Reliability-Optimal Offloading in Multi-Server Edge Computing Networks with Transmissions Carried by Finite Blocklength Codes

Yao Zhu[1], Yulin Hu[1*], Tao Yang[2] and Anke Schmeink[1]
[1] ISEK Research Group, RWTH Aachen University, Aachen, Germany,
Email: $zhu|hu|schmeink$@umic.rwth-aachen.de
[2] Research Center of Smart Networks and Systems, Fudan University, Shanghai, China
Email: $taoyang$@fudan.edu.cn

***Abstract*—In this paper, we consider a multi-server mobile edge computing (MEC) network supporting low latency computation services, where the wireless data transmission/offloading are carried by finite blocklength (FBL) codes to satisfy the latency constraints. We characterize the FBL reliability of the transmission phase and investigate the extreme event of queue length violation in the computation phase by applying extreme value theory. Following the obtained characterizations, we provide an optimal framework design including time allocation and server selection, aiming at minimizing the overall error probability. Via simulations, we validate our analytical model and show the impact of the number available servers and total workloads on the system performance.**

***Keywords*—ultra-reliable and low-latency communication, edge computing, finite blocklength, extreme value theory**

I. Introduction

The recent emerged mobile edge computing (MEC) technologies enables the flexible and rapid deployment for the latency-sensitive applications by pushing the computation, control and storage to the edge of networks [1]. In comparison to the mobile cloud computing (MCC) which utilizes the centralized servers that are logically and spatially far away from the user [2], the servers in MEC are distributed in the close proximity, e.g., with cellular base stations (BSs) and WiFi access points (APs), which shows significant advantages to reduce the communication latency. On the other hand, the computation capability of single server in MEC is relatively limited and insufficient to satisfy all application for all users, due to the nature of edge networks. Therefore, the cooperative offloading of multiple servers becomes a potential solution to enhance the capacity in the edge network cell [3]–[5].

On the other hand, in the new era of 5G, the arising demands on the ultra-reliable communications, such as in the applications of Internet-of-Thing (IoT) and vehicle-to-everything (V2X), attract a lot of attention from both academic and industrial area. The requirement of the ultra-reliability, i.e., being greater than 99.999% in the ultra-reliable and low-latency communication (URLLC) standard of 3GPP [6], pushes the system design to be revised by taking the extreme cases into account e.g., the transmission error probability in the short blocklength regime from the communication perspective [7], [8] and the deadline violation probability due to waiting time at the computing server's buffer in the computation phase [9].

*Y. Hu is the corresponding author.

In the ultra-reliable and low-latency scenario, the reliability in the perspective of either communication or computation is coupled with the available delay tolerance. Note that the task offloading involves both the data transmission via wireless links and computing process at the MEC servers. For a given maximal allowed service delay, there exists a trade-off in the time allocation for the communication phase and computation phase. On the other hand, due to the user location as well as the random channel behavior, the channel qualities from the user to different MEC servers are not same. In addition, the computing capability and the buffer statuses of MEC servers are also different. Hence, it is not necessarily gainful with respect to reliability to let the user offload its tasks to more and more servers, as the partial offloading to a server via a poor channel likely results in a transmission error. On the other hand, always offloading tasks to less servers (or only one server) may lead to a significant delay (including waiting and computing time) in the computing phase, which increases the latency violation probability. Thus, it is beneficial and important to optimally select multiple servers to improve the reliability of the service to the MEC user. Nevertheless, the framework optimization in the FBL regime has been studied for a two hop relaying network [11], [12], multiple access network [13], [14]. In [9], [15], the task offloading in the ultra-reliable and low-latency communication scenario has been investigated. In particular, [9] studies the extreme probabilistic cases that queue length in the servers violates the delay threshold. [15] proposes an offloading scheme by jointly considering the latency and reliability as a cost function. However, the FBL performance is not considered in those optimal offloading design, i.e., making the results inaccurate for low-latency networks [16]. To the best of our knowledge, the optimal offloading in the multiple servers MEC network operating with FBL codes, especially the joint framework optimization with multi-server selection, is still an open problem.

Motivated by the above observations, in this work we provide an optimal framework design for a multi-server MEC network with respect to time allocation and server selection, aiming at minimizing the overall error probability. In particular, we leverage the communication model in the finite blocklength from Polyanskiy [16] to determine the reliability in the transmission and the extreme value theory (EVT) [20] to study possible error incurred in the cloud server side at the same time. Moreover, the selection of cooperative servers is also optimized to carry the offloading tasks.

The rest of the paper is organized as follows. In Section II, we describe the system. We characterize the end-to-end reliability of the considered network in Section III, following which we introduce our design with respect to framework and server selection in Section IV. We provide our simulation results in Section V and concluded the whole work in Section VI.

II. System Model

We consider a MEC network with K available servers $\mathcal{K} = \{1, .., K\}$, supporting an user equipment (UE) with a compute-intensive application to be completed, as shown in Fig. 1. Due to the lack of the local computation capability, the tasks of the compute-intensive application at the UE has to be offloaded to and computed by the servers remotely.

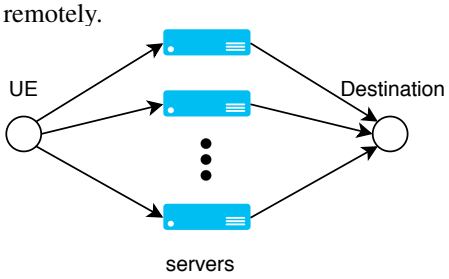

Fig. 1. An example of the considered system.

The system operates in a time-slotted fashion, where time is divided into frames. The service to the application (with a group of computing tasks) is required to be finished in a frame. In addition, as showed in Fig. 2, each frame contains three phases: a communication phase with length of t_1 and a computation phase with length of t_2 and a feedback phase with length of $\bar{t}$. In the communication

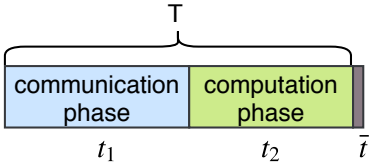

Fig. 2. The structure of a frame.

phase, the device broadcasts τ bits to different servers via wireless channels, including the input data of the tasks needing to be computed as well as the server selection information. Subsequently, in the computation phase the selected servers compute the corresponding tasks with workloads c_k. Denote by T the total cost of the communication and computation phases, i.e., $T = t_1 + t_2$. Finally, the servers transmit computation results to the destination in the feedback phase. Clearly, the total service time of the application satisfies $T + \bar{t}$. However, since the transmit power of the severs is generally high and the data size of the computation results is small, the length of feedback phase $\bar{t}$ usually be considered to be negligible in comparison to T [10]. In this work, we assume $\bar{t}$ to be significant small and constant. In other words, in the framework optimization problem considered later in the work, we only determine the optimal values for t_1 and t_2 while satisfying the maximal allowed T.

Suppose that the duration of one symbol is denoted by T_S. Then, the blocklength of the communication is given by $n_1 = \frac{t_1}{T_S}$. Note that in total τ bits information are transmitted in the communication phase. The corresponding coding rate is actually given by $r = \frac{\tau}{n_1}$ (in bits/symbol). Wireless channels between the UE and the servers are assumed to be independent to each other. We denote by h_k the channel power gain between the UE and server k, by σ_k^2 the noise power of server k, by ϕ_k the path-loss from the UE to the server k and by P the transmit power of the device. Then, the SNR γ_k of the received signal in the communication phase at server k is be expressed as

$$\gamma_k = \frac{h_k P}{\phi_k \sigma_k^2}. \tag{1}$$

Finally, for the computation phase, we assume that the server starts to execute the task as long as the input data of the tasks is successfully received. Note that only the selected servers are active to the computation tasks from the UE. The decision vector of the server selection results is denoted by

$$\mathbf{A} = \{a_1, ..., a_k\}, \tag{2}$$

where a_k is the computation decision of server k. Specially, $a_k = 1$ implies the device chooses to offload the task to server k, while $a_k = 0$ indicates the server k is not selected. In addition, we denote by c_o the total required workloads for the application and by f_k the computation power of server k. We denote by $\hat{\mathcal{K}} = \{\hat{k} | \forall a_{\hat{k}} = 1\}$ the set of the selected servers. Hence, the number of selected servers, i.e., the size of set $\hat{\mathcal{K}}$, is $\sum_k a_k$. Then, the total workloads are equally assigned to selected servers, where the assigned workload of the selected server k is given by $c_k = \frac{c_o}{\sum_k a_k}$, $\forall k \in \hat{\mathcal{K}}$*. Specifically, we define the workload of the non-selected server $\hat{k}$ as $c_{\hat{k}} = \frac{c_o}{\sum_l a_l}, \forall \hat{k} \notin \hat{\mathcal{K}}$ and $\forall l \in \hat{\mathcal{K}}$. This definition does not affect the system operation (as the non-selected server will be idle anyway) but significantly facilitates the problem formulation and optimal design in Section III . Hence, we have

$$c_k = \frac{c_o}{\sum_k a_k}, \forall k \in \mathcal{K}. \tag{3}$$

Since we consider the reliable low-latency communication and computation service, the total service time must be lower than a stringent threshold. Moreover, the reliability is also one of the major concern in the system design. To this end, we investigate the communication behaviour via the wireless channel following the FBL theory and characterize the computation delay by exploiting the extreme value theory.

III. Characterization of the End-to-End Error Probability

In this section, we characterize the end-to-end error probability after modeling the FBL communication errors and computation errors.

A. Communication error in the FBL regime

Recall that the blocklength and the corresponding coding rate are given by $n_1 = \frac{t_1}{T_S}$ (in symbols) and $r = \frac{\tau}{n_1}$ (in bits/symbol). Following the FBL transmission model [16], the (block) error probability of the transmission to server k is given by:

$$\varepsilon_{1,k} = \mathcal{P}(\gamma_k, r, n_1) \approx Q\left(\sqrt{\frac{n_1}{V_k(\gamma_k)}}(\mathcal{C}_k(\gamma_k) - r)\log_e 2\right), \tag{4}$$

*Though the assignment of workloads can be further optimized, which results in a joint optimization problem, it is beyond the scope of this paper.

where $\mathcal{C}_k = \log_2(1+\gamma_k)$ is the Shannon capacity. Moreover, $V_k(\gamma_k)$ is the channel dispersion between the UE and the server k [18]. Under a complex AWGN channel, $V_k = 1 - \frac{1}{(1+\gamma_k)^2}$.

B. Computation Error

Next, we investigate the computation model at the servers in the computation phase t_2. We denote D_k the computing time of server k. In general, the execution time at each server k consists of computing time and queuing delay (waiting time delay in the queue buffer), which can be expressed as:

$$D_k = \frac{c_k}{f_k} + W_k, \tag{5}$$

where W_k is the queuing delay at server k, which is decided by the arriving tasks and computation power f_k.

A computation delay violation error occurs at server k, if the server fails in finishing the assigned tasks within t_2. The probability of this computation error at server k is expressed by

$$\varepsilon_{2,k} = \Pr(D_k \geq t_2), \tag{6}$$

where $\Pr(D_k \geq t_2)$ is the probability that the computing time exceeds t_2. Assuming that the server follows the *first-come, first-served* principle, the queue delay W_k is generally proportional to current queue length Q_k according to the little's law [19]. Obviously, $\varepsilon_{2,k}$ is a monotonously decreasing function with respect to t_2. i.e., a loose threshold leads to a low computation delay violation probability. In addition, for a server k, the corresponding frequency f_k is fixed[†].

The distribution of the execution time is coupled to the distribution of the waiting time W_k. In particular, we have

$$\varepsilon_{2,k} = \Pr(D_k \geq t_2) = \Pr(W_k \geq t_2 - \frac{c_k}{f_k}). \tag{7}$$

For the sake of simplicity, we denote by $\hat{t}_2 = \max\{t_2 - \frac{c_k}{f_k}, 0\}$ the modified delay tolerance in the computation phase.

Due to the high reliability requirement, the computation error probability (computation delay violation probability) should be extremely low, i.e., the complementary cumulative distribution function (CCDF) of the queue delay satisfies $\bar{F}_{W_k}(\hat{t}_2) = \varepsilon_{2,k} = \Pr(W_k \geq \hat{t}_2) \ll 1$. In other words, the tail performance of the monotonically increasing CCDF (with a sufficiently high $\hat{t}_2$) is with the high interests in the design of such reliable MEC network. According to the extreme value theory (EVT) [9], [20], the tail of the probability distribution of $\varepsilon_{2,k}$ can be characterized. Considering the distribution of D_k conditionally exceeding a high threshold d, we denote $X_k = \max\{\hat{t_2} - d, 0\}$ the exceedance of delay tolerance. According to [20], if the threshold d closely approaches $F^{-1}_{W_k}(1)$, the conditional CDF of the exceedance X_k can be expressed as

$$\begin{aligned} F_{X_k|D_k>d}(x_k) &= \Pr(D_k - d \leq x_k | D_k > d) \\ &\approx G(x;\sigma,\xi) = 1 - \left(1 + \frac{\xi x}{\sigma}\right)^{-1/\xi}, \end{aligned} \tag{8}$$

where $G(x;\sigma,\xi)$ is the generalized Pareto distribution (GPD) characterized by the scale parameter $\sigma > 0$ and shape parameter $\xi < 1/2$. Therefore, the error probability with given time slot of computation phase t_2 in the ultra-reliable scenario with the threshold d is given by:

$$\varepsilon_{2,k} = (1 - F_{D_k}(d))(1 - G(\max\{t_2 - \frac{c_k}{f_k} - d, 0\}; \sigma, \xi)). \tag{9}$$

σ and ξ are parameters, which are influenced by the computing task arrival rate and the computation power of the server k, i.e, they can be obtained by the sufficient historical data. More importantly, the validity of the expression does not depend on any specific task distribution [9].

C. End-to-End Error Probability

Note that multiple servers are expected to be selected, while each of them computes a part of the total workloads. The overall service is successful as long as all parts via the corresponding selected servers are successful, i.e., no communication and computation error occurs at each server. We denote ε_k the error probability of the part of the workloads processed by server k. Clearly, an error occurs at server k if the transmission to server k fails or the computation delay violates t_2. On the other hand, if server k is not selected, it contributes nothing to the error probability, i.e., $\varepsilon_k = 0, \forall k \notin \mathcal{K}$. Hence, for all $k \in \mathcal{K}$, ε_k can be generally given by

$$\begin{aligned} \varepsilon_k &= a_k(\varepsilon_{1,k} + (1-\varepsilon_{1,k})\varepsilon_{2,k}) \\ &= a_k(\varepsilon_{1,k} + \varepsilon_{2,k} - \varepsilon_{1,k}\varepsilon_{2,k}). \end{aligned} \tag{10}$$

Denote by ε_O the end-to-end error probability over all the selected servers. ε_O can be obtained by

$$\varepsilon_O = 1 - \prod_k (1 - \varepsilon_k). \tag{11}$$

IV. Framework Optimization with Server Selection

In this section, we propose a framework optimization design to minimize the end-to-end error probability by optimally allocating the sum (of communication and computation) time T to the two phases t_1 and t_2, while optimally determining the offloading decisions with consideration of both the task-partition and the server selection.

A. Problem Formulation

We aim at minimizing the end-to-end error probability by optimally allocating the maximal allowed T, denoted by $T_{\max}$, to t_1 and t_2, and optimally selecting multiple severs. Hence, the optimization problem is formulated by:

$$\begin{aligned}
\underset{\mathbf{t_1}, \mathbf{t_2}, \mathbf{A}}{\text{minimize}} \quad & \varepsilon_O && \text{(12a)} \\
\text{subject to} \quad & \mathbf{A} \in \{0,1\}^K, && \text{(12b)} \\
& t_1 + t_2 \leq T_{\max}, && \text{(12c)} \\
& \varepsilon_k \leq \varepsilon_{\max}, \qquad \forall k \in \mathcal{K}, && \text{(12d)} \\
& c_k = \frac{c_O}{\sum_k a_k}, \quad \forall k \in \mathcal{K}, , && \text{(12e)} \\
& \sum\nolimits_{k=1}^{K} a_k \geq 1, && \text{(12f)}
\end{aligned}$$

[†]Although the server may adopt the dynamic frequency and voltage scaling (DVFS) technique frame-wise, we assume the CPU frequency during the single frame is fixed.

where $\mathbf{A}$ is the server selection results defined in (2). In addition, the constraint (12d) ensures the reliability of the selected links fulfills a given threshold to prevent wasting of network resource.

B. Optimal Solution to (12)

In this subsection, we solve Problem (12) based on the following methodology: We first decompose the original problem in (12) subproblems. Subsequently, we characterize the subproblem and the relationship between the optimal solutions of t_1 and t_2. Finally, according to the characterization of the sub-problems, we reformulate it and combine it back to achieve a solvable reformulation of the original problem in (12).

1) Decomposition and subproblems of (12)*:* With K available servers, there exists 2^K-1 possible server selection combinations of $\hat{\mathcal{K}}$. Therefore, the original problem in (12) can be decomposed into 2^K-1 subproblems with different $\hat{\mathcal{K}}$. For a given set of $\hat{\mathcal{K}}$, the subproblem can be formulated as follows:

$$\begin{aligned}
\underset{\mathbf{t_1},\mathbf{t_2}}{\text{minimize}} \quad & \varepsilon_O && \text{(13a)}\\
\text{subject to} \quad & \varepsilon_k \le \varepsilon_{\max}, \quad \forall k \in \hat{\mathcal{K}}, && \text{(13b)}\\
& c_k = \frac{c_O}{\sum_k a_k}, \forall k \in \hat{\mathcal{K}}, && \text{(13c)}\\
& a_k = 1, \qquad \forall k \in \hat{\mathcal{K}}. && \text{(13d)}
\end{aligned}$$

2) Characterization of Subproblem (13)*:* We have the following two lemmas characterizing Subproblem (13).

Lemma 1. *The end-to-end error probability ε_O is convex in both t_1 and t_2.*

Proof: First, we investigate the convexity of the error probability ε_k . For the non-selected server $k \notin \hat{\mathcal{K}}$, we have $\varepsilon_k = 1$, which is convex in both t_1 and t_2. For the selected server $k \in \hat{\mathcal{K}}$,the second derivative with respect to t_1 is given by:

$$\begin{aligned}
\frac{\partial^2 \varepsilon_k}{\partial t_1^2} &= \frac{\partial^2 \varepsilon_{1,k}}{\partial t_1^2} + 0 - \frac{\partial^2 \varepsilon_{1,k}}{\partial t_1^2}\varepsilon_{2,k}\\
&= \frac{\partial^2 \varepsilon_{1,k}}{\partial n_1^2}\left(\frac{\partial n_1}{\partial t_1}\right)^2 (1-\varepsilon_{2,k})\\
&= \frac{\partial^2 \varepsilon_{1,k}}{\partial n_1^2}\frac{(1-\varepsilon_{2,k})}{T_s^2}.
\end{aligned} \tag{14}$$

Our previous work [21] shows $\frac{\partial^2 \varepsilon_{1,k}}{\partial n_1^2} \ge 0$. Moreover, the error probability in the computation phase always holds $\varepsilon_{2,k} \le 1$. Hence, it also holds $\frac{\partial^2 \varepsilon_k}{\partial t_1^2} \ge 0$, i.e., the error probability for the selected server k is convex in t_1.

Similarly, the second derivative with respect to t_2 is given by:

$$\begin{aligned}
\frac{\partial^2 \varepsilon_k}{\partial t_2^2} &= \frac{\partial^2 \varepsilon_{2,k}}{\partial t_2^2} + 0 - \frac{\partial^2 \varepsilon_{2,k}}{\partial t_1^2}\varepsilon_{1,k}\\
&= \frac{\partial^2 \varepsilon_{2,k}}{\partial t_2^2}(1-\varepsilon_{1,k})
\end{aligned} \tag{15}$$

Considering an ultra-reliable scenario, where the server is only selected if the error probability ε_k is lower than a given low threshold $\varepsilon_{\max} < 0.01$, the error probability in each phase must also lower than the threshold according to (11). Therefore, we assume that t_2 is sufficient to apply the EVT for the selected server, i.e., $t_2 > d$. Based on EVT in (8), the second derivative of $\varepsilon_{2,k}$ with respect to t_2 can be written as

$$\begin{aligned}
\frac{\partial^2 \varepsilon_{2,k}}{\partial t_2^2} &= F_{D_k}(d)\frac{\partial^2 G(t_2-d_k;\sigma,\xi)}{\partial t_2^2}\\
&= F_{D_k}(d)\frac{(1+\xi)}{\sigma^2}\\
&\qquad \cdot \left(1-G(t_2-\frac{c_k}{f_k}-\frac{c_k}{f_k}-d;\sigma,\xi)\right)^{-\frac{2+\xi}{xi}}\\
&\ge 0.
\end{aligned} \tag{16}$$

Note that it also holds $1-\varepsilon_{1,k} \ge 0$. Hence, we have $\frac{\partial^2 \varepsilon_{2,k}}{\partial t_2^2} \ge 0$.

Next, we investigate the convexity of the end-to-end error probability ε_O. For the convenience of notations, we denote $v_k = 1-\varepsilon_k$ the reliability of the server k. Obviously, if $a_k = 1$, we have $v_k = 1$. It implies that if the sever is not selected, the "empty" offloading between such server and the UE is always reliable.

Therefore, the second derivative of ε_O with respect to t_1 is expressed as

$$\frac{\partial^2 \varepsilon_O}{\partial t_1^2} = -\sum_k \frac{\partial^2 v_k}{\partial t_1^2}\prod_{l\ne k} v_l + \sum_k \sum_{l\ne k}\frac{\partial v_k}{\partial t_1}\frac{\partial v_l}{\partial t_1}\prod_{p\ne k, p\ne l} v_p \tag{17}$$

As showed in (14), $\varepsilon_k \le 1$ is a convex and monotonic function with respect to t_1. Therefore, $v_k = 1-\varepsilon_k \ge 0$ is a concave and monotonic function. We have $\frac{\partial^2 v_k}{\partial t_1^2} \ge 0, \forall a_k = 1$ and $\frac{\partial^2 v_k}{\partial t_1^2} = 0, \forall a_k = 0$. Moreover, we have $\operatorname{sgn}\left(\frac{\partial v_k}{\partial t_1}\right) = \operatorname{sgn}\left(\frac{\partial v_l}{\partial t_1}\right), \forall a_k = a_l = 1$ and $\frac{\partial v_k}{\partial t_1} = 0, \forall a_k = 0$, where $\operatorname{sgn}(\cdot)$ is the sign function. Hence, it holds $\frac{\partial^2 \varepsilon_O}{\partial t_1^2} \ge 0$. Analog to t_1, we can show that $\frac{\partial^2 \varepsilon_O}{\partial t_2^2} \ge 0$ since ε_k is also a convex and monotonic function with respect to t_2 according to (15).

As a result, the end-to-end error probability ε_k is convex in both t_1 and t_2. ■

Next, in the following lemma, we state the relationship between the optimal values of t_1 and t_2.

Lemma 2. *The optimal solutions to Problem* (13)*, given by t_1^* and t_2^*, satisfy $t_1^* + t_2^* = T_{\max}$.*

Proof: We prove the lemma with contradiction. Assuming that the optimal solution t_1' and t_2' satisfies the strict inequality of constraint (12e), i.e., $T_{\max} - (t_1' + t_2') = \alpha > 0$. Since the solution is optimal, $\varepsilon_O'(t_1', t_2')$ is always the global minimum. It always holds that $\varepsilon_O'(t_1', t_2') \ge \varepsilon_O(t_1, t_2)$. However, there exists a feasible solution $(t_1'' = t_1' + \alpha, t_2'' = t_2') \in \{t_1, t_2 | t_1 + t_2 \le T_{\max}$. We showed in the proof of lemma 1 that ε_O is an decreasing function with respect of blocklength $n = \frac{t_1}{T_S}$, namely with respect of t_1. Hence, we conclude that the solution (t_1'', t_2'') results a lower error probability comparing to (t_1', t_2'), i.e., $\varepsilon_O''(t_1'', t_2'') < \varepsilon_O'(t_1', t_2')$. Therefore, the assumption of the optimal solution (t_1', t_2') is violated. ■

3) Reformulation of Problem (12)*:* According to the above lemmas characterizing the subproblem (13), the original problem in (12) can be reformulated as

$$\underset{\mathbf{t_1},\mathbf{a}}{\text{minimize}} \quad \varepsilon_O = 1 - \prod_k (1-\varepsilon_k) \tag{18a}$$

$$\text{subject to} \quad t_1 + t_2 = T_{\max}, \tag{18b}$$

$$\varepsilon_k \le \varepsilon_{\max}, \forall k \in \mathcal{K}, \tag{18c}$$

$$c_k = \frac{c_O}{\sum_k a_k}, \forall k \in \mathcal{K}, \tag{18d}$$

$$\sum_k a_k \ge 1, \tag{18e}$$

$$\varepsilon_k \le a_k, \forall k \in \mathcal{K}, \tag{18f}$$

$$\varepsilon_k \le \varepsilon_{1,k} + \varepsilon_{2,k} - \varepsilon_{1,k}\varepsilon_{2,k}, \forall k \in \mathcal{K}, \tag{18g}$$

where (18f) and (18g) are the constraints for the linearization of objective function, which helps us to avoid the multiplication of variable a_k and t_1.

The objective function and constraints of Problem (18) are either affine or convex. Therefore, the reformulated problem in (18) becomes a mixed integer convex problem (MICP) which can be solved efficiently via the recently developed algorithm in [22].

V. Numerical Simulation

In this section, we provide the numerical simulation to evaluate the proposed design. In the simulation, we consider the following parameter setups: First, we consider the UE with a location in the center of a square area of 50 m^2. In addition, for the offloading via the wireless links, we set the bandwidth B= 5Mhz, carrier frequency $F = 2.4$ Ghz, thermal noise power $N = -174$ dBm and transmit power $P = 20$ dBm. Moreover, we adopt the path-loss model in [23], given by $PL = 17.0 + 40.0\log_{10}(r)$. We further set the input data $\tau = 1600$ bits and the total workloads $c_o = 24$ Mcycles. Each of the K servers has a computation power of $f = 3$ Ghz, and the task Poisson arriving rate $\lambda = 3$ Mcycles/s. We obtain the shape parameter $\xi = -0.0214$ and scale parameter $\sigma = 3.4955 \times 10^6$ with a threshold $d = 5.7$ms above 99.9% reliability. Finally, the total delay tolerance is assumed to be $T = 25$ms and the error probability threshold is set to $\varepsilon_{\max} = 0.01$.

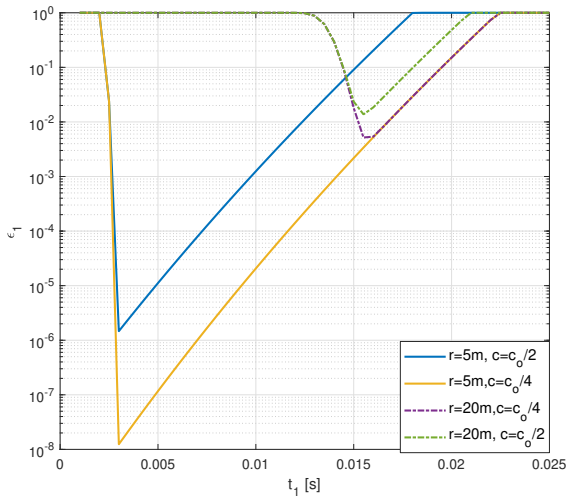

Fig. 3. The error probability in the selected link versus the duration of communication phase t_1 with different setups of the average transmission distances r (from the UE to servers) and the average assigned workloads c

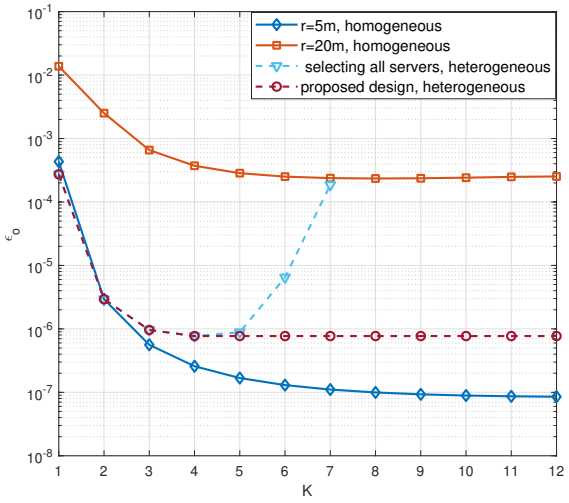

Fig. 4. The overall error probability ε_O versus the number of available servers K with homogeneous servers or heterogeneous servers.

We start with Fig. 3 to present the error probability of a selected link k versus the duration of time slot t_1. First of all, the error probability is observed to be convex in t_1, which confirms Lemma 1. Secondly, the optimal solution of t_1 are not the same for scenarios with different values of distances r (from the UE to the servers). A small t_1 is more preferred, when r is short. On the other hand, for fixed r, changing c does not effects the optimal solution of t_1 too much. Moreover, selecting more servers, corresponding to reducing c, improves the reliability. However, when the server is relatively far from the UE, e.g., $r = 20$m, the performance improvement by selecting more servers becomes significantly smaller than the case with $r = 5$m.

To further investigate the relationship of the selected server and the performance, in Fig. 4 we study the impact of the total number of servers K on the end-to-end error probability ε_O. Both the homogeneous (servers with same distances from the UE) and heterogeneous servers (with different distances from the UE) are considered. Generally, increasing the number of servers (deployed in the system) improves the reliability of the MEC network, which matches with the results of Fig. 3. In the homogeneous cases and the heterogeneous case with the proposed optimal design, the performance improvements by adding one more server in the system are decreasing in K, i.e., the curves become flat when K becomes relatively large. Under such cases, the bottleneck of the system performance is the computation error. It should be pointed out that with the help of the proposed design, the reliability of the MEC network outperforms other cases. More importantly, owning to our design, the performance improvement by adding more servers is more considerable than the rest cases. Finally, it can be observed that without the proposed server selection process, i.e., selecting all the servers deployed in the system (only keeps the computing resource allocation process of the proposed design), the performance is not always improved by adding more servers. In particular, it becomes even worse when K is relatively large. This indicates the proposed joint design is more important in practical scenarios with heterogeneous servers.

Finally, in Fig. 5 we valuate the impact of the total workloads to the reliability of the considered MEC net-

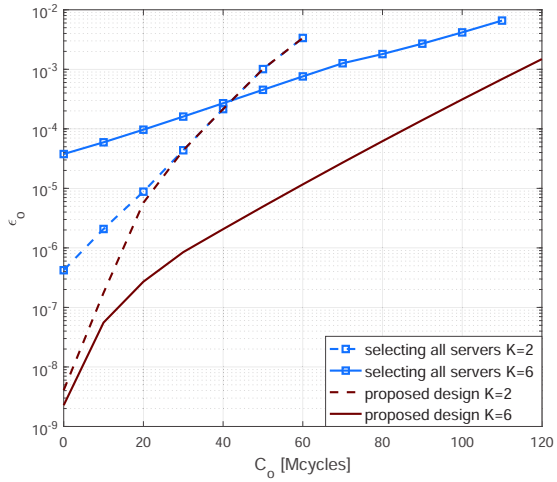

Fig. 5. Overall error probability ε_O versus total workloads c_o. Both the proposed design and the case selecting all servers are provided.

work. As expected, increasing of c_O increases the overall error probability. However, these curves increase in different manners. First of all, the proposed design outperforms the case without the proposed server selection process (simply selecting all servers) no matter when $K = 2$ or $K = 6$. Secondly, the performance advantage of the proposed design is more significant for the scenario with more servers, which matches with the results observed from Fig. 4. Finally and interestingly, it can be observed that under the case without the selection process, the curves of scenarios with $K = 6$ and $K = 2$ across with each other. This indicates that without an appropriate server selection process, it is only beneficial to let more servers join in the offloading when the workload is slight. On the other hand, in the proposed design, the gain from having more servers (for selection) is more remarkable for the scenario with relatively heavier workload.

VI. Conclusion

In this work, we propose a reliability-optimal design in a multi-server edge computing network by joint optimally selecting multiple servers and optimally allocating the allowed time for the communication and computation phases. In particular, both the communication errors due to FBL codes and computation errors caused by delay violation are taken into account in the characterization of the overall error probability. We then formulated an optimization problem which minimizes the overall error probability of the whole service within the maximal allowed service delay. Based on the analysis of the decomposed problems, we reformulated the original problem as a MICP problem, which can be solved efficiently. Simulation results confirm our analytical model and the performance advantage of the proposed design.

References

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

[2] M.R.Rahimi, et al.,"Mobile Cloud Computing: A Survey, State of Art and Future Directions", in *Mobile Netw. Appl.*, vol. 19, no. 2, pp.133-143, April 2014.

[3] L. Chen, S. Zhou and J. Xu, "Computation Peer Offloading for Energy-Constrained Mobile Edge Computing in Small-Cell Networks," in *IEEE/ACM Transactions on Networking,* vol. 26, no. 4, pp. 1619-1632, Aug. 2018.

[4] X. Cao, F. Wang, J. Xu, R. Zhang and S. Cui, "Joint Computation and Communication Cooperation for Energy-Efficient Mobile Edge Computing," in *IEEE Internet of Things Journal.* early access, Oct 2018

[5] R. Yu, J. Ding, S. Maharjan, S. Gjessing, Y. Zhang, and D. Tsang, "Decentralized and optimal resource cooperation in geo-distributed mobile cloud computing," in IEEE *Trans. Emerg. Topics Comput.*, vol. PP, no. 99, pp. 1–13, Sep. 2015.

[6] 3GPP Release 16. Avaliable: http://www.3gpp.org/release-16.

[7] G. Durisi, T. Koch and P. Popovski, "Toward Massive, Ultra-reliable, and Low-Latency Wireless Communication With Short Packets," in Proceedings of the IEEE, vol. 104, no. 9, pp. 1711-1726, Sept. 2016.

[8] C. Sun, C. She, C. Yang, T. Q. S. Quek, Y. Li and B. Vucetic, "Optimizing Resource Allocation in the Short Blocklength Regime for Ultra-Reliable and Low-Latency Communications," in IEEE Transactions on Wireless Communications, vol. 18, no. 1, pp. 402-415, Jan. 2019.

[9] 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.

[10] F. Wang, J. Xu, X. Wang and S. Cui, "Joint Offloading and Computing Optimization in Wireless Powered Mobile-Edge Computing Systems," in *IEEE Transactions on Wireless Communications*, vol. 17, no. 3, pp. 1784-1797, March 2018.

[11] Y. Hu, A. Schmeink and J. Gross, "Blocklength-Limited Performance of Relaying Under Quasi-Static Rayleigh Channels," *IEEE Transactions on Wireless Communications*, vol. 15, no. 7, pp. 4548-4558, July 2016.

[12] Y. Hu, M. C. Gursoy and A. Schmeink, "Relaying-Enabled Ultra-Reliable Low-Latency Communications in 5G," in *IEEE Network*, vol. 32, no. 2, pp. 62-68, March-April 2018.

[13] Y. Hu, M. C. Gursoy and A. Schmeink, "Efficient transmission schemes for low-latency networks: NOMA vs. relaying," Proc. *IEEE PIMRC 2017*, Montreal, QC, 2017, pp. 1-6.

[14] Y. Hu, M. Ozmen, M. C. Gursoy and A. Schmeink, "Optimal Power Allocation for QoS-Constrained Downlink Multi-User Networks in the Finite Blocklength Regime," *IEEE Transactions on Wireless Communications*, vol. 17, no. 9, pp. 5827-5840, Sept. 2018.

[15] J. Liu and Q. Zhang, "Offloading Schemes in Mobile Edge Computing for Ultra-Reliable Low Latency Communications," in IEEE Access, vol. 6, pp. 12825-12837, 2018.

[16] Y. Polyanskiy, H. Poor, and S. Verdu, "Channel coding rate in the finite blocklength regime," *IEEE Trans. Inf. Theory*, vol. 56, no. 5, pp. 2307–2359, May 2010.

[17] S. Coles, *An Introduction to Statistical Modeling of Extreme Values.* Springer, London. 2001.

[18] C. Chen, J. Yan, N. Lu, Y. Wang, X. Yang and X. Guan, "Ubiquitous monitoring for industrial cyber-physical systems over relay-assisted wireless sensor networks," *IEEE Transactions on Emerging Topics in Computing*, vol. 3, no. 3, pp. 352-362, Sept. 2015.

[19] Little, John D. C., "A Proof for the Queuing Formula: L = λ W", *Oper. Res.*, vol. 9, no. 3, pp.383-387, Jun.1961.

[20] S. Coles, *An Introduction to Statistical Modeling of Extreme Values*. Springer, 2001.

[21] Y. Hu, Y. Zhu, M. C. Gursoy, A. Schmeink, "SWIPT-Enabled Relaying in IoT Networks Operating with Finite Blocklength Codes", *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 2, pp.1-14, Feb. 2019.

[22] Lubin, M., Yamangil, E., Bent, R. et al. , "Polyhedral approximation in mixed-integer convex optimization" *Math. Program.*, vol. 172, no. 1, pp.139-168, Nov. 2018.

[23] Y. Corre, J. Stephan and Y. Lostanlen, "Indoor-to-outdoor path-loss models for femtocell predictions," in Proc. *IEEE PIMRC* Toronto, ON, 2011, pp. 824-828.