Stochastic Performance Analysis of Edge-Device Collaboration

Zhidu Li[1,2,3](✉), Xue Jiang[1,2,3], Ruyan Wang[1,2,3], and Dapeng Wu[1,2,3]

[1] School of Communication and Information Engineering, Chongqing University of Posts and Telecommunications, Chongqing, China
lizd@cqupt.edu.cn

[2] Key Laboratory of Optical Communication and Networks, Chongqing, China

[3] Key Laboratory of Ubiquitous Sensing and Networking, Chongqing, China

Abstract. As a new computing paradigm, edge-device collaboration is able to reduce the task execution delay for the devices with limited energy. However, deterministic delay guarantee is unavailable due to the random task arrivals and time-varying channel fading. This paper proposes to study the task execution delay of edge-device collaboration from the probabilistic view of point. Specifically, an optimization problem is formulated with objective of minimizing the proportion of tasks that are not served in time. Then, a general delay violation probability for stochastic system is derived, based on which, the task execution delay violation probability at local device and that at the edge server are obtained respectively. Thereafter, the optimization problem can be transferred into a tractable one where task offloading proportion and task offloading rate can be jointly optimized. The effectiveness of the proposed scheme is finally validated by extensive numerical simulations.

Keywords: Edge-device collaboration · Stochastic delay guarantee · Task offloading · Resource allocation

1 Introduction

With the rapid development of 5G techniques, communication and computing at the edge has become a new trend in the current network [1–3]. Edge-device collaboration is able to makes full use of the advantages of local computing and fast edge computing, such that various types of tasks can be executed efficiently. Therefore, edge-device collaboration reveals its huge potential in different fields,

This work was supported in part by the National Natural Science Foundation of China under grants 61901078, 61871062 and 61771082, and in part by Natural Science Foundation of Chongqing under grant cstc2020jcyj-zdxmX0024, and in part by the Science and Technology Research Program of Chongqing Municipal Education Commission under grant KJQN201900609, and in part by University Innovation Research Group of Chongqing under grant CXQT20017.

J. Xiong et al. (Eds.): MobiMedia 2021, LNICST 394, pp. 94–106, 2021.
https://doi.org/10.1007/978-3-030-89814-4_7

such as industrial Internet, smart city, intelligent transportation and intelligent medical treatment [4–6].

In addition to advantages, there are some challenges in practical edge-device collaboration. On one hand, the processing capability of a device is limited, which may not meet the requirements of delay sensitive tasks, especially when task load is heavy [7]. On the other hand, though edge server has relatively high processing capability, the limited wireless bandwidth becomes the bottleneck of edge computing [8]. Thus, how to assign tasks to the device and edge server to execute is critical for an efficient edge-device collaboration system. However, as task arrivals and wireless channel are time-varying, task offloading may fail sometime, which brings additional difficulty in the optimization of edge-device collaboration [9].

In the literature, edge-device collaboration has been widely concerned. The existing research on edge-device collaboration mainly considers the optimization of the overall performance of the system, such as energy efficiency, computational efficiency and resource utilization. In [10], computational efficiency maximization problem of wireless mobile edge computing (MEC) networks was studied under partial and binary computing offloading modes, where an energy harvesting model was considered at the device side. In [11], task offloading and resource allocation among multiple users served by a base station was studied. With consideration of limited resources, mobility of UE and task delay requirements, a scheme was proposed to obtained the trade-off between service delay and energy consumption. Work [12] designed a game theory based deep reinforcement learning (DRL) algorithm to maximize the energy utility of a network where users may refuse to disclose their network bandwidth and preference information. In order to realize reliable computing offloading of delay sensitive services in Internet of Vehicle, a scheme was proposed for an edge-device collaboration system [13]. In [14], a two-stage joint optimization model was studied to maximize the energy efficiency of an edge-device collaboration system. Work [15] also considered the system reliability and used a double time scale mechanism to minimize the energy consumption of users. However, there are some shortcakes in the existing researches of edge-device collaboration. Firstly, existing studies assume that the task arrivals are deterministic. Hence, the schemes proposed by those studies are not appropriate to the scenario where tasks are randomly generated. Secondly, perfect instantaneous channel information is assumed to be available in those works, which are not reasonable in practical network. Thirdly, existing schemes are based on time block, meaning that a policy is optimized and carried out in a short time, which brings heavy overheads to the system. In summary, how to guarantee the stochastic delay performance of task in an edge-device collaboration is still an open problem.

Motivated by this, we focus on an edge-device collaboration system where the randomness of task arrivals and that of wireless channel are both taken into account. We aim to serve as many tasks as possible under a given delay requirement. The formulated optimization problem is regard to the task allocation and offloading rate configuration. The violation probability of task execution is derived at the local device side and edge server respectively. Thereafter, a two-dimension search approach is proposed to obtain the optimal edge-device collaboration decision. Numerical results verify that the proposed scheme is effective and the task execution delay can be guaranteed from the statistical point of view.

The remaining of this paper is organized as follows. Section 2 introduces the system model. In Sect. 3, performance analysis as well as system optimization are conducted. In Sect. 4, numerical results were presented, compared and discussed. Finally, Sect. 5 concludes the paper.

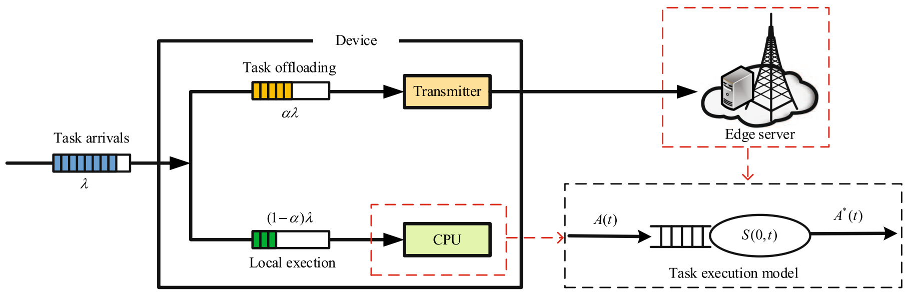

Fig. 1. Task execution in edge-device collaboration manner.

2 System Model

As depicted in Fig. 1, we consider a single-server-single-device edge-device collaboration scenario where the random arrival tasks can be executed by the local device directly or uploaded to the edge server to execute based on first-in-first-out (FIFO) manner. The task arrival process at the device side is assumed to follow Poisson distribution with average rate λ. The packet length of each task is constant and denoted by L. Due to the limited energy of device and the time-varying channel fading, in order to reduce the task execution delay, some tasks should be executed locally and some should be offloaded to the edge server. Here, we use α $(0 \leq \alpha \leq 1)$ to denote the edge-device collaboration parameter, representing the proportion of task offloaded to the edge server. According to the superposition of Poisson process, the task arrival process at the device side and that at the edge server side both follow Poisson distribution with average rates $(1-\alpha)\lambda$ and $\alpha\lambda$ respectively.

In regard to task offloading process, the small-scale fading gain of the channel, denoted by $h(t)$, is assumed to follow Rayleigh distribution with envelope probability density function

$$g_{|h|}(x) = 2xe^{-x^2}. \tag{1}$$

Let p denote the transmission power of the device, l denote the distance between the device and the edge server, ζ denote the path loss constant, φ denote the path loss factor, W denote the uplink bandwidth, N_0 denote the power spectral density of white Gaussian noise. The instantaneous channel capacity for task offloading at time t hold as

$$C(t) = W\log_2\left(1 + \frac{p\zeta l^{-\varphi}|h(t)|^2}{N_0 W}\right), \tag{2}$$

Additionally, as instantaneous channel state is usually unavailable to the local device, it is practical for the device to upload tasks with a fixed rate. Typically, a task is considered to be transmitted successfully if $R \le C(t)$, otherwise, the task offloading fails. Hence, with a given task offloading rate R, the instantaneous channel fading parameter for successful task offloading should hold as follows

$$\begin{gathered} R \le W\log_2\left(1 + \frac{p\zeta l^{-\varphi}|h(t)|^2}{N_0 W}\right) \\ \Updownarrow \\ |h(t)| \ge \sqrt{\frac{N_0 W}{p\zeta l^{-\varphi}}\left(2^{\frac{R}{W}} - 1\right)} \triangleq \delta, \end{gathered} \tag{3}$$

where δ denotes the threshold of channel fading gain, beyond which the task offloading is successful. As a result, the probability of successful task offloading P^{ON} and that of failure task offloading P^{OFF} can be obtained as

$$P^{\mathrm{ON}} = \Pr\{|h(t)| \ge \delta\} = \int_{\delta}^{\infty} 2xe^{-x^2}dx = e^{-\delta^2}, \tag{4}$$

$$P^{\mathrm{OFF}} = \Pr\{|h(t)| < \delta\} = 1 - P^{\mathrm{ON}} = 1 - e^{-\delta^2}. \tag{5}$$

In this paper, we use $D(t(i))$ to denote the execution delay of the i-th task arriving at time t. Besides, the maximum tolerance delay is denoted by d. As mentioned before, deterministic delay guarantee is unavailable to the considered system. Hence, the aim of this paper is to minimize the proportion of tasks that are not served under the delay requirement. Let N denote the number of tasks. The optimization problem can be formulated as follows

$$\begin{aligned} &\min_{R,\alpha} \lim_{N\to\infty} \frac{\sum_{i=1}^{N} I_{\{D(t(i))>d\}}}{N}, \\ &\text{s.t.} \quad 0 \le \alpha \le 1 \\ &\qquad R > 0 \end{aligned} \tag{6}$$

where $I_{\{D(t(i))>d\}}$ is the indicator function. If the execution delay of the ith task exceeds the maximum tolerable delay d, there holds $I_{\{D(t(i))>d\}} = 1$, otherwise,

$I_{\{D(t(i))>d\}} = 0$. The challenge of solving problem (7) lies in characterization of the proportion of tasks that do not meet the delay requirement. In what follows, we use deal with the optimization problem with the help of network calculus theory.

3 Performance Analysis

As the task arrival process is Poisson distributed and the wireless channel is Rayleigh distributed, the task arrival process and task execution process are both as independently and identically distributed (i.i.d). Hence, if the considered system is stable, the delay violation probability suffered by each task is identical. Consequently, we have

$$\begin{aligned}
&\min_{R,\alpha} \lim_{N\to\infty} \frac{\sum\limits_{i=1}^{N} I_{\{D(t(i))>d\}}}{N} \\
=&\min_{R,\alpha} \lim_{N\to\infty} \frac{\sum\limits_{i=1}^{\alpha N} I_{\{D_e(t(i))>d\}} + \sum\limits_{i=1}^{(1-\alpha)N} I_{\{D_l(t(i))>d\}}}{N} \\
=&\alpha \Pr\{D_e > d\} + (1-\alpha)\Pr\{D_l > d\}
\end{aligned} \tag{7}$$

Here, $\Pr\{D_e > d\}$ and $\Pr\{D_l > d\}$ denote the delay violation probabilities of the tasks executed by the edge server and device respectively. Therefore, problem (7) can be transferred to the following problem

$$\begin{aligned}
&\min_{R,\alpha}\ \alpha \Pr\{D_e > d\} + (1-\alpha)\Pr\{D_l > d\} \\
&\text{s.t.}\ \ 0 \le \alpha \le 1 \\
&\qquad R > 0
\end{aligned}. \tag{8}$$

To deal with problem (8), we need to derive the delay violation probability of the tasks executed at the device and at the edge server respectively.

3.1 General Delay Violation Probability

In this section, we first derive a general delay violation probability for a system with both i.i.d arrival and i.i.d service processes. Let $A(s,t)$ and $A^*(s,t)$ denote the cumulative arrivals and cumulative departures of a system during time $(s,t]$. And let $S(s,t)$ denote the cumulative service capacity during time $(s,t]$. According to network calculus theory [16], we have

$$A^*(0,t) = \inf_{0\le s\le t}\{A(0,s) + S(s,t)\}. \tag{9}$$

As for any $d \geq 0$, there always holds $\{D(t) > d\} \subseteq \{A(0,t) > A^*(0,t+d)\}$, we then have

$$\begin{aligned} &\Pr\{D > d\} \leq \Pr\{A(0,t) - A^*(0,t+d) > 0\} \\ =&\Pr\{A(0,t) - \inf_{0 \leq s \leq t+d}\{A(0,s) + S(0,t+d-s)\} > 0\} \\ =&\Pr\{\sup_{0 \leq s \leq t}\{A(s,t)\} - S(0,t)\} > S(0,d)\}. \end{aligned} \tag{10}$$

We then let

$$\begin{aligned} V_s &= e^{\theta(A(t-s,t)-S(t-s,t))} \\ Y_k &= A(k-1,k) \\ Z_k &= S(k-1,k) \end{aligned} \tag{11}$$

There holds

$$\begin{aligned} V_{s+1} &= e^{\theta(A(t-s-1,t)-S(t-s-1,t))} \\ &= e^{\theta\sum_{k=t-s}^{t}(Y_k - Z_k)} \\ &= V_s e^{\theta(Y_{t-s}-Z_{t-s})}. \end{aligned} \tag{12}$$

Because the task arrival and system service processes are independent of each other and both have independent and identically distributed increments, there holds

$$\begin{aligned} &\mathbb{E}\left[V_{s+1}\,|V_1, V_2, \cdots, V_s\right] \\ &= \mathbb{E}\left[V_{s+1}\,|Y_t, Y_{t-1}, \cdots, Y_{t-s+1}\ , Z_t, Z_{t-1}, \cdots, Z_{t-s+1}\right] \\ &= \mathbb{E}\left[V_s e^{\theta(Y_{t-s}-Z_{t-s})}\,|Y_t, Y_{t-1}, \cdots, Y_{t-s+1}\right] \\ &= \mathbb{E}\left[V_s\,|Y_t, Y_{t-1}, \cdots, Y_{t-s+1}\right]\mathbb{E}\left[e^{\theta Y_{t-s}}\right]\mathbb{E}\left[e^{-\theta Z_{t-s}}\right] \\ &= V_s \mathbb{E}\left[e^{\theta A(0,1)}\right]\mathbb{E}\left[e^{-\theta S(0,1)}\right] \\ &\overset{(a)}{\leq} V_s, \end{aligned} \tag{13}$$

Here, step (a) holds since the considered system is stable. Hence, $V_1, V_2, \cdots, V_s$ constitute a nonnegative supermartingale [17,18], we can then obtain the general delay violation probability as

$$\begin{aligned} &\Pr\{D(t) > d\} \\ &= \Pr\{\sup_{0 \leq s \leq t}\{e^{A(s,t)-S(s,t)}\} > e^{S(0,d)}\} \\ &\leq \Pr\{\sup_{1 \leq s \leq t}\{V_{t-s}\} > e^{S(0,d)}\} \\ &\leq \Pr\{V_1 > e^{S(0,d)}\} \\ &\overset{(a)}{\leq} \mathbb{E}[e^{-\theta S(0,d)}]\mathbb{E}\left[e^{\theta A(0,1)}\right]\mathbb{E}[e^{-\theta S(0,1)}] \\ &\overset{(b)}{\leq} \mathbb{E}[e^{-\theta S(0,d)}] \end{aligned} \tag{14}$$

Here step (a) holds based on Chernoff boundary. In step (b), we apply the system stability. More specific, the following equation holds for a stable system.

$$\mathbb{E}[e^{\theta A(0,1)}]\mathbb{E}[e^{-\theta S(0,1)}] \leq 1 \tag{15}$$

where θ is a free parameter. According to (14), delay violation probability decreases as θ increases. Hence, optimal θ can be obtained as

$$\theta^{\text{opt}} = \max\{\theta : \mathbb{E}[e^{\theta A(0,1)}]\mathbb{E}[e^{-\theta S(0,1)}] \leq 1\} \tag{16}$$

Based on (14) and (16), the delay violation probability of the tasks executed at the device and at the edge server can be further derived respectively.

3.2 Local Execution Delay Performance

Let λ_l denote the task arrival rate for the local execution, and there holds $\lambda_l = (1-\alpha)\lambda$. Besides, let $A_l(s,t)$ and $S_l(s,t)$ denote the cumulative arrivals and cumulative service capacity of at the device side during time $(s,t]$. We then have

$$S_l(s,t) = \frac{f_l}{k}(t-s), \tag{17}$$

where f_l denotes the CPU frequency of the device and k denotes the required cycles for executing one bit of each task. Therefore, the delay violation probability at the device side holds as

$$\begin{aligned} \Pr\{D_l > d\} =& \Pr\{D_l(t(i)) > d\} \\ \leq& \mathbb{E}[e^{-\theta_l S_l(0,d)}] \\ =& e^{-\theta_l \frac{f_l}{k} d} \end{aligned} \tag{18}$$

Here, as the task arrival process is Poisson distributed, we have

$$\begin{aligned} &\mathbb{E}\left[e^{\theta_l A_l(0,1)}\right] \\ &= \sum_{n=0}^{\infty} \mathbb{E}\left[e^{\theta_l A_l(0,1)} \,|N(t) = n\right] \Pr\{N(t) = n\}] \\ &= \sum_{n=0}^{\infty} \mathbb{E}\left[e^{\theta_l A_l(0,1)} \,|N(t) = n\right] e^{-\lambda_l} \frac{\lambda_l^n}{n!} \\ &= \sum_{n=0}^{\infty} \mathbb{E}\left[e^{\theta_l n L}\right] e^{-\lambda_l} \frac{\lambda_l^n}{n!} \\ &= e^{-\lambda_l} \sum_{n=0}^{\infty} \frac{\left(\lambda_l e^{\theta_l L}\right)^n}{n!} \\ &= e^{(1-\alpha)\lambda\left(e^{\theta_l L}-1\right)}. \end{aligned} \tag{19}$$

Thus, the optimal free parameter θ_l holds as

$$\begin{aligned} \theta_l^{\text{opt}} &= \max\{\theta_l : \mathbb{E}[e^{\theta_l A_l(0,1)}]\mathbb{E}[e^{-\theta_l S_l(0,1)}] \leq 1\} \\ &= \max\{\theta_l : \frac{(1-\alpha)\lambda(e^{\theta_l L}-1)}{\theta_l} \leq \frac{f_l}{k}\} \end{aligned} \tag{20}$$

3.3 Edge Execution Delay Performance

As the task processing capacity of the edge server is much more powerful than that of the device, we only consider the task offloading delay as the task execution delay at the edge server. Let λ_e denote the task arrival rate for the local execution, and there holds $\lambda_e = \alpha\lambda$. Besides, let $A_e(s,t)$ and $S_e(s,t)$ denote the cumulative arrivals and cumulative service capacity of at the edge server during time $(s,t]$. As tasks can only be offloaded successfully when $R < C(t)$, we have

$$S_e(t-1,t)\begin{cases} = R, & R \le C(t) \\ = 0, & R > C(t) \end{cases} \tag{21}$$

Therefore, the delay violation probability at the edge server holds as

$$\begin{aligned} \Pr\{D_e > d\} &= \Pr\{D_e(t(i)) > d\} \\ &\le \mathbb{E}[e^{-\theta_e S_e(0,d)}] \\ &= (e^{-\theta_e R} P^{\text{ON}} + P^{\text{OFF}})^d \end{aligned}, \tag{22}$$

where P^{ON} and P^{OFF} can be obtained from (4) and (5) respectively.

Similar to (19), we also have

$$\mathbb{E}\left[e^{\theta_e A_e(0,1)}\right] = e^{\alpha\lambda\left(e^{\theta_e L}-1\right)}. \tag{23}$$

Thus, the optimal free parameter θ_e holds as

$$\begin{aligned} \theta_e^{\text{opt}} &= \max\{\theta_e : \mathbb{E}[e^{\theta_e A_e(0,1)}]\mathbb{E}[e^{-\theta_e S_e(0,1)}] \le 1\} \\ &= \max\{\theta_e : \alpha\lambda(e^{\theta_e L} - 1) \le -\ln(e^{-\theta_e R}P^{\text{ON}} + P^{\text{OFF}})\} \end{aligned} \tag{24}$$

3.4 Problem Solution

From (18) and (22), problem (8) can be further transferred into a tractable one

$$\begin{aligned} \min_{R,\alpha} \ & \alpha(e^{-\theta_e R}P^{\text{ON}} + P^{\text{OFF}})^d + (1-\alpha)(e^{-\theta_l \frac{f_l}{k} d}) \\ \text{s.t.} \ & 0 \le \alpha \le 1 \\ & R > 0 \end{aligned} . \tag{25}$$

The above problem can be solved through two-dimension search. Note that the system optimization is from the statistical view of point, which means it is applied to long-term edge-device collaboration. Thus, it is reasonable to apply two-dimension search though such approach has a high computation complexity.

4 Results

Simulation results are provided and discussed in this section. And we use MATLAB to carried out the simulation experiments. Different activation factors and contention intensity are considered in the simulation. The simulation parameters are set as Table 1.

The value of the average transmission rate is related to the user's transmission rate, and the relationship is shown in Fig. 2. And it depicts that there is an

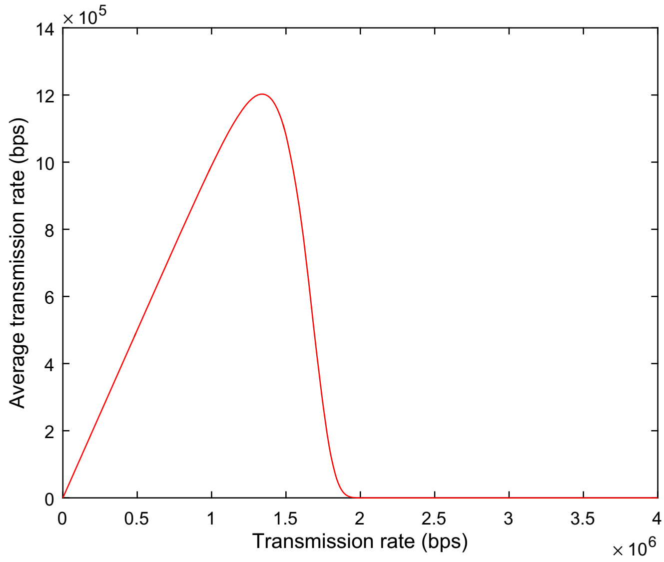

Fig. 2. Relationship between average transmission rate and user's transmission rate.

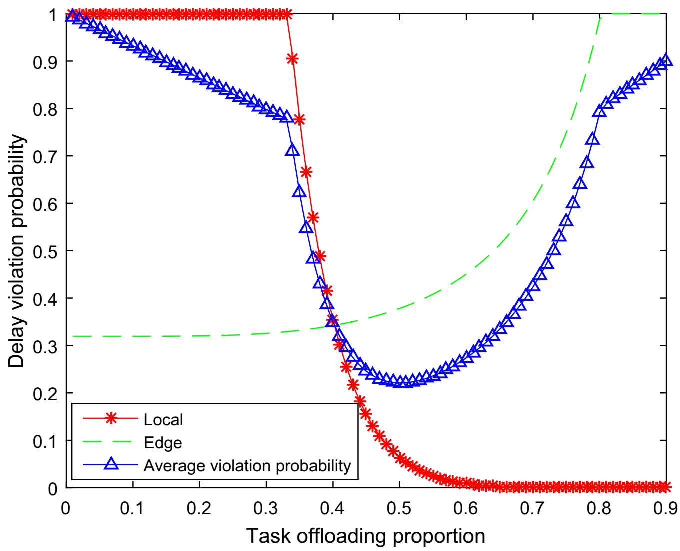

Fig. 3. Relationship between violation probability and task offloading proportion.

Table 1. Simulation parameters

Symbol	Value
Wireless channel bandwidth W	100 kHz
Noise power N_0	1×10^{-17} W
Transmission power of device p	0.1 W
Distance between device and edge server l	10 m
Task arrival rate λ	15 packets/s
Size of each task package	1×10^5 bits
CPU frequency of device f	1×10^9 cycles/s
Number of cycles required for 1 bit data processing k	1×10^3 cycles
Maximum tolerable delay d	0.5 s
Path loss	$10^{-3}l^{-3}$

optimal transmission data rate $R = R^{opt}$ to maximize the average transmission rate of the channel $\bar{R}$. From Fig. 2, we can get $R^{opt}= 1.34 \times 10^6$ bit/s.

Figure 3 shows the relationship between local and offloading delay violation probabilities and task offloading proportion. For local computing, the local delay violation probability decreases with the increase of task offloading proportion. The delay violation probability of edge computing increases with the increase of task offloading proportion. This is because with the increase of task offloading proportion, more tasks are allocated to the edge server for processing, which increases the load of wireless channel and is difficult to guarantee the delay requirement. On the contrary, with the increase of task offloading proportion, the number of tasks left for local processing gradually decreases, and the delay violation probability of local devices becomes smaller and smaller under the condition of the same computing frequency. It can be seen from Fig. 3 that there is an optimal task offloading proportion of 0.50 to minimize the average delay violation probability.

Figure 4 is the relationship between the offloading violation probability and the delay requirement at three transmission rates. It can be seen from the figure that with the increase of delay demand, the offloading violation probability decreases exponentially, and the delay guarantee reliability also increases. Moreover, when the transmission rate is smaller, the offloading violation probability is smaller. This is because the smaller the transmission rate, the less likely it is to exceed the capacity of the channel, thus the greater the probability of transmission success and the smaller the probability of offloading violation.

Figure 5 shows the influence of task offloading rate on the probability of offloading violation under different delay requirements. When the time delay requirement is 0.6 s, the violation probability under different offloading rates is the lowest among them. The reason is that the lower the delay requirement, the less the violation of task delay occurs, that is, the smaller the probability of offloading failure. It can be seen from the figure that there are optimal task offloading rates under different delay requirements, and the delay violation probability is the minimum.

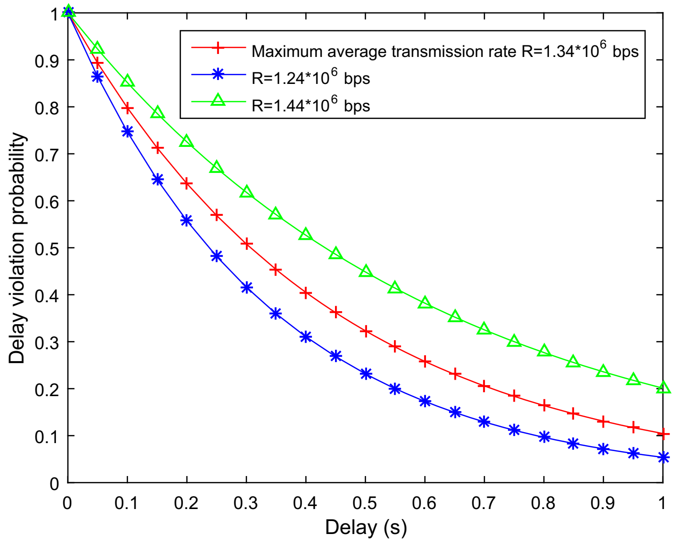

Fig. 4. Relationship between delay violation probability and delay demand.

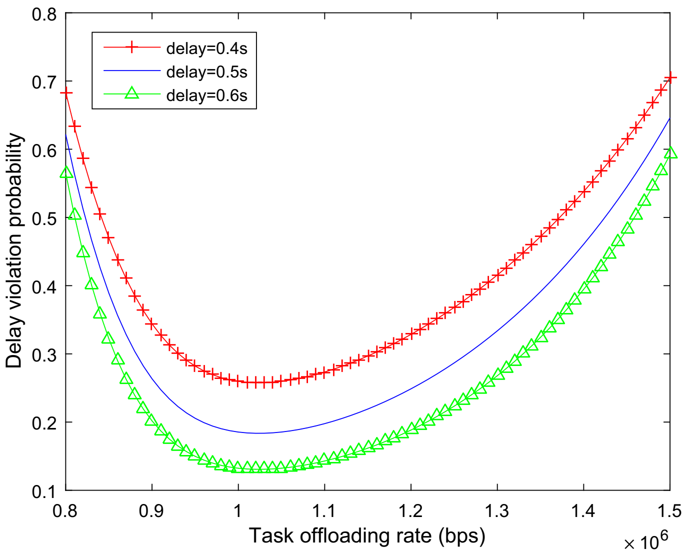

Fig. 5. The relationship between delay violation probability and task offloading rate.

5 Conclusions

In this paper, stochastic delay guarantee was studied in an edge-device collaboration system. The task execution delay violation probability at the local device and that at the edge server were derived respectively. The task allocation and task offloading rate were jointly optimized to minimize the proportion of tasks that could not be served in time. The effectiveness of the proposed scheme was finally validated by extensive numerical simulations. Since this paper only considers a single-node-single-server scenario, the future research will pay more attention to a multi-node-multi-server scenario, where edge-device collaboration is more complex.

References

1. Kong, X., et al.: Mobile edge collaboration optimization for wearable internet of things: a network representation-based framework. IEEE Trans. Ind. Inf. **17**, 5050–5058 (2020). https://doi.org/10.1109/TII.2020.3016037
2. Chen, Z., Li, D., Cuo, J.: Analysis and implementation of a qos-aware routing protocol based on a heterogeneous network. Int. J. Performability Eng. **16**(8), 1235–1244 (2020)
3. Xiong, J., Chen, X., Yang, Q., Chen, L., Yao, Z.: A task-oriented user selection incentive mechanism in edge-aided mobile crowdsensing. IEEE Trans. Netw. Sci. Eng. **7**(4), 2347–2360 (2020)
4. Liu, Y., Peng, M., Shou, G., Chen, Y., Chen, S.: Toward edge intelligence: multiaccess edge computing for 5G and internet of things. IEEE Internet Things J. **7**(8), 6722–6747 (2020)
5. Taleb, T., Samdanis, K., Mada, B., Flinck, H., Dutta, S., Sabella, D.: On multi-access edge computing: a survey of the emerging 5G network edge cloud architecture and orchestration. IEEE Commun. Surv. Tutorials **19**(3), 1657–1681 (2017)
6. Xiong, J., Bi, R., Zhao, M., Guo, J., Yang, Q.: Edge-assisted privacy-preserving raw data sharing framework for connected autonomous vehicles. IEEE Wirel. Commun. **27**(3), 24–30 (2020)
7. Li, Y., Wang, X., Gan, X., Jin, H., Fu, L., Wang, X.: Learning-aided computation offloading for trusted collaborative mobile edge computing. IEEE Trans. Mob. Comput. **19**(12), 2833–2849 (2020)
8. Gao, W., Zhao, Z., Yu, Z., Min, G., Yang, M., Huang, W.: Edge-computing-based channel allocation for deadline-driven IoT networks. IEEE Trans. Ind. Inf. **16**(10), 6693–6702 (2020)
9. Li, S., Lin, S., Cai, L., Li, W., Zhu, G.: Joint resource allocation and computation offloading with time-varying fading channel in vehicular edge computing. IEEE Trans. Veh. Technol. **69**(3), 3384–3398 (2020)
10. Zhou, F., Hu, R.Q.: Computation efficiency maximization in wireless-powered mobile edge computing networks. IEEE Trans. Wirel. Commun. **19**, 3170–3184 (2020)
11. Zhan, W., Luo, C., Min, G., Wang, C., Zhu, Q., Duan, H.: Mobility-aware multi-user offloading optimization for mobile edge computing. IEEE Trans. Veh. Technol. **69**(3), 3341–3356 (2020)
12. Zhan, Y., Guo, S., Li, P., Zhang, J.: A deep reinforcement learning based offloading game in edge computing. IEEE Trans. Comput. **69**(6), 883–893 (2020)

13. Hou, X., et al.: Reliable computation offloading for edge-computing-enabled software-defined IoV. IEEE Internet Things J. **7**(8), 7097–7111 (2020)
14. Cao, T., et al.: Reliable and efficient multimedia service optimization for edge computing-based 5G networks: game theoretic approaches. IEEE Trans. Netw. Serv. Manage. **17**(3), 1610–1625 (2020)
15. Liu, C., Bennis, M., Debbah, M., Poor, H.V.: Dynamic task offloading and resource allocation for ultra-reliable low-latency edge computing. IEEE Trans. Commun. **67**(6), 4132–4150 (2019)
16. Jiang, Y.: Network calculus and queueing theory: two sides of one coin. In: ICST Conference on Performance Evaluation Methodologies and Tools, ICST, pp. 37–48 (2009)
17. Ding, J., Zhu, X., Chen, X.: State equations in stochastic process algebra models. IEEE Access **7**, 61195–61203 (2019)
18. Kingman, J.F.C.: A martingale inequality in the theory of queues. Math. Proc. Cambridge Philos. Soc. **60**(2), 359–361 (1964)