Mobile Edge Computing-Enabled Resource Allocation for Ultra-Reliable and Low-Latency Communications

Yun Yu(✉), Siyuan Zhou, Xiaocan Lian, Guoping Tan, and Yingchi Mao

College of Computer and Information, Hohai University, Nanjing, China
yuyun555@126.com, gptan@hhu.edu.cn

Abstract. Mission critical services and applications with computation-intensive tasks require extremely low latency, while task offloading for mobile edge computing (MEC) incurs extra latency. In this work, the optimization of power consumption and delay are studied under ultra reliable and low latency (URLLC) framework in a multiuser MEC scenario. Delay and reliability are relying on users' task queue lengths, which is attested by probabilistic constraints. Different from the current literature, we consider a comprehensive system model taking into account the effects of bandwidth, computation capability, and transmit power. By introducing the approach of Lyapunov stochastic optimization, the problem is solved by splitting the multi-objective optimization problem into three single optimization problems. Performance analysis is conducted for the proposed algorithm, which illustrates that the tradeoff parameter indicates the tradeoff between power and delay. Simulation results are presented to validate the theoretical analysis of the impact of various parameters and demonstrate the effectiveness of the proposed approach.

Keywords: Mobile edge computing · Resource allocation · Probability constraints · Ultra reliable and low latency (URLLC) · Stochastic network optimization

1 Introduction

5G mobile network promotes extensive and deep integrations with vertical industry. The concept of Mobile Edge Computing (MEC) emerged and gradually evolved as one of the possible basic core structures of 5G. As a structure under 5G architecture, MEC adapts to dissimilar computing, caching and communication deployments in various scenarios [1–4]. Relying on the structure MEC, mobile network and Internet service achieve effective integration and further expand

This work was supported in part by the National Natural Science Foundation of China (No. 61701168, 61832005, 61571303) and the Fundamental Research Funds for the Central Universities (No. 2019B15614).

X. B. Zhai et al. (Eds.): MLICOM 2019, LNICST 294, pp. 347–360, 2019.
https://doi.org/10.1007/978-3-030-32388-2_30

to other fields, like location service, advanced reality, Internet-of-Things, and computation assistant. Ultra reliable and low latency communication (URLLC) corresponds to services requiring low latency and high reliability, such as self-driving, industrial automation, etc.

MEC system is located between wired network and wireless access point and established by one or more MEC servers which are the core of the whole system. By offloading computation-intensive tasks to nearby MEC servers could significantly enhance the performance of user devices, including battery life and latency [5]. Applying MEC on URLLC needs emphasis on reducing latency and power consumption. Nevertheless, task offloading introduces extra latency, and its efficiency relying highly on channel conditions. Therefore, it is of value to consider bandwidth which is closely connected to channel conditions and wireless radio resources [6].

Resource allocation on MEC has attracted great attention. You *et al.* proposed a MECO system with multiple users to optimize energy consumption [7]. In [8], a joint allocation of computation and communication resources are studied in multi-user mobile cloud computing under power and latency constraints. Work of [9] focus on completion time minimization and compare two different access schemes. Furthermore, Liu *et al.* introduce stochastic network optimization on MEC system and establish a multi-user multi-server system to study the tradeoff between power and delay [10].

Nevertheless, [10] consider radio resource allocation and [11] considers violation constraints, channel condition on [11] is interference-related which makes it complicated to estimate transmission rate. Also, CPU cores are independent on servers with one-to-one correspondence to user devices on multi-user condition. In this case, queuing on servers can be simplified and better channel model should be felicitated.

In our work, seeking clarification of the relationship between power and delay indulges the optimization of our work. We consider an MEC system with multiple mobile devices in which computing tasks arrive on mobile devices in a random manner. Based on the Lyapunov optimization theory [12], the radio and computational resources are joint considered to make the power consumption minimize under the latency and reliability constraints. Pickands-Balkema-de Haan theorem of extreme value theory is used to descript the queue length which exceed the threshold. And the results show the trade-off between power consumption and latency of mobile devices.

The organization of this paper is characterized as follows. We describe a system model that satisfies the requirements of URLLC structure in Sect. 2, latency and reliability constraints are imposed in Sect. 3, and optimization problems are schemed and solved in Sect. 4. Simulation results are displayed in Sect. 5 with analysis, and we will conclude this paper in Sect. 6.

2 System Model

As shown in Fig. 1, a set U of UEs with local computation capacity is considered in our system. A single server with N CPU cores deals with the tasks offloaded

by UEs in parallel. Tasks are emerged by UEs and part of them are offloaded to MEC server, meanwhile the UEs simultaneously process the rest of tasks locally. Therefore, queues exist at both users and server side. Channel access scheme is chosen as Frequency Division Multiple Access (FDMA), transmission rate is proportional to bandwidth. End-to-End delay includes transmission delay on offloading condition and computing delay. Resource allocation directed at URLLC works is reflected at computational resource allocation and power control, and constrained by power consumption and delay.

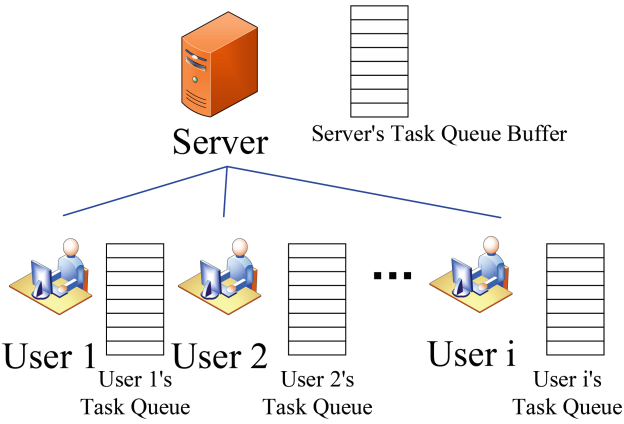


Fig. 1. System model

2.1 Queuing Model at User Side

Tasks arrive in stochastically and follow an arbitrary probability distribution. Task queue length is $Q_i(t)$ on time slot $t \in \{0, 1, 2, ...\}$ for the user $i \in U$. Task arrivals $A_i^u(t)$ meets Poisson task arrivals during time slot t with mean value γ in the unit of bits and are independent and identically distributed [12]. $B_i^u(t)$ is the task accomplishment in time slot t in the unit of bits, in which the local computation tasks $B_i^{u1}(t) = \tau \frac{f_i^u(t)}{L_i}$ and offloaded tasks $B_i^{u2}(t) = \tau R_i(t)$ are both considered, $B_i^u(t) = B_i^{u1}(t) + B_i^{u2}(t)$. L_i denotes the required CPU cycle frequency per bit, CPU cycle frequency is $f_i^u(t)$. The queue length on slot $t+1$ (in the unit of bits) evolves as $Q_i(t+1) = \max\{Q_i(t) - B_i^u(t), 0\} + A_i^u(t)$. The transmission rate at UE side for task offloading is

$$R_i(t) = \alpha_i(t) W \log_2 \left(1 + \frac{p_i(t) H_i(t)}{N_0 \alpha_i(t) W}\right). \tag{1}$$

W is total system bandwidth, N_0 is the power spectral density of the additive white Gaussian noise, and $\alpha_i(t)$ is a bandwidth allocation vector applying FDMA

for user $i \in U$. $H_i(t)$ denotes the channel power gain from user $i \in U$ to the server with transmit power $p_i(t)$.

For local computational resource and transmit power allocation, we impose following constraints for each UE $i \in U$:

$$\begin{cases} \sum\limits_{i \in U} \alpha_i(t) \leq 1 \\ \alpha_i(t) \geq 0 \\ 0 \leq p_i(t) \leq {p_i}^{\max} \\ 0 \leq f_i^u(t) \leq {f_u}^{\max} \end{cases} \tag{2}$$

where ${p_i}^{\max}$ and ${f_u}^{\max}$ are the upper bound of transmission power and local computational capability.

2.2 Queuing Model at Server Side

We denote the task offloading queue length as $Z_i(t)$ bits at the server side on time slot t, the queue length at time slot $t+1$ evolves as $Z_i(t+1) = \max\{Z_i(t) - B_i^s(t), 0\} + A_i^s(t)$, $A_i^s(t)$ denotes task arrivals at server in time slot t at server side, $A_i^s(t) = \min\left\{\max\left\{Q_i(t) - B_i^{u1}(t), 0\right\}, \tau R_t(t)\right\}$. Therefore $Z_i(t+1) \leq \max\{Z_i(t) - B_i^s(t), 0\} + \tau R_t(t)$. Computing accomplishment in time slot t is $B_i^s(t) = \tau \frac{f_i^s(t)}{L_i}$ in which $f_i^s(t)$ is the CPU cycle frequency that allocated to each CPU core to serve user $i \in U$.

At the server side, the computational resource is allocated by constraints as follows:

$$\begin{cases} \sum\limits_{i \in U} \mathbb{1} \cdot \{f_i^s(t) > 0\} \leq N \\ f_i^s(t) \in \{0, f_s^{\max}\}, i \in U \end{cases} \tag{3}$$

where $\mathbb{1}\{\cdot\}$ is an indicator function, $f_s^{\max}$ is the upper bound of the computational capability at server side.

3 Latency and Reliability Constraints

According to Little's Law, the average queuing delay is proportional to the average queue length. However, relying only on the average queue length without considering queuing length probability distribution to evaluate latency and reliability lacks accuracy. Taking the statistic results of queue length and queuing delay into account could increase accuracy immensely. Furthermore, violation of the queue length and queuing delay constraints could decrease the reliability of computation tasks. For instance, offloaded tasks would be deleted if a finite-length queuing buffer is overloaded. So, we impose the queue length probability constraint:

$$\lim_{T \to \infty} \frac{1}{T} \sum_{t=1}^{T} \Pr\left(Q_i(t) > d_i^u\right) \leq \varepsilon_i^u, \tag{4}$$

in which d_i^u is the queue length bound, ε_i^u is the tolerable violation probability and $\varepsilon_i^u \ll 1$.

According to Pickands-Balkema-de Haan theorem for exceedances over thresholds, when the threshold d_i^u is platitudinous high, the cumulative distribution function (CDF) of the excess part of the queue closely approaches generalized Pareto Distribution (GPD) [13].

Applying Pickands-Balkema-de Haan Theory to our problem, the expectation and variance of the conditional excess queue value on user side are $\frac{\sigma_u}{1-\xi_u}$ and $\frac{\sigma_u^2}{(1-\xi_u)^2(1-2\xi_u)}$, where σ_u is the scale parameter and ξ_u is the shape parameter. The mean value and the variance of the CDF would decline while the scale parameter and shape parameter are reduced. The threshold of the scale parameter and the shape parameter are given as $\sigma_u \leq \sigma_u^{th}$ and $\xi_u \leq \xi_u^{th}$.

Define the excess value of queue length of user $i \in U$ on time slot t is $X_i^u(t)\left|_{Q_i(t)>d_i^u}\right. = Q_i(t) - d_i^u$, and $Y_i^u(t) = [X_i^u(t)]^2$.

Time averaged mean value of excess queue length $\overline{X_i^u}$ and its second moment $\overline{Y_i^u}$ are:

$$\overline{X_i^u} = \lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}[X_i^u(t) \,|Q_i(t) > d_i^u\,] \leq \frac{\sigma_u^{th}}{1-\xi_u^{th}}, \tag{5}$$

$$\overline{Y_i^u} = \lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}[Y_i^u(t) \,|Q_i(t) > d_i^u\,] \leq \frac{2\left(\sigma_u^{th}\right)^2}{\left(1-\xi_u^{th}\right)\left(1-2\xi_u^{th}\right)}, \tag{6}$$

Likewise, the average queue length $Z_i(t)$ and average queuing delay on the server side are proportional to the average task offloading rate. The queuing delay probability constraint at server side is:

$$\lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \Pr\left(\frac{Z_i(t)}{\widetilde{R}_i(t-1)} > d_i^s\right) \leq \varepsilon_i^s, \tag{7}$$

in which $\widetilde{R}_i(t-1) = \frac{1}{t}\sum_{\omega=0}^{t-1} R_i(\omega)$, d_i^s denotes the queuing delay bound and ε_i^s denotes the tolerable violation probability at server side, $\varepsilon_i^s \ll 1$.

Define the excess queue length at server side for user $i \in U$ on time slot t as $X_i^s(t)\left|_{Z_i(t)>\widetilde{R}_i(t-1)d_i^s}\right. = Z_i(t) - \widetilde{R}_i(t-1)d_i^s$, and $Y_i^s(t) = [X_i^s(t)]^2$. We have the expectation and variance of the conditional excess queue value on server side as $\frac{\sigma_s}{1-\xi_s}$ and $\frac{\sigma_s^2}{(1-\xi_s)^2(1-2\xi_s)}$, and $\sigma_s \leq \sigma_s^{th}$, $\xi_s \leq \xi_s^{th}$, where σ_s^{th} and ξ_s^{th} are the thresholds of scale and shape parameter at server side.

Thus, similar to the above, we have constraints as below:

$$\overline{X_i^s} = \lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}\left[X_i^s(t) \,\middle|\, Z_i(t) > \widetilde{R}_i(t-1)d_i^s\right] \leq \frac{\sigma_s^{th}}{1-\xi_s^{th}} \tag{8}$$

$$\overline{Y_i^s} = \lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}\left[Y_i^s(t) \,\middle|\, Z_i(t) > \widetilde{R}_i(t-1)d_i^s\right] \leq \frac{2\left(\sigma_s^{th}\right)^2}{\left(1-\xi_s^{th}\right)\left(1-2\xi_s^{th}\right)} \tag{9}$$

At the user side, computational delay $f_i^u(t)$ and transmission delay $R_i(t)$ are inversely proportional. As users executing local computing tasks, allocating higher local CPU cycle frequency could partially reduce computational delay. For decreasing transmission delay, larger transmit power is needed instead. Therefore, queue length constraints from both user side and server side have already taken these two delays into account. On the other hand, UE's battery consumption would pay for higher computation capability and/or transmit power. Consequently, the tradeoff between power and delay is fatal. As for server side, the computational delay can be neglected, since a better CPU core with preferable computing capability is focusing on the one UE's offloaded task.

Teasing above factors, the end-to-end delay is composed by three components:

- Queuing delay from both user side and server side;
- Computing delay from both user side and server side;
- Transmission delay for users' task offloading.

4 Optimization Framework and Resource Allocation Scheme

Denoting the user side computational resource allocation as $\mathbf{f^u(t)} = \left(f_i^u(t), i \in U\right)$, transmit power allocation as $\mathbf{p(t)} = (p_i(t), i \in U)$, bandwidth resource allocation as $\alpha(\mathbf{t}) = (\alpha_i(t), i \in U)$, the server side computational resource allocation as $\mathbf{f^s}(\mathbf{t}) = (f_i^s(t), i \in U)$. The power consumption is influenced by hardware architecture and CPU-cycle frequency $f_i^u(t)$, so we give out the local power consumption as $\kappa[f_i^u(t)]^3$, $P(t) = \sum_{i \in U} \left(\kappa[f_i^u(t)]^3 + p_i(t)\right)$.

We formulate an optimization problem as follows:

$$\begin{aligned} \min_{\mathbf{f^{u(t)}}, \mathbf{p(t)}, \alpha(\mathbf{t}), \mathbf{f^s(t)}} \quad & \lim_{T \to \infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}[P(t)] \\ \text{s.t.} \quad & \text{(2) and (3) for resource allocation} \\ & \text{(4) and (7) for queue length and delay} \\ & \text{(5), (6), (8) and (9) for GDP} \end{aligned} \tag{10}$$

4.1 Lyapunov Optimization

The constraints above are dedicated to make corresponding virtual queues and will be satisfied if the time averaged rate is stable. (11) shows the corresponding virtual queues, and $[\cdot]^+ = max\{\cdot, 0\}$.

$$
\begin{aligned}
Q_i^{(Q)}(t+1) &= \left[Q_i^{(Q)}(t) + \mathbb{1}\cdot\left\{Q_i(t+1) > d_i^u\right\} - \varepsilon_i^u, 0\right]^+ \\
Q_i^{(X)}(t+1) &= \left[Q_i^{(X)}(t) + \left(X_i^u(t+1) - \frac{\sigma_u^{th}}{1-\xi_u^{th}}\right) \times \mathbb{1}\cdot\left\{Q_i(t+1) > d_i^u\right\}, 0\right] \\
Q_i^{(Y)}(t+1) &= \left[Q_i^{(Y)}(t) + \left(Y_i^u(t+1) - \frac{2\left(\sigma_u^{th}\right)^2}{\left(1-\xi_u^{th}\right)\left(1-2\xi_u^{th}\right)}\right) \times \mathbb{1}\cdot\left\{Q_i(t+1) > d_i^u\right\}, 0\right]^+ \\
Z_i^{(Z)}(t+1) &= \left[Z_i^{(Z)}(t) + \mathbb{1}\cdot\left\{Z_i(t+1) > \widetilde{R}_i(t) d_i^s\right\} - \varepsilon_i^s, 0\right]^+ \\
Z_i^{(X)}(t+1) &= \left[Z_i^{(X)}(t) + \left(X_i^s(t+1) - \frac{\sigma_s^{th}}{1-\xi_s^{th}}\right) \times \mathbb{1}\cdot\left\{Z_i(t+1) > \widetilde{R}_i(t) d_i^s\right\}, 0\right]^+ \\
Z_i^{(Y)}(t+1) &= \left[Z_i^{(Y)}(t) + \left(Y_i^s(t+1) - \frac{2\left(\sigma_s^{th}\right)^2}{\left(1-\xi_s^{th}\right)\left(1-2\xi_s^{th}\right)}\right) \times \mathbb{1}\cdot\left\{Z_i(t+1) > \widetilde{R}_i(t) d_i^s\right\}, 0\right]^+
\end{aligned}
\tag{11}
$$

Combining these virtual queues, we can get system queue vector $\mathbf{Q}(t) = \left(Q_i^{(Q)}(t), Q_i^{(X)}(t), Q_i^{(Y)}(t), Z_i^{(Z)}(t), Z_i^{(X)}(t), Z_i^{(Y)}(t), i \in U\right)$ and then have the Lyapunov function $L(\mathbf{Q}(t)) = \frac{1}{2}\sum_{i\in U}\left[\left(Q_i^{(Q)}(t)\right)^2 + \left(Q_i^{(X)}(t)\right)^2 + \left(Q_i^{(Y)}(t)\right)^2 + \left(Z_i^{(Z)}(t)\right)^2 + \left(Z_i^{(X)}(t)\right)^2 + \left(Z_i^{(Y)}(t)\right)^2\right]$ and conditional Lyapunov drift-plus-penalty for time slot t:

$$
\begin{aligned}
&\mathbb{E}\left[\Delta L(t) + VP(t)|Q(t)\right] \\
&\le C + E\Bigg[-\sum_{i\in U}\Big[B_i^u(t)\Big(Q_i^{(x)}(t)Q_i(t) + A_i^u(t) + 2Q_i^Y(t)\cdot\left(Q_i(t) + A_i^u(t)\right) \\
&+ 2\left(Q_i(t) + A_i^u(t)\right)^3\Big) + Q_i^{(Q)}(t)\Big] \times \mathbb{1}\cdot\left\{\max\left\{Q_i(t) - B_i^u(t), 0\right\} + A_i^u(t) > d_i^u\right\} \\
&+ \sum_{i\in U}\left[\left(\tau R_i(t) - B_i^s(t)\right)\left(Z_i^{(X)}(t) + Z_i(t) + 2Z_i^Y(t)Z_i(t) + 2\left(Z_i(t)\right)^3\right) + Z_i^{(Z)}(t)\right] \\
&\times \mathbb{1}\cdot\left\{\max\left\{z_i(t) - B_i^s(t), 0\right\} + \tau R_i(t) > \widetilde{R}_i(t) d_i^s\right\} \\
&+ V\sum_{i\in U}\left(\kappa[f_i^u(t)]^3 + p_i(t)\right)\Big|Q(t)\Bigg]
\end{aligned}
\tag{12}
$$

where $V \in (0, +\infty)$ is a non-negative Lyapunov tradeoff parameter in the unit of $bits^2/W$. According to Lyapunov Optimization Framework, the optimal solution of our problem P is the upper bound of (12).

Resolve this problem into three optimization problems in one time slot : CPU resources allocation at user side and server side, and transmission resource at user side.

4.2 CPU Computational Resource Allocation at User Side

Since the users are independent to each other, the optimal solution of CPU computational resource at user side can be obtained directly by resolving the above optimal problem.

Constraints at user side can be rewritten as:

$$\min_{0\le f_i^u(t)\le f_u^{\max}} \sum_{i\in U} V\kappa[f_i^u(t)]^3 - a_i(t)\tau f_i^u(t)/L_i \tag{13}$$

with $a_i(t) = Q_i^{(Q)}(t) + Q_i(t) + A_i^u(t) + \left(Q_i^{(X)}(t) + Q_i(t) + A_i^u(t) + 2Q_i^{(Y)}(t) \times (Q_i(t) + A_i^u(t)) + 2(Q_i(t) + A_i^u(t))^3\right) \times \mathbb{1} \cdot \{Q_i(t) + A_i^u(t) > d_i^u\}$. Since users are independent to each other, $f_i^*(t) = \min\left\{\sqrt{\frac{a_i(t)\tau}{3V\kappa L_i}}, f_u^{\max}\right\}$ is the answer after differentiation.

Algorithm 1. User side bandwidth allocation

Make accuracy parameter $\mu = 10^{-7}$, allow maximum iteration number $I_{\max} = 200$.
Initialize $l = 0$, $\alpha_i(t) = 0$, $\widetilde{\lambda}_L = \lambda_L(t)$, $\widetilde{\lambda}_U = \lambda_U(t)$.
while $\left|\sum_{i\in U} \alpha_i(t) - 1\right| \ge \mu$ and $l \le I_{\max}$ **do**
  $\widetilde{\lambda} = \frac{1}{2}\left(\widetilde{\lambda}_L + \widetilde{\lambda}_U\right)$.
  $l = l + 1$.
  $\alpha_i(t) = max\left\{\mathcal{A}_i\left(\widetilde{\lambda}\right), 0\right\}$.
  if $\sum_{i\in U} \alpha_i(t) > 1$ **then**
    $\widetilde{\lambda}_L = \widetilde{\lambda}$.
  else
    $\widetilde{\lambda}_U = \widetilde{\lambda}$.
  end if
end while

4.3 Transmit Power and Bandwidth Allocation at User Side

$$\min_{\mathbf{p}(t),\alpha(t)} \sum_{i\in U} Vp_i(t) + (b_i(t) - a_i(t))\,\tau R_i(t) \tag{14}$$

with $b_i(t) = Z_i^{(Z)}(t) + Z_i(t) + \left(Z_i^{(X)}(t) + Z_i(t) + 2Z_i^{(Y)}(t) \cdot Z_i(t) + 2(Z_i(t))^3\right) \times \left\{Z_i(t) + \tau R_i(t) > \tilde{R}_i(t-1)d_i^s\right\}$.

For user set $U'(t) = \left\{i \,|\, i \in U, a_i(t) \le b_i(t)\right\}$, if $a_i(t) \le b_i(t)$, the optimal transmit power and bandwidth are $p_i^*(t) = 0$ and $\alpha_i^*(t) = 0$. Only if the quantity of local task buffer is larger than that on server side would the mobile device

execute offloading. For $U^c(t) = U \backslash U'(t)$, apart from user $i \in U'(t)$, we consider solving transmit power and bandwidth allocation through alternating optimization. In each iteration, all numerical results are obtained in closed forms by Lagrangian Method. The alternating minimization process ensures the global optimal solution, which is literally termed as the Gaussian-Seidel Method.

– Transmit Power Allocation:

$$\min_{0 \le p_i(t) \le p_i^{\max}, i \in U^c(t)} V p_i(t) + (b_i(t) - a_i(t))\, \tau R_i(t). \tag{15}$$

On condition that the transmission bandwidth is fixed, we can get $p_i^*(t) = \min\left\{\alpha_i(t) W \max\left\{\frac{\tau(a_i(t)-b_i(t))}{V \ln 2} - \frac{N_0}{H_i(t)}, 0\right\}, p_i^{\max}\right\}$.

– Transmission Bandwidth Allocation:

$$\begin{aligned} &\min_{0 \le \alpha_i(t) \le 1} \sum_{i \in U} (b_i(t) - a_i(t))\, \tau R_i(t) \\ &\sum_{i \in U} \alpha_i(t) \le 1, i \in U^c(t) \\ &\alpha_i(t) > 0, i \in U^c(t) \end{aligned} \tag{16}$$

For a fixed transmit power, the Lagrangian method provides an efficient way to obtain optimal results: $\mathcal{L}(\alpha(t), \lambda(t)) = \sum_{i \in U^c(t)} (b_i(t) - a_i(t))\, \tau \alpha_i(t) W \cdot \log_2\left(1 + \frac{p_i(t) H_i(t)}{N_0 \alpha_i(t) W}\right) + \lambda(t)\left(\sum_{i \in U^c(t)} \alpha_i(t) - 1\right)$, $\lambda(t)$ is the Lagrange multiplier, $\alpha^*(t)$ and $\lambda^*(t)$ are the optimal results of this problem.

Applying the Karush-Kuhn-Tucker (KKT) conditions to our problem,

$$\begin{aligned} &\tfrac{\partial \mathcal{L}(\alpha(t), \lambda(t))}{\partial \alpha_i(t)} \big|_{\alpha_i(t) = \alpha_i^*(t)} \\ &= \tau (b_i(t) - a_i(t)) \tfrac{dR_i(t)}{d\alpha_i(t)} + \lambda(t) = 0 \\ &\sum_{i \in U^c(t)} \alpha_i^*(t) - 1 \le 0 \\ &\lambda^*(t) \ge 0 \\ &\lambda^*(t)\left(\sum_{i \in U^c(t)} \alpha_i^*(t) - 1\right) = 0 \end{aligned} \tag{17}$$

When $\lambda^*(t) > 0$, $\sum_{i \in U^c(t)} \alpha_i^*(t) - 1 = 0$; if $\lambda^*(t) = 0$, $\sum_{i \in U^c(t)} \alpha_i^*(t) - 1 \le 0$. Also, if transmit power $p_i(t) = 0$, $\alpha_i(t) \triangleq 0$. $\frac{dR_i(t)}{d\alpha_i(t)}$ is inversely proportional to $\alpha_i(t)$, and $\lim_{\alpha_i(t) \to +\infty} \frac{dR_i(t)}{d\alpha_i(t)} = 0$, $\lim_{\alpha_i(t) \to 0^+} \frac{dR_i(t)}{d\alpha_i(t)} = +\infty$. Apply bisection search over $[\lambda_L(t), \lambda_U(t)]$ for the optimal $\lambda^*(t)$.

$$\begin{cases} \lambda_L(t) = \max\limits_{i \in U} \tau (a_i(t) - b_i(t)) \tfrac{dR_i(t)}{d\alpha_i(t)} \big|_{\alpha_i(t) = 1} \\ \lambda_U(t) = \max\limits_{i \in U} \tau (a_i(t) - b_i(t)) \tfrac{dR_i(t)}{d\alpha_i(t)} \big|_{\alpha_i(t) \to 0} \end{cases} \tag{18}$$

Obtain $\alpha_i^*(t) = max\{\mathcal{A}_i(\lambda^*(t)), 0\}$, in which $\mathcal{A}_i(\lambda^*(t))$ is the solution of $\tau (b_i(t) - a_i(t)) \frac{dR_i(t)}{d\alpha_i(t)} + \lambda^*(t) = 0$. Detail of the solution is particulized in Algorithm 1.

4.4 Server Side CPU Computational Resource Allocation

At the server side, computational resource allocation is measured by CPU cycle frequency which is solved as follows:

$$\begin{aligned} &\max_{f_i^s(t)} \sum_{i \in U} b_i(t) \tau f_i^s(t) / L_i \\ &\sum_{i \in U} \mathbb{1} \cdot \{f_i^s(t) > 0\} \leq N \\ &f_i^s(t) \in \{0 f_s^{\max}\}, \forall i \in U. \end{aligned} \tag{19}$$

Solution to (19) is elaborated in Algorithm 2.

Algorithm 2. Server side computational resource allocation

1: Initialize $k = 1$ and $U = U^c$.
2: **while** $k \leq N$ and $\widetilde{U} \neq \emptyset$ **do**
3: $m^* = argmax_{i \in \widetilde{U}} \{b_i(t)/L_i\}$.
4: $f_{i^*}(t) = f_j^{max}$.
5: $k = k + 1$.
6: $U^c = U^c \backslash U'$.
7: **end while**

5 Numerical Results

We consider an MEC system with 8 users and 1 server, the server is deployed with 8 CPU cores that can serve different users simultaneously. Maximum local computation capability is 10^9 cycle/s and maximum computation capability at server is 10^{10} cycle/s. Assuming the transmission frequency is 5.8 GHz with path loss $L = 60 + 20\log_{10}(5.8) + 24\log_{10} d(\text{dB})$. Users are evenly distributed near the base station. d is the distance between base station and users. We set the parameter $d_i^u = 4\tau\gamma(\text{bit})$, $\varepsilon_i^u = 0.01$, $\sigma_u^{th} = 4\tau\gamma(\text{bit})$, $\xi_u^{th} = 0.3$, $d_i^s = 20\,\text{s}$, $\varepsilon_i^s = 0.01$, $\sigma_s^{th} = 4\tau\widetilde{R}_i(\infty)(\text{bit})$, $\xi_s^{th} = 0.3$. Path loss increases with the transmission frequency, the coherence time is 40 ms. A single wireless channel experience Rayleigh fading with unit variance. Slot length $\tau = 40\,\text{ms}$. Besides, $N_0 = -174\,\text{dBm/Hz}$, $W = 10\,\text{MHz}$, $\kappa = 10^{-27}\,\text{Watt}\,\text{s}^3/cycle^3$, $P_i^{max} = 20\,\text{dBm}$. We first show the convergence of optimal objective function in (14) in Fig. 2. As iteration time rises, the value of optimization objective function converge to the minimum, which proves the validation of the our optimal framework. V is the tradeoff factor (Lyapunov tradeoff parameter) that indicates the relationship between power consumption and latency in our work. Results in Fig. 3 show that, the power consumption at user side decreases as V increases, which means the optimization lays particular emphasis on task queue length with small V. On condition that $V = 0$, local power consumption is small because tasks are unnecessary to be offloaded to MEC server and there is no offloading power

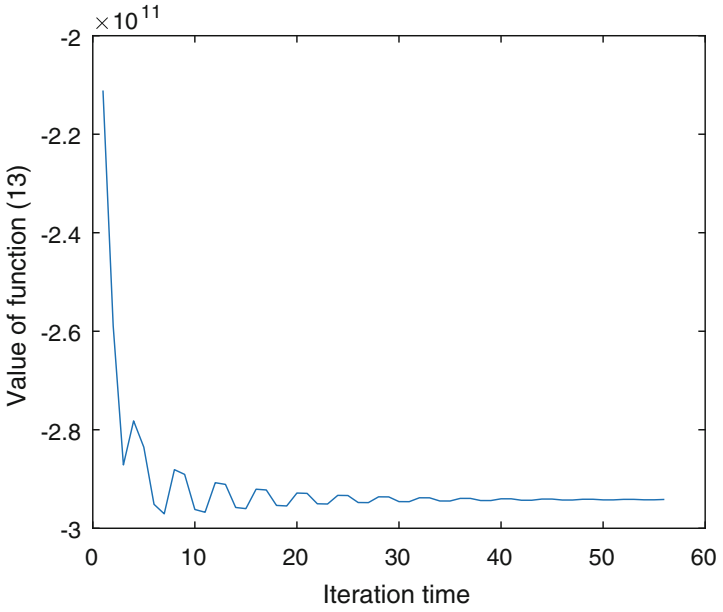
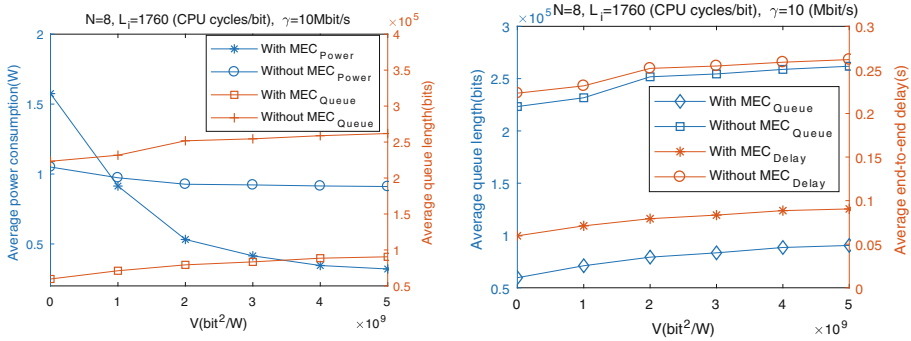


Fig. 2. Iteration convergence of objective function in constraint (14).

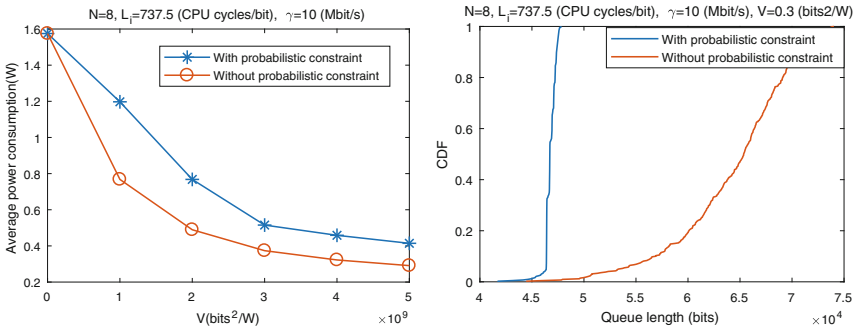


(a) Tradeoff between queue length and power.

(b) Tradeoff between queue length and delay.

Fig. 3. Tradeoff between a UE's average power consumption and task queue length.

consumption but the power consumption of local CPU computing. Main work of optimization at this moment reflects on the control of task queue length. To decrease the task queue length under the condition of having MEC server, users offload data to server with its local CPU still functioning, so the power consumption is high. With the increasement of V, optimization inclines to power consumption and the requirement of queue length abates, while the server could afford part of users' tasks, therefore transmission power is smaller comparing to



(a) Average power consumption versus V.

(b) CDF of queue length.

Fig. 4. Influence of probabilistic constraint on task queue length and V.

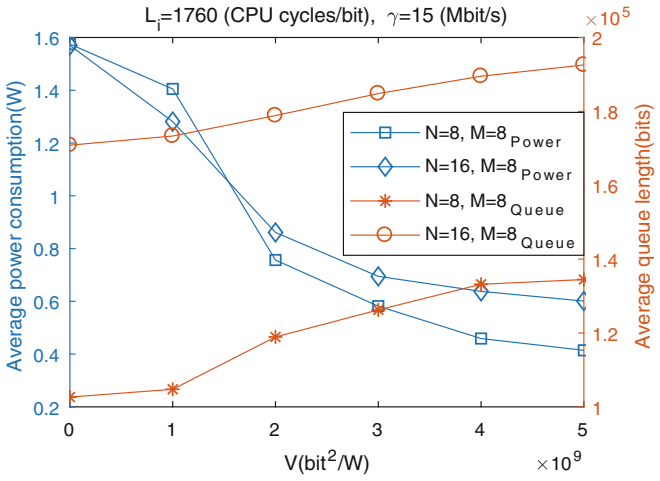


Fig. 5. Power consumption versus V with various numbers of UEs.

local process, and the benefit of using MEC server emerges. Task queue length of users and end-to-end delay are increasing as V increases. Local computing capability is limited and tasks come in continuously, therefore the local task queue length is always longer than that on server. Also, the optimizations on power consumption under both conditions emerge with increase of V and correspond sacrifice part of the queue length, which explains that queue length is getting longer as V increases. According to Little's Law, delay is proportional to queue length, and their variation trend converge gradually.

We consider the delay bound violation of task queue length in Fig. 4(a). Adding the probabilistic constraints of task queue length potentially constraint the delay of task queue and make performance of delay better. The CDF of task queue length in Fig. 4(b) demonstrates this theory. With these probabilistic con-

straints, the quantity of exceed task queue lengths suffer sharp decreases. And for reducing queuing delay, higher CPU cycle frequency and/or higher transmission power are/is required.

Furthermore, let's focus on quantity of users M and CPU core number of servers N in Fig. 5. For users who offload tasks on server, if $N < M$, computational resources can be allocated to each user and there is no extra delay; for $N > M$, with limited CPU cores, allocation of computational resource demand optimization which leads to additional waiting time.

6 Conclusion

In this work, we focus on MEC offloading structure under URLLC framework with multiple users and single server. Each UE could process local computation and offload tasks to servers and the tasks are piled up at both user and server side. The offloading rate is proportional to bandwidth with a coefficient, and the probability constraints claim restricts on task queue length. By applying Lyapunov optimization framework on our work, we transfer constraints into virtual queues and reform our constraints into three parts, different methods are applied to solve these optimization problems. We analyze the influence of the tradeoff factor and the probability constraints, discover that there is a tradeoff between delay and power consumption, and the performance of delay with an MEC server is better than no MEC situation. Quantity of UEs could increase delay if it is larger than server's CPU core number. Convergence of iteration and the CDF of task queue length shows the accuracy of optimization.

References

1. Andreev, S., et al.: Exploring synergy between communications, caching, and computing in 5G-grade deployments. IEEE Commun. Mag. **54**(8), 60–69 (2016)
2. Chiang, M., Ha, S., Chih-Lin, I., Risso, F., Zhang, T.: Clarifying fog computing and networking: 10 questions and answers. IEEE Commun. Mag. **55**(4), 18–20 (2017)
3. Mao, Y., You, C., Zhang, J., Huang, K., Letaief, K.B.: A survey on mobile edge computing: the communication perspective. IEEE Commun. Surv. Tutor. **19**(4), 2322–2358 (2017)
4. Elbamby, M.S., Bennis, M., Saad, W.: Proactive edge computing in latency-constrained fog networks. In: 2017 European Conference on Networks and Communications (EuCNC), Oulu, pp. 1–6 (2017)
5. Sardellitti, S., Barbarossa, S., Scutari, G.: Distributed mobile cloud computing: joint optimization of radio and computational resources. In: IEEE Globecom Workshops (GC Wkshps), Austin, TX, pp. 1505–1510 (2014)
6. Kwak, J., Kim, Y., Lee, J., Chong, S.: DREAM: dynamic resource and task allocation for energy minimization in mobile cloud systems. IEEE J. Sel. Areas Commun. **33**(12), 2510–2523 (2015)
7. You, C., Huang, K.: Multiuser resource allocation for mobile-edge computation offloading. In: IEEE Global Communications Conference (GLOBECOM), Washington, DC, pp. 1–6 (2016)

8. Barbarossa, S., Sardellitti, S., Di Lorenzo, P.: Joint allocation of computation and communication resources in multiuser mobile cloud computing. In: IEEE 14th Workshop on Signal Processing Advances in Wireless Communications (SPAWC), Darmstadt, pp. 26–30 (2013)
9. Le, H.Q., Al-Shatri, H., Klein, A.: Efficient resource allocation in mobile-edge computation offloading: completion time minimization. In: IEEE International Symposium on Information Theory (ISIT), Aachen, pp. 2513–2517 (2017)
10. Mao, Y., Zhang, J., Song, S.H., Letaief, K.B.: Power-delay tradeoff in multi-user mobile-edge computing systems. In: IEEE Global Communications Conference (GLOBECOM), Washington, DC, pp. 1–6 (2016)
11. Liu, C., Bennis, M., Poor, H.V.: Latency and reliability-aware task offloading and resource allocation for mobile edge computing. In: IEEE Globecom Workshops (GC Wkshps), Singapore, pp. 1–7 (2017)
12. Neely, M.J.: Stochastic Network Optimization with Application to Communication and Queueing Systems. Morgan and Claypool Publishers, San Rafael (2010)
13. Coles, S.: An Introduction to Statistical Modeling of Extreme Values. Springer, London (2001). https://doi.org/10.1007/978-1-4471-3675-0