  - 9: **until** No swap-blocking pair exists in the current matching  $\eta$ .
  - 10: Transfer  $\eta$  to the UE-server association indicator as per (39).
- 

applicable when the UE  $i$  changes to another server  $j'$  from the current matched server  $j$ , i.e.,  $i'$  is an open spot of server  $j'$ .

**Definition 4.** A matching  $\eta$  is two-sided exchange-stable if there is no swap-blocking pair [39].

In summary, we first initialize a matching  $\eta$  and calculate utilities (40) and (41). Then, we iteratively find a swap-block pair and update the swap matching until two-sided exchange stability is achieved. The steps to solve problem **P1-1** by many-to-one matching with externalities are detailed in Algorithm 1. In each iteration of Algorithm 1, there are  $\binom{U}{2} + U(S-1)$  possibilities to form a swap-blocking pair, in which there are  $\binom{U}{2}$  pairs of two different UEs. The term  $U(S-1)$  accounts for all the swap-blocking pairs consisting of a UE  $i \in \mathcal{U}$  and an open spot of another MEC server  $j' \in \mathcal{S} \setminus j$ . Given that the UEs are more than the MEC servers in our considered network, the complexity of Algorithm 1 is in the order of  $\mathcal{O}(U^2)$  which increases binomially with the number of UEs. Additionally, let us consider an exhaustive search in problem **P1-1**. Since each UE  $i \in \mathcal{U}$  can only access one out of  $S$  servers, there are  $S^U$  association choices satisfying constraint (1). In the exhaustive search approach, the complexity increases exponentially with the number of UEs.

### C. Resource Allocation and Task Splitting at the UE Side

Now denoting the representative UE  $i$ 's associated MEC server as  $j^*$ , we formulate the UEs' resource allocation problem of **P1** as

$$\begin{aligned} \mathbf{P1-2}: \quad & \underset{\mathbf{f}(t), \mathbf{P}(t)}{\text{minimize}} \quad \sum_{i \in \mathcal{U}} (\beta_{j^*i}(t) - \beta_i^O(t)) \tau W \log_2 \left( 1 + \frac{P_i(t) h_{ij^*}(t)}{N_0 W + \sum_{i' \in \mathcal{U} \setminus i} \eta_{i'j^*} P_{i'}(t) h_{i'j^*}(t)} \right) \\ & - \sum_{i \in \mathcal{U}} \frac{\beta_i^L(t) \tau f_i(t)}{L_i} + \sum_{i \in \mathcal{U}} V (\kappa [f_i(t)]^3 + P_i(t)) \\ & \text{subject to (6).} \end{aligned}$$

When solving problem **P1-2** in a centralized manner, each UE  $i$  needs to upload its local information  $\beta_i^L(t)$  and  $\beta_i^O(t)$  in every time slot to a central unit, e.g., the associated server. This can incur high overheads, especially in dense networks. In order to alleviate this issue, we decompose the summation (over all UEs) in the objective and let each UE  $i \in \mathcal{U}$  locally allocate its CPU-cycle frequency and transmit power by solving

$$\begin{aligned} \mathbf{P1-2'}: \quad & \underset{f_i(t), P_i(t)}{\text{minimize}} \quad (\beta_{j^*i}(t) - \beta_i^O(t)) \tau W \times \mathbb{E}_{I_{ij^*}} \left[ \log_2 \left( 1 + \frac{P_i(t) h_{ij^*}(t)}{N_0 W + I_{ij^*}(t)} \right) \right] \\ & - \frac{\beta_i^L(t) \tau f_i(t)}{L_i} + V (\kappa [f_i(t)]^3 + P_i(t)) \\ & \text{subject to (6).} \end{aligned}$$

The expectation is with respect to the locally-estimated distribution,  $\hat{\text{Pr}}(I_{ij^*}; t)$ , of the aggregate interference  $I_{ij^*}(t) = \sum_{i' \in \mathcal{U} \setminus i} \eta_{i'j^*} P_{i'}(t) h_{i'j^*}(t)$ . Note that when  $V = 0$ , **P1-2'** is equivalent to the rate maximization problem, where the power budget will be fully allocated. As  $V$  is gradually increased, we pay more attention on power cost reduction, and the solution values will decrease correspondingly. Moreover, although the downlink is not considered in this work, we implicitly assume that the UE has those executed tasks and can locally track the state of the offloaded-task queue. In other words, the full information about  $\beta_{j^*i}(t)$  is available at the UE.

**Lemma 1.** *The optimal solution to problem **P1-2'** is that UE  $i$  allocates the CPU-cycle frequency*

$$f_i^*(t) = \sqrt{\frac{\beta_i^L(t) \tau}{3L_i \kappa (V + \gamma^*)}}$$

*for local computation. The optimal allocated transmit power  $P_i^* > 0$  satisfies*

$$\mathbb{E}_{I_{ij^*}} \left[ \frac{(\beta_i^O(t) - \beta_{j^*i}(t)) \tau W h_{ij^*}}{(N_0 W + I_{ij^*} + P_i^* h_{ij^*}) \ln 2} \right] = V + \gamma^*$$



if  $\mathbb{E}_{I_{ij}^*} \left[ \frac{(\beta_i^O(t) - \beta_{j^*i}(t))\tau W h_{ij^*}}{(N_0 W + I_{ij^*}) \ln 2} \right] > V + \gamma^*$ . Otherwise,  $P_i^* = 0$ . Furthermore, the optimal Lagrange multiplier  $\gamma^*$  is 0 if  $\kappa[f_i^*(t)]^3 + P_i^*(t) < P_i^{\max}$ . When  $\gamma^* > 0$ ,  $\kappa[f_i^*(t)]^3 + P_i^*(t) = P_i^{\max}$ .

*Proof.* Please refer to Appendix B.  $\square$

The second decomposed problem **P2** given in (35) determines whether the arrival task is assigned to the local-computation queue or task-offloading queue. In this regard, each UE  $i \in \mathcal{U}$  solves

$$\begin{aligned} \mathbf{P2}: \quad & \underset{A_i^L(t), A_i^O(t)}{\text{minimize}} \quad \beta_i^L(t) A_i^L(t) + \beta_i^O(t) A_i^O(t) \\ & \text{subject to} \quad A_i(t) = A_i^L(t) + A_i^O(t), \\ & \quad A_i^L(t), A_i^O(t) \in \{0, A_{\text{unit}}, 2A_{\text{unit}}, \dots\}. \end{aligned}$$

We can straightforwardly find an optimal solution  $(A_i^{L*}(t), A_i^{O*}(t))$  as

$$\begin{cases} (A_i^{L*}(t), A_i^{O*}(t)) = (A_i(t), 0), & \text{if } \beta_i^L(t) \leq \beta_i^O(t), \\ (A_i^{L*}(t), A_i^{O*}(t)) = (0, A_i(t)), & \text{if } \beta_i^L(t) > \beta_i^O(t). \end{cases} \quad (42)$$

#### D. Computational Resource Scheduling at the Server Side

The third decomposed problem **P3** given in (35) is addressed at the server side, where each MEC server  $j \in \mathcal{S}$  solves

$$\begin{aligned} \mathbf{P3}: \quad & \underset{\mathbf{f}_j(t)}{\text{maximize}} \quad \sum_{i \in \mathcal{U}} \frac{\beta_{ji}(t) f_{ji}(t)}{L_i} \\ & \text{subject to} \quad (9), \end{aligned}$$

to schedule its CPU cores. Note that the server considers the allocated CPU-cycle frequencies to all UEs in the objective, constraints, and optimization variables even though some UEs are not associated in the current time frame  $n$ . However, since the corresponding weight of the UE's offloaded-task queue, i.e.,  $\beta_{ji}(t)$ , is taken into account in the objective, the resource allocation at the server will not be over-provisioned. These insights are illustrated as follows.

Assuming that the server is not able to complete a UE's offloaded tasks in the previous time frame  $n-1$  and is not associated with this UE in the current time frame  $n$ . Although the offloaded-task queue length  $Z_{ji}(t)$  does not grow anymore, those incomplete tasks will experience severe delay if they are ignored in the current time frame. Furthermore, since  $\tilde{R}_{ij}(t)$  that is considered in the URLLC constraints decreases in the current frame, the weight  $\beta_{ji}(t)$  grows as per (26),

---

**Algorithm 2** Computational Resource Scheduling at the Server

---

- 1: Initialize  $n = 1$ ,  $\mathcal{U}_j = \{i \in \mathcal{U} | \beta_{ji}(t) > 0\}$ , and  $f_{ji} = 0, \forall i \in \mathcal{U}$ .
  - 2: **while**  $n \leq N_j$  and  $\mathcal{U}_j \neq \emptyset$  **do**
  - 3:     Find  $i^* = \operatorname{argmax}_{i \in \mathcal{U}_j} \{\beta_{ji}(t)/L_i\}$ .
  - 4:     Let  $f_{ji^*}(t) = f_j^{\max}$ .
  - 5:     Update  $n \leftarrow n + 1$  and  $\mathcal{U}_j \leftarrow \mathcal{U}_j \setminus i^*$ .
  - 6: **end while**
- 

---

**Algorithm 3** Two-Timescale Mechanism for UE-Server Association, Task Offloading, and Resource Allocation

---

- 1: Initialize  $t = 0$ , predetermine the system lifetime as  $T$ , and set the initial queue values of (3), (4), (7), and (22)–(27) and (31)–(33) as zero.
  - 2: **repeat**
  - 3:     **if**  $t/T_0 \in \mathbb{N}$  **then**
  - 4:         Run Algorithm 1 to associate the UEs with the MEC servers.
  - 5:     **end if**
  - 6:     The UE allocates the local CPU-cycle frequency and transmit power as per Lemma 1, and splits the task arrivals for local computation and offloading according to (42).
  - 7:     The server schedules its computational resources by following Algorithm 2.
  - 8:     The UE updates the queue lengths in (3), (4), (22)–(25), (31), and (32).
  - 9:     The server updates the queue lengths in (7), (26), (27), and (33).
  - 10:     Update  $t \leftarrow t + 1$ .
  - 11: **until**  $t > T$
- 

(27), (33), and (36). In other words, the more severe the experienced delay of the ignored UE's incomplete tasks, the higher the weight  $\beta_{ji}(t)$ . To address this severe-delay issue, the server takes into account all UEs via  $\beta_{ji}(t)$  in the objective. Once the offloaded tasks are completed,  $\beta_{ji}(t)$  remains zero in the rest time slots. In addition, for the UEs which have not been associated with this server, we have  $\beta_{ji}(t) = 0$ . Therefore, the server only considers the UEs with  $\beta_{ji}(t) > 0$  while scheduling the computational resources. In summary, the optimal solution to problem **P3** is that server  $j$  dedicates its CPU cores to, at most,  $N_j$  UEs with largest positive values of  $b_{ji}(t)/L_i$ . The steps of allocating the server's CPU cores are detailed in Algorithm 2.

Table I  
SIMULATION PARAMETERS [7], [10], [17], [40], [41]

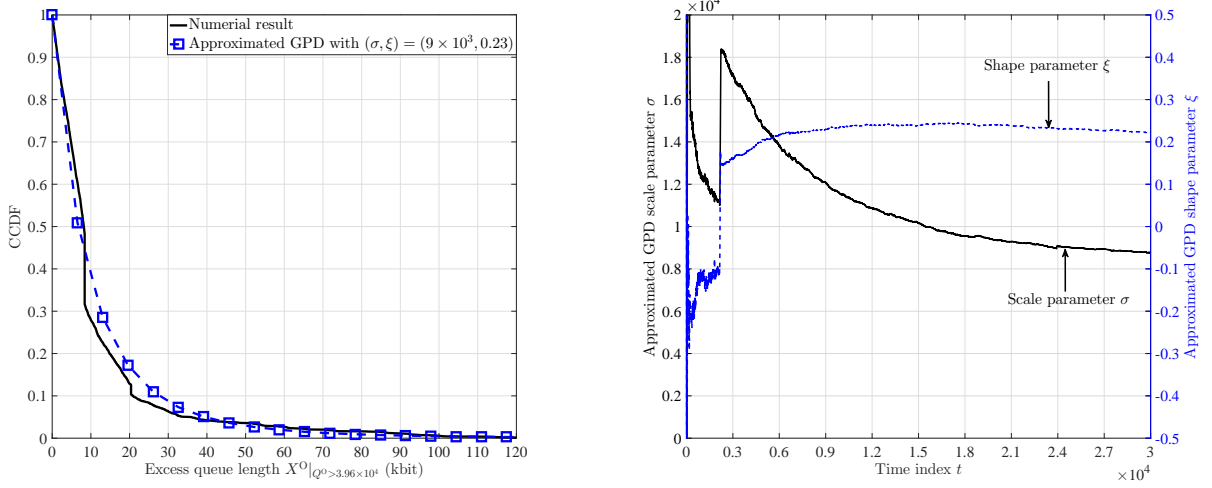
Parameter	Value	Parameter	Value
$T_0$	100	$A_{\text{unit}}$	1500 bytes
$\kappa$	$10^{-27} \text{ Watt} \cdot \text{s}^3/\text{cycle}^3$	$W$	10 MHz
$N_0$	-174 dBm/Hz	$P_i^{\max}$	30 dBm
$L_i$	$[1 \times 10^3, 4 \times 10^4] \text{ cycle/bit}$	$V$	0
$\lambda_i$	[10, 150] kbps	$f_j^{\max}$	$10^{10} \text{ cycle/s}$
$d_i^{\text{L}}$	$100\tilde{A}_i^{\text{L}}(t-1) \text{ (bit)}$	$\epsilon_i^{\text{L}}$	0.01
$\sigma_i^{\text{L,th}}$	40 (Mbit)	$\xi_i^{\text{L,th}}$	0.3
$d_i^{\text{O}}$	$100\tilde{A}_i^{\text{O}}(t-1) \text{ (bit)}$	$\epsilon_i^{\text{O}}$	0.01
$\sigma_i^{\text{O,th}}$	40 (Mbit)	$\xi_i^{\text{O,th}}$	0.3
$d_{ji}$	20 sec	$\epsilon_{ji}$	0.01
$\sigma_{ji}^{\text{th}}$	40 (Mbit)	$\xi_{ji}^{\text{th}}$	0.3

After computing and offloading the tasks in time slot  $t$ , each UE updates its physical and virtual queue lengths in (3), (4), (22)–(25), (31), and (32) when the MEC servers update (7), (26), (27), and (33) for the next slot  $t+1$ . Moreover, based on the transmission rate  $R_{ij}(t)$ , the UE empirically estimates the statistics of  $I_{ij}$  for slot  $t+1$  as per  $\hat{\text{Pr}}(\tilde{I}_{ij}; t+1) = \frac{\mathbb{1}_{\{I_{ij}(t)=\tilde{I}_{ij}\}}}{t+2} + \frac{(t+1)\hat{\text{Pr}}(\tilde{I}_{ij}; t)}{t+2}$ . The procedures of the proposed two-timescale mechanism are outlined in Algorithm 3.

## V. NUMERICAL RESULTS

We consider an indoor  $100 \times 100 \text{ m}^2$  area in which four MEC servers are deployed at the centers of four equal-size quadrants, respectively. Each server is equipped with eight CPU cores. Additionally, multiple UEs, ranging from 30 to 80, are randomly distributed. For task offloading, we assume that the transmission frequency is 5.8 GHz with the path loss model  $24 \log x + 20 \log 5.8 + 60$  (dB) [42], where  $x$  in meters is the distance between any UE and server. Further, all wireless channels experience Rayleigh fading with unit variance. Coherence time is 40 ms [43]. Moreover, we consider Poisson processes for task arrivals. The rest parameters for all UEs  $i \in \mathcal{U}$  and servers  $j \in \mathcal{S}$  are listed in Table I.<sup>4</sup>

<sup>4</sup> $\tilde{A}_i^{\text{L}}(t-1) = \frac{1}{t} \sum_{\tau=0}^{t-1} A_i^{\text{L}}(\tau)$  and  $\tilde{A}_i^{\text{O}}(t-1) = \frac{1}{t} \sum_{\tau=0}^{t-1} A_i^{\text{O}}(\tau)$ .



(a) Tail distributions of the excess value of a UE's task-offloading queue and the approximated GPD of exceedances. (b) Convergence of the approximated GPD scale and shape parameters of exceedances.

Figure 3. Effectiveness of applying the Pickands–Balkema–de Haan theorem in the MEC network.

We first verify the effectiveness of using the Pickands–Balkema–de Haan theorem to characterize the excess queue length in Fig. 3. Although the task-offloading queue is used to verify the results, the local-computation and offloaded-task queues can be used. Let us consider  $L_i = 8250$  cycle/bit,  $\lambda_i = 100$  kbps, and  $d_i^O = 10\tilde{A}_i^O(t - 1)$  bit for all 30 UEs. Then, given  $\Pr(Q^O > 10\tilde{A}^O(\infty)) = 3.4 \times 10^{-3}$  with  $10\tilde{A}^O(\infty) = 3.96 \times 10^4$ , Fig. 3(a) shows the CCDFs of exceedances  $X^O|_{Q^O > 3.96 \times 10^4} = Q^O - 3.96 \times 10^4 > 0$  and the approximated GPD in which the latter provides a good characterization for exceedances. Further, the convergence of the scale and shape parameters of the approximated GPD is shown in Fig. 3(b). Once convergence is achieved, characterizing the statistics of exceedances helps to locally estimate the network-wide extreme metrics, e.g., the maximal queue length among all UEs as in [44], and enables us to proactively deal with the occurrence of extreme events.

Subsequently, we measure the ratio between the task amounts split to the task-offloading queue and local-computation queues, i.e.,  $\zeta = \frac{\tilde{A}_i^O(\infty)}{\tilde{A}_i^L(\infty)}$ . If  $\zeta < 1$ , the UE pays more attention to local computation. More tasks are offloaded when  $\zeta > 1$ . As shown in Fig. 4, more fraction of arrival tasks is offloaded for the intenser processing density  $L$  or higher task arrival rate  $\lambda$ . In these situations, the UE's computation capability becomes less supportable, and extra computational resources are required as expected. Additionally, since stronger interference is

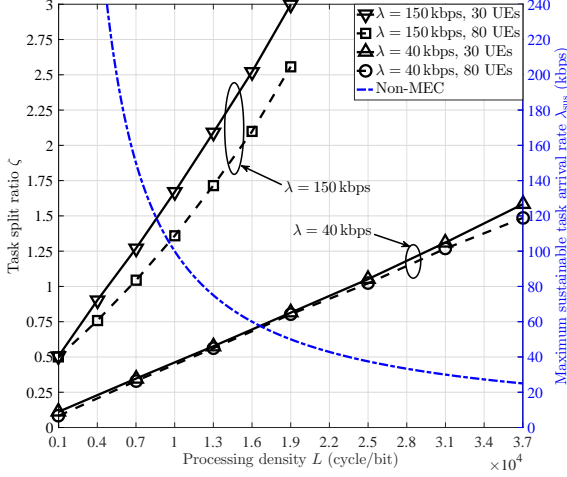


Figure 4. Task split ratio versus processing density.

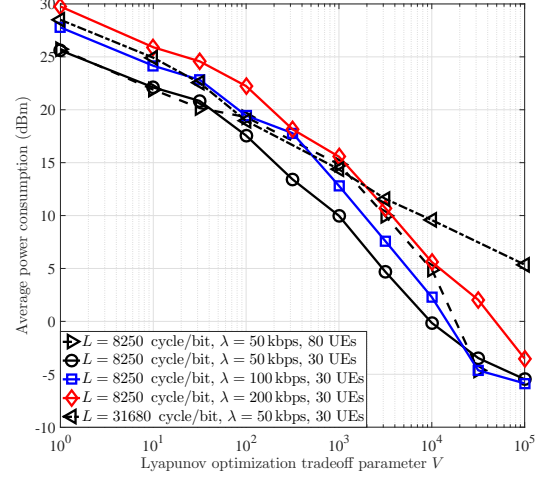


Figure 5. Average power consumption versus Lyapunov optimization tradeoff parameter.

incurred in denser networks, the UE lowers the task portion of offloading, especially when more computational resources of the server are required, i.e., the cases of higher  $L$  and  $\lambda$ . In the scenario without MEC servers, the local computation rate cannot be less than the task arrival rate to maintain queue stability, i.e.,  $10^9/L \geq \lambda$ . In this regard, we also plot the curve of the maximum sustainable task arrival rate  $\lambda_{\text{sus}} = 10^9/L$  without MEC servers in Fig. 4. Comparing the curves of the MEC schemes with the maximum sustainable task arrival rate, we can find

$$\begin{cases} \zeta > 1, & \text{when } \lambda > 10^9/L, \\ \zeta = 1, & \text{when } \lambda = 10^9/L, \\ \zeta < 1, & \text{otherwise.} \end{cases}$$

That is,  $\lambda_{\text{sus}}$  is the watershed between task offloading and local computation. More than 50% of arrival tasks are offloaded if the task arrival rate is higher than  $\lambda_{\text{sus}}$ .

Varying the Lyapunov tradeoff parameter  $V$ , we show the corresponding average power consumption in Fig. 5 and further break down the power cost in Fig. 6. As realized from the objective function of problem **P1-2'**, total power consumption is reduced with increasing  $V$  in all network settings, and the minimal power cost can be asymptotically approached. For any given total power consumption of Fig. 5, we investigate, in Fig. 6, the corresponding task split ratio and the average power consumed by local computation. When less power is consumed, more tasks are assigned to the task-offloading queue. In other words, if the UE is equipped with

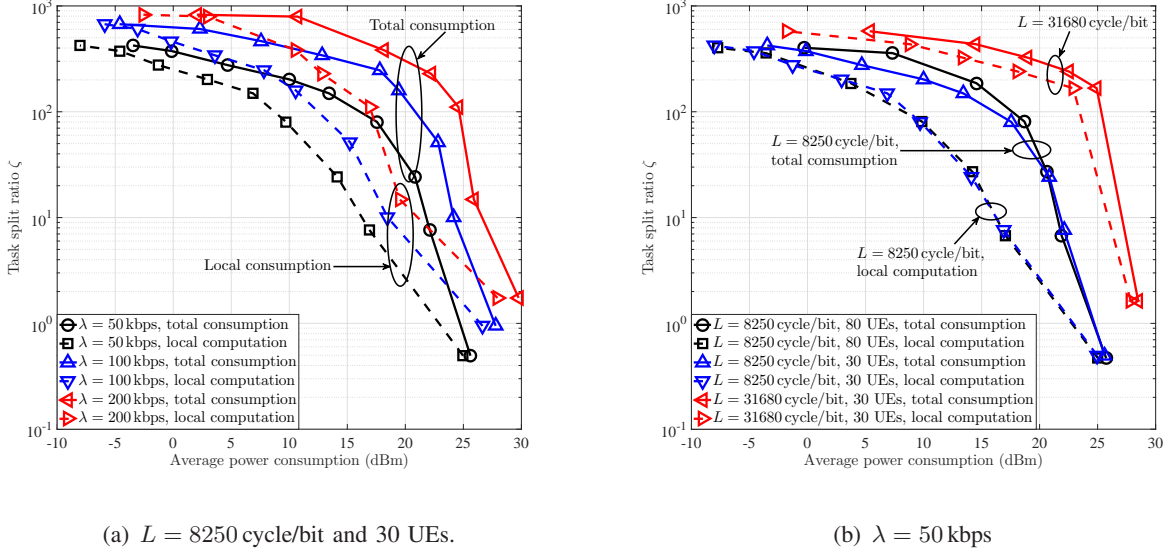
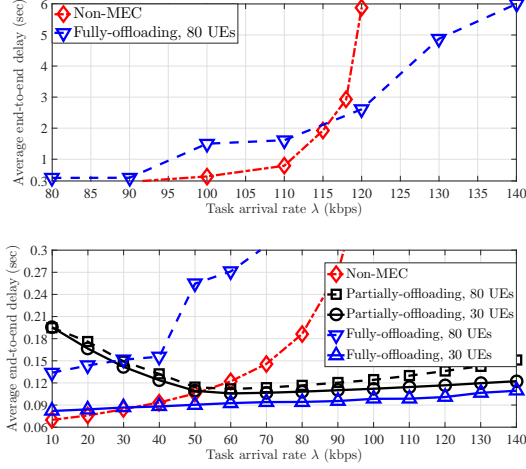


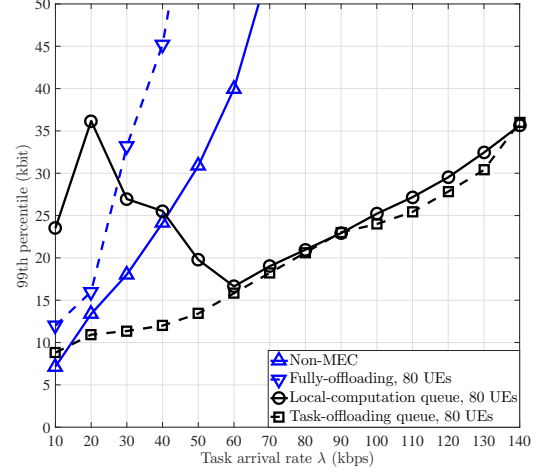
Figure 6. Task split ratio versus average power costs of local computation and total consumption.

weak computation capability or has lower power budget, offloading most tasks (i.e., increasing  $\zeta$ ) helps to meet the URLLC requirements. Given that the same fraction of arrival tasks with a specific processing density  $L$  is computed locally, the same portion of power will be consumed in local computation regardless of the arrival rate  $\lambda$ . In this regard, as shown in Fig. 6(a), the power consumption gap between local computation and total consumption is around 5 dB at  $\zeta = 10$  for different values of  $\lambda$ . However, computation-intensive tasks require higher CPU cycles in local computation. Comparing the curves in Figs. 6(a) and 6(b), we can find that the gap is smaller with a larger  $L$ . Moreover, since higher rates and more intense processing densities demand more resources for task execution, the UE consumes more power when these two parameters  $L$  and  $\lambda$  increase as expected. Additionally, for the specific values of  $L$ ,  $\lambda$  and  $\zeta$ , the locally-executed task amounts and required computational resources in local computation are identical for different numbers of UEs. Thus, as shown in Fig. 6(b), the local power consumption is the same regardless of the network density, but each UE consumes more transmit power (and total consumption) due to stronger transmission interference in the denser network.

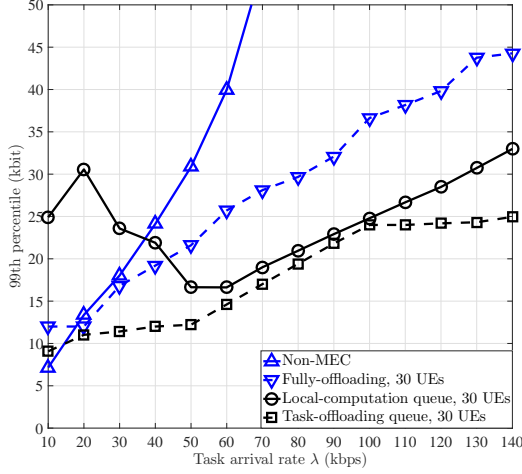
In addition to the discussed MEC architecture of this work, we consider another two baselines for performance comparison: (i) a baseline with no MEC servers for offloading, and (ii) a baseline that offloads all the tasks to the MEC server due to the absence of local computation capability. In the following part, we compare the performance of the proposed method with these two



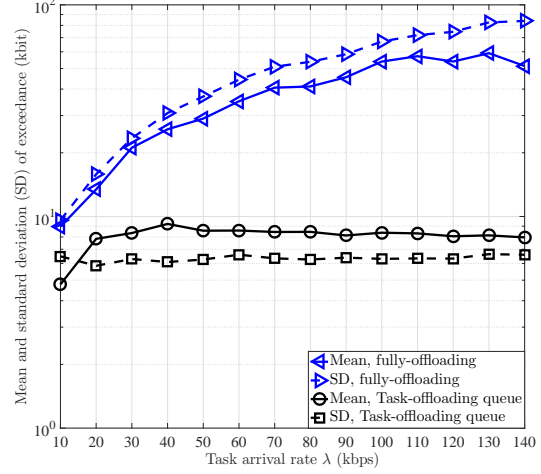
(a) Average end-to-end delay.



(b) 99th percentile of the UE's queue length with 80 UEs.



(c) 99th percentile of the UE's queue length with 30 UEs.

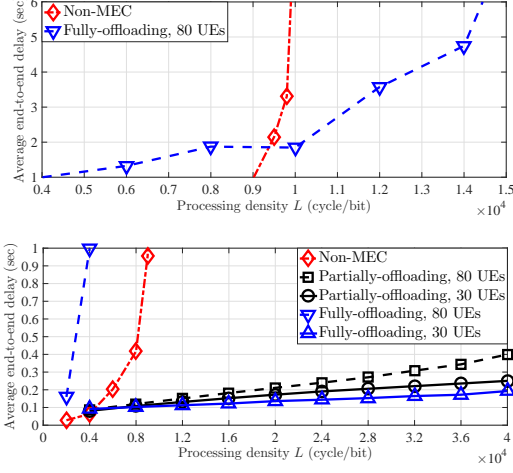


(d) Mean and standard deviation of exceedances with 30 UEs.

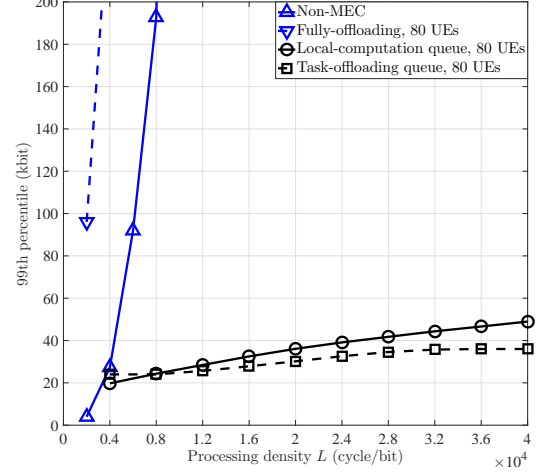
Figure 7. 1) Average end-to-end delay, 2) 99th percentile of the UE's queue length, and 3) mean and standard deviation of exceedances over the 99th percentile queue length, versus task arrival rate with  $L = 8250$  cycle/bit.

baselines for various task arrival rates in Fig. 7 and various processing densities in Fig. 8. We first compare the average end-to-end delay in Figs. 7(a) and 8(a). For the very small arrival rate and processing density requirement, the UE's computation capability is sufficient to execute tasks rapidly, whereas transmission delay and the extra queuing delay incurred at the server degrade the performance in the fully-offloading and partially-offloading schemes. While increasing  $L$  or  $\lambda$ , the UE's computation capability becomes less supportable as mentioned previously. The two (fully- and partially-) offloading schemes will eventually provide better delay performance

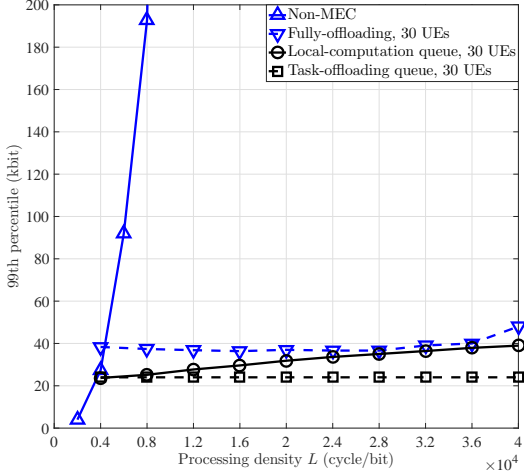




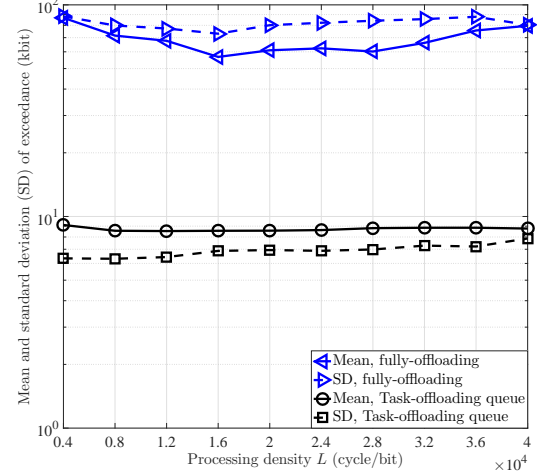
(a) Average end-to-end delay.



(b) 99th percentile of the UE's queue length with 80 UEs.



(c) 99th percentile of the UE's queue length with 30 UEs.



(d) Mean and standard deviation of exceedances with 30 UEs.

Figure 8. 1) Average end-to-end delay, 2) 99th percentile of the UE's queue length, and 3) mean and standard deviation of exceedances over the 99th percentile queue length, versus processing density with  $\lambda = 100$  kbps.

in various network settings. Additionally, for the offloading schemes, the average end-to-end delay is larger in the denser network due to stronger interference and longer waiting time for the server's computational resources. Compared with the fully-offloading schemes, our approach has a remarkable delay performance improvement in the denser network since the fully-offloaded tasks incur tremendous queuing delay. In addition to the average end-to-end delay, we also show the 99th percentile of the UE's queue length, i.e.,  $q_{99} := F_Q^{-1}(0.99)$ , as a reliability measure in Figs. 7(b), 7(c), 8(b), and 8(c). When the proposed approach outperforms the non-MEC scheme

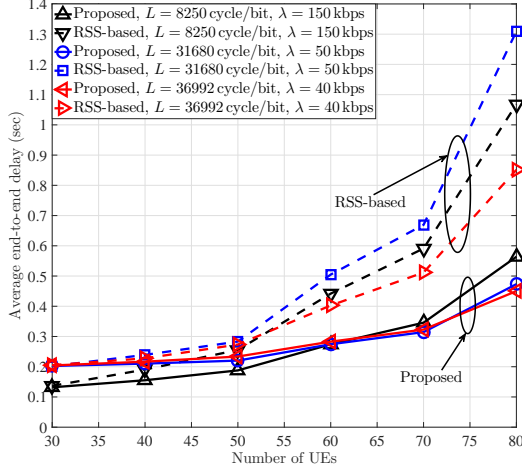
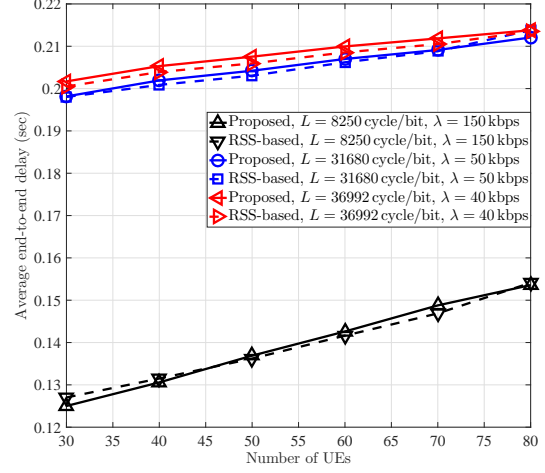
(a) Four MEC servers with  $\{2, 4, 8, 16\}$  CPU cores.(b) Four MEC servers with  $\{8, 8, 8, 8\}$  CPU cores.

Figure 9. Average end-to-end delay versus number of UEs for different UE-server association schemes.

in terms of the average end-to-end delay, the local-computation queue has a lower 99th percentile than the 99th percentile queue length of the non-MEC scheme. Similar results can be found for the task-offloading queue and the queue of the fully-offloading scheme if our proposed approach has a lower average end-to-end delay. In the 30-UE case, although our proposed approach has a higher average end-to-end delay than the fully-offloading scheme, splitting the tasks decreases the loading of each queue buffer and results the lower queue length. Let us zoom in on the excess queue value over the 99th percentile in the 30-UE case, the mean and standard deviation of exceedances, i.e.,  $\mathbb{E}[Q - q_{99} | Q > q_{99}]$  and  $\sqrt{\text{Var}(Q - q_{99} | Q > q_{99})}$ , are investigated in Figs. 7(d) and 8(d), where we can find that our approach has a smaller amount and more concentrated extent of the extreme events. Although fully offloading tasks achieves lower delay in the sparse network, the partially-offloading scheme can lower the loading in the queue buffer. In practice, if the queue buffer is equipped with the finite-size storage, our proposed approach can properly address the potential overflowing issue and achieve more reliable task computation.

Finally in Figs. 9 and 10, we show the advantage of our proposed UE-server association approach in the heterogeneous MEC architecture. As a baseline, we consider the mechanism in which the UE accesses the MEC server with the highest RSS. Provided that the deployed MEC servers have different computation capabilities (in terms of CPU cores), associating some UEs with the stronger-capability server but the lower RSS can properly balance the servers' loadings

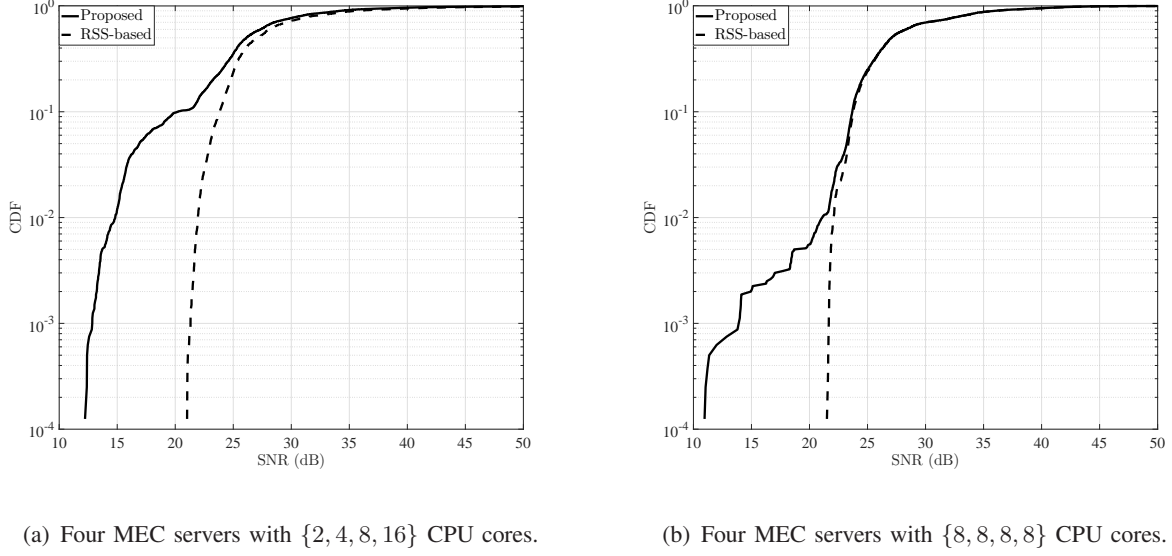


Figure 10. CDF of the wireless channel SNR between the UE and the associated MEC server for different UE-server association schemes with  $L = 8250$  cycle/bit,  $\lambda = 150$  kbps, and 80 UEs.

although the transmission rates are sacrificed. As a result, compared with the baseline, the waiting time for the server's computational resources and the end-to-end delay of the proposed association approach are alleviated. The advantage is more prominent in the dense network since there are more UEs waiting for the server's resources. If the servers' computation capabilities are identical, associating the UE with the server with the lower RSS does not give a computation gain. Thus, the proposed approach and baseline have identical association outcome and delay performance irrespective of the network setting. We further show the CDF of the wireless channel signal-to-noise ratio (SNR), measured as  $\frac{P_i^{\max} \mathbb{E}[h_{ij}]}{N_0 W}$ , between the UE and its associated server in Fig. 10. When the MEC servers are homogeneous, the proposed approach and baseline have similar association results as mentioned above. This is verified by the 1 dB gap at the 1st percentile of the SNR. Nevertheless, in the heterogeneous MEC architecture, the gap is 7 dB at the 1st percentile of the SNR and reduces to 1 dB until the 30th percentile.

## VI. CONCLUSIONS

The goal of this work is to enable a URLLC design for the MEC network with multiple UEs and servers. In this regard, the URLLC requirement has been formulated with respect to the threshold deviation probability of the task queue length. By leveraging extreme value theory, we have characterized the statistics of the threshold deviation event with a low occurrence

probability and imposed another URLLC constraint on the high-order statistics. The studied problem has been cast as UEs' computation and communication power minimization subject to the URLLC constraints. Furthermore, incorporating techniques from Lyapunov stochastic optimization and matching theory, we have proposed a two-timescale framework for UE-server association, task offloading, and resource allocation. UE-server association is formulated as a many-to-one matching game with externalities which is addressed, in the long timescale, via the notion of swap matching. In every time slot, each UE allocates the computation and communication resources, and splits the task arrivals for local computation and offloading. In the meantime, each server schedules its multiple CPU cores to compute the UEs' offloaded tasks. Numerical results have shown the effectiveness of characterizing the extreme queue length by extreme value theory. Partially offloading tasks provides more reliable task computation in contrast with the non-MEC and fully-offloading schemes. When the servers have different computation capabilities, the proposed UE-server association approach achieves lower delay than the RSS-based approach, particularly in denser networks.

## APPENDIX A

Table II  
SUMMARY OF NOTATIONS

Notation	Definition	Notation	Definition	Notation	Definition
$A_i$	UE $i$ 's task arrivals	$A_i^L$	Split tasks for local computation	$A_i^O$	Split tasks for offloading
$A_{\text{unit}}$	Unit task	$\tilde{A}_i^L$	Moving time-averaged value of $A_i^L$	$\tilde{A}_i^O$	Moving time-averaged value of $A_i^O$
$d_i^L$	Queue length bound for $Q_i^L$	$d_i^O$	Queue length bound for $Q_i^O$	$f_i$	UE $i$ 's CPU-cycle frequency
$f_{ji}$	Server $j$ 's CPU-cycle frequency for UE $i$	$f_j^{\max}$	Server $j$ 's computation capability per CPU core	$\mathbf{f}$	Network-wide computation frequency vector
$\mathbf{f}_j$	Server $j$ 's computation frequency vector	$h_{ij}$	Channel gain between UE $i$ to server $j$	$I_{ij}$	Aggregate interference to $h_{ij}$
$L_i$	UE $i$ 's required processing density	$\mathcal{L}$	Lyapunov function	$n$	Time frame index
$N_0$	Power spectral density of AWGN	$N_j$	Number of server $j$ 's CPU cores	$P_i$	UE $i$ 's transmit power

$P_{\max}$	UE $i$ 's power budget	$\bar{P}_i^C$	UE $i$ 's long-term time-averaged computation power	$\bar{P}_i^T$	UE $i$ 's long-term time-averaged transmit power
$\mathbf{P}$	Network-wide transmit power vector	$\hat{\mathbf{P}}_r(I_{ij})$	Estimated distribution of $I_{ij}$	$Q_i^L$	UE $i$ 's local-computation queue length
$Q_i^O$	UE $i$ 's task-offloading queue length	$\mathbf{Q}$	Combined queue vector	$R_{ij}$	Transmission rate from UE $i$ to server $j$
$\tilde{R}_{ij}$	Moving time-averaged value of $R_{ij}$	$R_{ij}^{\max}$	Maximum offloading rate of $R_{ij}$	$\mathcal{S}$	Set of servers
$S$	Number of servers	$t$	Time slot index	$T_0$	Time frame length
$\mathcal{T}$	Time frame	$\mathcal{U}$	Set of UEs	$U$	Number of UEs
$V$	Lyapunov optimization parameter	$W$	Server's bandwidth	$X_i^L$	Excess value of $Q_i^L$
$\bar{X}_i^L$	Long-term time-averaged conditional expectation of $X_i^L$	$X_i^O$	Excess value of $Q_i^O$	$\bar{X}_i^O$	Long-term time-averaged conditional expectation of $X_i^O$
$X_{ji}$	Excess value of $Z_{ji}$	$\bar{X}_{ji}$	Long-term time-averaged conditional expectation of $X_{ji}$	$Y_i^L$	Square of $X_i^L$
$\bar{Y}_i^L$	Long-term time-averaged conditional expectation of $Y_i^L$	$Y_i^O$	Square of $X_i^O$	$\bar{Y}_i^O$	Long-term time-averaged conditional expectation of $Y_i^O$
$Y_{ji}$	Square of $X_{ji}$	$\bar{Y}_{ji}$	Long-term time-averaged conditional expectation of $Y_{ji}$	$Z_{ji}$	Server $i$ 's offloaded-task queue for UE $i$
$\beta_i^L$	Related weight of $Q_i^L$	$\beta_i^O$	Related weight of $Q_i^O$	$\beta_{ij}$	Related weight of $Z_{ji}$
$\tilde{\beta}_i^L$	Estimated average of $\beta_i^L$	$\tilde{\beta}_i^O$	Estimated average of $\beta_i^O$	$\tilde{\beta}_{ij}$	Estimated average of $\beta_{ji}$
$\eta_{ij}$	Association indicator between UE $i$ and server $j$	$\boldsymbol{\eta}$	Network-wide association vector	$\epsilon_i^L$	Tolerable bound violation probability for $Q_i^L$
$\epsilon_i^O$	Tolerable bound violation probability for $Q_i^O$	$\epsilon_{ji}$	Tolerable bound violation probability for $Z_{ji}$	$\kappa$	Computation power parameter
$\lambda_i$	UE $i$ 's average task arrival rate	$\Psi_i(\eta)$	UE $i$ 's utility under a matching $\eta$	$\Psi_i(\eta)$	Server $j$ 's utility under a matching $\eta$
$\sigma_i^L$	GPD scale parameter of $X_i^L$	$\sigma_i^{L,th}$	Threshold for $\sigma_i^L$	$\sigma_i^{O,th}$	Threshold for the GPD scale parameter of $X_i^O$
$\sigma_{ji}^{th}$	Threshold for the GPD scale parameter of $X_{ji}$	$\tau$	Time slot length	$\xi_i^L$	GPD shape parameter of $X_i^L$
$\xi_i^{L,th}$	Threshold for $\xi_i^L$	$\xi_i^{O,th}$	Threshold for the GPD shape parameter of $X_i^O$	$\xi_{ji}^{th}$	Threshold on the GPD shape parameter of $X_{ji}$

APPENDIX B  
PROOF OF LEMMA 1

We first express the Lagrangian of problem **P1-2'** as

$$\begin{aligned} L(f_i(t), P_i(t), \gamma, \alpha_1, \alpha_2) = & (\beta_{j^*i}(t) - \beta_i^O(t))\tau W \times \mathbb{E}_{I_{ij^*}} \left[ \log_2 \left( 1 + \frac{P_i(t)h_{ij^*}(t)}{N_0W + I_{ij^*}(t)} \right) \right] \\ & - \frac{\beta_i^L(t)\tau f_i(t)}{L_i} + V(\kappa[f_i(t)]^3 + P_i(t)) + \gamma(\kappa[f_i(t)]^3 + P_i(t) - P_i^{\max}) - \alpha_1 f_i(t) - \alpha_2 P_i(t), \end{aligned} \quad (43)$$

where  $\gamma$ ,  $\alpha_1$ , and  $\alpha_2$  are the Lagrange multipliers. Taking the partial differentiations of (43) with respect to  $f_i$  and  $P_i$ , we have

$$\begin{aligned} \frac{\partial}{\partial f_i} L &= -\frac{\beta_i^L(t)\tau}{L_i} + 3\kappa(V + \gamma)[f_i(t)]^2 - \alpha_1, \\ \frac{\partial}{\partial P_i} L &= \mathbb{E}_{I_{ij^*}} \left[ \frac{(\beta_{j^*i}(t) - \beta_i^O(t))\tau W h_{ij^*}}{(N_0W + I_{ij^*} + P_i h_{ij^*}) \ln 2} \right] + V + \gamma - \alpha_2. \end{aligned}$$

Subsequently, since **P1-2'** is a convex optimization problem, we apply the Karush–Kuhn–Tucker (KKT) conditions to derive the optimal solution in which the optimal CPU-cycle  $f_i^*(t)$ , optimal transmit power  $P_i^*(t)$ , and optimal Lagrange multipliers (i.e.,  $\gamma^*$ ,  $\alpha_1^*$ , and  $\alpha_2^*$ ) satisfy

$$f_i^*(t) = \sqrt{\frac{\alpha_1^* + \frac{\beta_i^L(t)\tau}{L_i}}{3\kappa(V + \gamma^*)}}, \quad (44)$$

$$\mathbb{E}_{I_{ij^*}} \left[ \frac{(\beta_i^O(t) - \beta_{j^*i}(t))\tau W h_{ij^*}}{(N_0W + I_{ij^*} + P_i^*(t)h_{ij^*}(t)) \ln 2} \right] = V + \gamma^* - \alpha_2^*, \quad (45)$$

$$f_i^*(t) \geq 0, \quad \alpha_1^* \geq 0, \quad \alpha_1^* f_i^*(t) = 0, \quad (46)$$

$$P_i^*(t) \geq 0, \quad \alpha_2^* \geq 0, \quad \alpha_2^* P_i^*(t) = 0, \quad (47)$$

and

$$\begin{cases} \kappa[f_i^*(t)]^3 + P_i^*(t) - P_i^{\max} \leq 0, \\ \gamma^* \geq 0, \\ \gamma(\kappa[f_i^*(t)]^3 + P_i^*(t) - P_i^{\max}) = 0. \end{cases} \quad (48)$$

From (44) and (46), we deduce

$$f_i^*(t) = \sqrt{\frac{\beta_i^L(t)\tau}{3L_i\kappa(V + \gamma^*)}}.$$

Additionally, from (45) and (47), we can find that if

$$\mathbb{E}_{I_{ij}^*} \left[ \frac{(\beta_i^O(t) - \beta_{j^*i}(t))\tau W h_{ij^*}}{(N_0 W + I_{ij^*}) \ln 2} \right] > V + \gamma^*,$$

we have a positive optimal transmit power  $P_{ij}^* > 0$  which satisfies

$$\mathbb{E}_{I_{ij}^*} \left[ \frac{(\beta_i^O(t) - \beta_{j^*i}(t))\tau W h_{ij^*}}{(N_0 W + I_{ij^*} + P_i^* h_{ij^*}) \ln 2} \right] = V + \gamma^*.$$

Otherwise,  $P_i^* = 0$ . Moreover, note that  $\gamma^*$  is 0 if  $\kappa[f_i^*(t)]^3 + P_i^*(t) < P_i^{\max}$ . When  $\gamma^* > 0$ ,  $\kappa[f_i^*(t)]^3 + P_i^*(t) = P_i^{\max}$ .

## REFERENCES

- [1] C.-F. Liu, M. Bennis, and H. V. Poor, "Latency and reliability-aware task offloading and resource allocation for mobile edge computing," in *Proc. IEEE Global Commun. Conf. Workshops*, Dec. 2017, pp. 1–7.
- [2] M. Chiang and T. Zhang, "Fog and IoT: An overview of research opportunities," *IEEE Internet Things J.*, vol. 3, no. 6, pp. 854–864, Dec. 2016.
- [3] M. Chiang, S. Ha, C.-L. I, F. Risso, and T. Zhang, "Clarifying fog computing and networking: 10 questions and answers," *IEEE Commun. Mag.*, vol. 55, no. 4, pp. 18–20, Apr. 2017.
- [4] S. Wang, X. Zhang, Y. Zhang, L. Wang, J. Yang, and W. Wang, "A survey on mobile edge networks: Convergence of computing, caching and communications," *IEEE Access*, vol. 5, pp. 6757–6779, 2017.
- [5] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: The communication perspective," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 4, pp. 2322–2358, Fourth quarter 2017.
- [6] C. Mouradian, D. Naboulsi, S. Yangui, R. H. Glitho, M. J. Morrow, and P. A. Polakos, "A comprehensive survey on fog computing: State-of-the-art and research challenges," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 1, pp. 416–464, First quarter 2018.
- [7] J. Kwak, Y. Kim, J. Lee, and S. Chong, "DREAM: Dynamic resource and task allocation for energy minimization in mobile cloud systems," *IEEE J. Sel. Areas Commun.*, vol. 33, no. 12, pp. 2510–2523, Dec. 2015.
- [8] Y. Kim, J. Kwak, and S. Chong, "Dual-side optimization for cost-delay tradeoff in mobile edge computing," *IEEE Trans. Veh. Technol.*, vol. 67, no. 2, pp. 1765–1781, Feb. 2018.
- [9] Z. Jiang and S. Mao, "Energy delay tradeoff in cloud offloading for multi-core mobile devices," *IEEE Access*, vol. 3, pp. 2306–2316, 2015.
- [10] Y. Mao, J. Zhang, S. H. Song, and K. B. Letaief, "Power-delay tradeoff in multi-user mobile-edge computing systems," in *Proc. IEEE Global Commun. Conf.*, Dec. 2016, pp. 1–6.
- [11] —, "Stochastic joint radio and computational resource management for multi-user mobile-edge computing systems," *IEEE Trans. Wireless Commun.*, vol. 16, no. 9, pp. 5994–6009, Sep. 2017.
- [12] S. Mao, S. Leng, K. Yang, Q. Zhao, and M. Liu, "Energy efficiency and delay tradeoff in multi-user wireless powered mobile-edge computing systems," in *Proc. IEEE Global Commun. Conf.*, Dec. 2017, pp. 1–6.
- [13] J. Xu, L. Chen, and S. Ren, "Online learning for offloading and autoscaling in energy harvesting mobile edge computing," *IEEE Trans. Cogn. Commun. Netw.*, vol. 3, no. 3, pp. 361–373, Sep. 2017.
- [14] Y. Sun, S. Zhou, and J. Xu, "EMM: Energy-aware mobility management for mobile edge computing in ultra dense networks," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 11, pp. 2637–2646, Nov. 2017.



- [15] S. Ko, K. Han, and K. Huang, "Wireless networks for mobile edge computing: Spatial modeling and latency analysis," *IEEE Trans. Wireless Commun.*, vol. 17, no. 8, pp. 5225–5240, Aug. 2018.
- [16] R. Deng, R. Lu, C. Lai, T. H. Luan, and H. Liang, "Optimal workload allocation in fog-cloud computing toward balanced delay and power consumption," *IEEE Internet Things J.*, vol. 3, no. 6, pp. 1171–1181, Dec. 2016.
- [17] G. Lee, W. Saad, and M. Bennis, "An online secretary framework for fog network formation with minimal latency," in *Proc. IEEE Int. Conf. Commun.*, May 2017, pp. 1–6.
- [18] Q. Fan and N. Ansari, "Workload allocation in hierarchical cloudlet networks," *IEEE Commun. Lett.*, vol. 22, no. 4, pp. 820–823, Apr. 2018.
- [19] —, "Application aware workload allocation for edge computing-based IoT," *IEEE Internet Things J.*, vol. 5, no. 3, pp. 2146–2153, Jun. 2018.
- [20] —, "Towards workload balancing in fog computing empowered IoT," *IEEE Trans. Netw. Sci. Eng.*, 2019, to be published.
- [21] M. Molina, O. Muñoz, A. Pascual-Iserte, and J. Vidal, "Joint scheduling of communication and computation resources in multiuser wireless application offloading," in *Proc. IEEE 25th Annu. Int. Symp. Pers., Indoor, Mobile Radio Commun.*, Sep. 2014, pp. 1093–1098.
- [22] H. Zhang, Y. Xiao, S. Bu, D. Niyato, F. R. Yu, and Z. Han, "Computing resource allocation in three-tier IoT fog networks: A joint optimization approach combining Stackelberg game and matching," *IEEE Internet Things J.*, vol. 4, no. 5, pp. 1204–1215, Oct. 2017.
- [23] B. Soret, P. Mogensen, K. I. Pedersen, and M. C. Aguayo-Torres, "Fundamental tradeoffs among reliability, latency and throughput in cellular networks," in *Proc. IEEE Global Commun. Conf. Workshops*, Dec. 2014, pp. 1391–1396.
- [24] M. Bennis, M. Debbah, and H. V. Poor, "Ultrareliable and low-latency wireless communication: Tail, risk, and scale," *Proc. IEEE*, vol. 106, no. 10, pp. 1834–1853, Oct. 2018.
- [25] C.-P. Li, J. Jiang, W. Chen, T. Ji, and J. Smee, "5G ultra-reliable and low-latency systems design," in *Proc. Eur. Conf. Net. Commun.*, Jun. 2017, pp. 1–5.
- [26] M. J. Neely, *Stochastic Network Optimization with Application to Communication and Queueing Systems*. San Rafael, CA, USA: Morgan and Claypool, Jun. 2010.
- [27] S. Coles, *An Introduction to Statistical Modeling of Extreme Values*. London, U.K.: Springer, 2001.
- [28] A. E. Roth and M. A. O. Sotomayor, *Two-Sided Matching: A Study in Game-Theoretic Modeling and Analysis*. New York, N.Y., USA: Cambridge University Press, 1992.
- [29] T. D. Burd and R. W. Brodersen, "Processor design for portable systems," *J. VLSI Signal Process. Syst.*, vol. 13, no. 2-3, pp. 203–221, Aug. 1996.
- [30] P. Mach and Z. Becvar, "Mobile edge computing: A survey on architecture and computation offloading," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 3, pp. 1628–1656, Third quarter 2017.
- [31] C. You, K. Huang, H. Chae, and B. Kim, "Energy-efficient resource allocation for mobile-edge computation offloading," *IEEE Trans. Wireless Commun.*, vol. 16, no. 3, pp. 1397–1411, Mar. 2017.
- [32] J. D. C. Little, "A proof for the queuing formula:  $L = \lambda W$ ," *Operations Research*, vol. 9, no. 3, pp. 383–387, 1961.
- [33] R. Hemmecke, M. Köppe, J. Lee, and R. Weismantel, "Nonlinear integer programming," in *50 Years of Integer Programming 1958-2008: From the Early Years to the State-of-the-Art*, M. Jünger, T. M. Liebling, D. Naddef, G. L. Nemhauser, W. R. Pulleyblank, G. Reinelt, G. Rinaldi, and L. A. Wolsey, Eds. Berlin/Heidelberg, Germany: Springer, 2010, ch. 15, pp. 561–618.
- [34] Y. Gu, W. Saad, M. Bennis, M. Debbah, and Z. Han, "Matching theory for future wireless networks: Fundamentals and applications," *IEEE Commun. Mag.*, vol. 53, no. 5, pp. 52–59, May 2015.

- [35] J. Zhao, Y. Liu, K. K. Chai, Y. Chen, and M. ElKashlan, "Many-to-many matching with externalities for device-to-device communications," *IEEE Wireless Commun. Lett.*, vol. 6, no. 1, pp. 138–141, Feb. 2017.
- [36] M. S. Elbamby, M. Bennis, and W. Saad, "Proactive edge computing in latency-constrained fog networks," in *Proc. Eur. Conf. Netw. Commun.*, Jun. 2017, pp. 1–6.
- [37] C. Perfecto, J. Del Ser, and M. Bennis, "Millimeter-wave V2V communications: Distributed association and beam alignment," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 9, pp. 2148–2162, Sep. 2017.
- [38] B. Di, L. Song, and Y. Li, "Sub-channel assignment, power allocation, and user scheduling for non-orthogonal multiple access networks," *IEEE Trans. Wireless Commun.*, vol. 15, no. 11, pp. 7686–7698, Nov. 2016.
- [39] E. Bodine-Baron, C. Lee, A. Chong, B. Hassibi, and A. Wierman, "Peer effects and stability in matching markets," in *Proc. 4th Int. Symp. Algorithmic Game Theory*, 2011, pp. 117–129.
- [40] A. Al-Shuwaili and O. Simeone, "Energy-efficient resource allocation for mobile edge computing-based augmented reality applications," *IEEE Wireless Commun. Lett.*, vol. 6, no. 3, pp. 398–401, Jun. 2017.
- [41] A. P. Miettinen and J. K. Nurminen, "Energy efficiency of mobile clients in cloud computing," in *Proc. 2nd USENIX Conf. on Hot Topics in Cloud Computing*, Jun. 2010.
- [42] Radiocommunication Sector of ITU, "P.1238-9: Propagation data and prediction methods for the planning of indoor radiocommunication systems and radio local area networks in the frequency range 300 MHz to 100 GHz," ITU-R, Tech. Rep., Jun. 2017.
- [43] Z. Pi and F. Khan, "System design and network architecture for a millimeter-wave mobile broadband (MMB) system," in *Proc. 34th IEEE Sarnoff Symp.*, May 2011, pp. 1–6.
- [44] C.-F. Liu and M. Bennis, "Ultra-reliable and low-latency vehicular transmission: An extreme value theory approach," *IEEE Commun. Lett.*, vol. 22, no. 6, pp. 1292–1295, Jun. 2018.