Data Transmission in Mobile Edge Networks: Whether and Where to Compress?

Jinke Ren, *Student Member, IEEE*, Yangjun Ruan, and Guanding Yu, *Senior Member, IEEE*

***Abstract*— Future multimedia communication systems aim to support high-rate data transmission and deliver millisecond-scale latency performance. To achieve this goal, this letter revisits the multimedia compression strategy in the framework of mobile edge computing from the perspective of latency minimization. The uniqueness of this letter is that the delay caused by both data compression/decompression and data transmission are considered. According to where to compress and decompress data, three different schemes are investigated and the end-to-end latency of each scheme is analyzed. We first present some comparative theoretical results on the latency performance of different schemes for given compression ratio. Then, we prove that the optimal compression ratio to minimize the end-to-end latency has a binary structure. This initial work reveals that it is optimal to transmit without compression in the case that the communication capacity is sufficiently large.**

***Index Terms*— Multimedia communication, mobile edge computing (MEC), end-to-end latency.**

I. Introduction

IN MOST current mobile Internet applications, multimedia data are usually compressed before being transmitted. Such data compression techniques could significantly shrink the data size and consequently reduce the bandwidth occupation on both wireless channel and the core network. By this means, the end-to-end packet delay could be effectively reduced since the wireless link is the main bottleneck in general while the computation delay of compression/decompression is usually small enough to be neglected.

However, with the coming era of 5G communications, there would be a significant improvement on the wireless capacity [1]. Also, the delay incurred by core network congestion can be significantly relieved by equipping computation capacity at the network edge, namely the mobile edge computing (MEC) [2], [3]. Therefore, 5G aims to provide ultra-reliable and low-latency communications for delay sensitive applications, such as vehicle networks and virtual/augmented reality. These applications may have a stringent end-to-end latency requirement in the order of milliseconds. In light of this, the current data compression strategy should be revisited since the delay of data compression/decompression and data transmission should be considered with equal importance.

Recently, significant efforts have been devoted into improving the energy-efficiency and end-to-end latency for multimedia communication systems. A joint power allocation and compression ratio selection strategy has been proposed to improve the energy efficiency in a multiuser MEC system [4]. A near-optimal rate control policy for data compression and transmission has been developed to reduce the end-to-end latency, while the framework of MEC has not been considered [5]. Motivated by this, this letter presents our initial work on the optimization of data compression strategy in the framework of MEC. Specifically, we consider a simple model that consists of a source node, a destination node, and one base station (BS) equipped with an edge server. The source node has a volume of multimedia data for the destination node through the BS. Our design objective is to minimize the end-to-end latency of the data transmission by jointly optimizing the compression strategy and the wireless resource allocation.

Unlike the existing works that focus on the development of traditional multimedia communication scheme, i.e., *compression before transmission*, our work initially proposes and analyzes three different compression schemes depending on where to compress and decompress data. We aim to address the following three fundamental issues from the perspective of latency optimization: 1) Whether to compress the raw data before transmission? 2) Where to compress/decompress the data? 3) How much data needs to be compressed? Through mathematical analysis, we prove that the optimal compression strategy can be expressed in a closed-form threshold-based structure for a given compression ratio. Moreover, the optimal compression ratio adaptation is shown to be a binary result. Our theoretical analysis has the great potential to provide an essential reference for the future development of multimedia communication systems.

II. System Model and Problem Formulation

We consider a system with a source node, a destination node, and one BS equipped with an edge server. The source node has a volume of multimedia data, i.e., L bits, that needs to be transmitted to the destination node via the BS. The raw data can be either compressed at the source node or the edge server, and the compressed data can be decompressed either at the edge server or the destination node. In this letter, we consider one pair of source-destination nodes for analysis. However, our proposed framework can be extended into more practical systems with multiple pairs of source-destination nodes with a minor modification on the delay analysis by taking into account the multiple channel access.

A. Wireless Model

Let p^{u} and p^{d} be the transmission powers of the source node and the BS, and g^{u}, g^{d} be the channel power gains of the uplink and downlink channels. Thus, the achievable data

Manuscript received December 30, 2018; accepted January 16, 2019. Date of publication January 23, 2019; date of current version March 8, 2019. The associate editor coordinating the review of this paper and approving it for publication was M. Khabbaz. *(Jinke Ren and Yangjun Ruan contributed equally to this work.) (Corresponding author: Guanding Yu.)*

The authors are with the College of Information Science and Electronic Engineering, Zhejiang University, Hangzhou 310027, China (e-mail: renjinke@zju.edu.cn; ruanyj3107@zju.edu.cn; yuguanding@zju.edu.cn).

Digital Object Identifier 10.1109/LCOMM.2019.2894415

rates for the uplink and downlink can be expressed as

$$r^{\mathrm{u}} = B\log_2\left(1+\frac{p^{\mathrm{u}}g^{\mathrm{u}}}{N_0}\right), \tag{1}$$

$$r^{\mathrm{d}} = B\log_2\left(1+\frac{p^{\mathrm{d}}g^{\mathrm{d}}}{N_0}\right), \tag{2}$$

where B and N_0 are the bandwidth and the variance of complex white Gaussian noise, respectively.

B. Data Compression Model

In this letter, instead of specific data compression technique, we adopt a general compression model. Let $\beta_0 \in (0,1]$ denote the original compression ratio, which means that one-bit raw data would be compressed into β_0 bit. Note that β_0 is a fixed value that is only related to the specific compression technique being used. Furthermore, we adopt the partial data compression model [6], [7] where only $\gamma \in [0,1]$ part of raw data would be compressed while the remaining part is transmitted without compression. Thus, the actual compression ratio of the whole task can be expressed as $\beta = \gamma\beta_0 + 1 - \gamma$. Consequently, we have $\gamma = (1-\beta)/(1-\beta_0)$.

We denote the number of CPU cycles required to compress and decompress one-bit data as n_{c} and n_{d}, respectively. Since only γL bits data are compressed, the number of CPU cycles for task compression and decompression can be expressed as

$$N_{\mathrm{c}}(\beta) = \gamma L n_{\mathrm{c}} \triangleq \frac{1-\beta}{1-\beta_0}N_{\mathrm{c}}, \tag{3}$$

$$N_{\mathrm{d}}(\beta) = \gamma L n_{\mathrm{d}} \triangleq \frac{1-\beta}{1-\beta_0}N_{\mathrm{d}}, \tag{4}$$

where $N_{\mathrm{c}} = Ln_{\mathrm{c}}$ and $N_{\mathrm{d}} = Ln_{\mathrm{d}}$ represent the overall CPU cycles for the whole task compression and decompression.

C. Latency Analysis

In the system, the source node, the edge server, and the destination node are all equipped with limited computation capacities, denoted as f_{s}, f_{e}, and f_{d} (in CPU cycle/s), respectively. To achieve the minimum end-to-end latency, there exist three different compression schemes, elaborated as follows.

Scheme 1 (Source Compression and Destination Decompression): In this scheme, the source node first compresses the data and then transmits the compressed data through the BS. The destination node decompresses the data after receiving the whole data.[1] This scheme is conventional and can be applied in the scenario when wireless channel capacity is limited. In this scheme, the end-to-end latency can be expressed as

$$\begin{aligned} T_1 &= \frac{N_{\mathrm{c}}(\beta)}{f_{\mathrm{s}}} + \frac{\beta L}{t^{\mathrm{u}}r^{\mathrm{u}}} + \frac{\beta L}{t^{\mathrm{d}}r^{\mathrm{d}}} + \frac{N_{\mathrm{d}}(\beta)}{f_{\mathrm{d}}} \\ &= \left(\frac{N_{\mathrm{c}}}{f_{\mathrm{s}}}+\frac{N_{\mathrm{d}}}{f_{\mathrm{d}}}\right)\left(\frac{1-\beta}{1-\beta_0}\right) + \frac{\beta L}{t^{\mathrm{u}}r^{\mathrm{u}}} + \frac{\beta L}{t^{\mathrm{d}}r^{\mathrm{d}}}, \end{aligned} \tag{5}$$

where t^{u}, $t^{\mathrm{d}} \in [0,1]$ are the fractions of one time frame that are allocated to the uplink and downlink channels, respectively.

[1]In practice, the data compression/decompression rely on the data structure as well as the correlation between adjacent data. To ensure the reliability and recoverability, we assume that data compression/decompression and transmission cannot proceed simultaneously.

Scheme 2 (Source Compression and Edge Decompression): In this scheme, the source node first compresses the data and then transmits the results to the BS. After receiving the whole data, the edge server decompresses the data before transmitting to the destination node. This case happens when the edge server is powerful while the destination node is computation-limited. The end-to-end latency of this scheme is given by

$$\begin{aligned} T_2 &= \frac{N_{\mathrm{c}}(\beta)}{f_{\mathrm{s}}} + \frac{\beta L}{t^{\mathrm{u}}r^{\mathrm{u}}} + \frac{N_{\mathrm{d}}(\beta)}{f_{\mathrm{e}}} + \frac{L}{t^{\mathrm{d}}r^{\mathrm{d}}} \\ &= \left(\frac{N_{\mathrm{c}}}{f_{\mathrm{s}}}+\frac{N_{\mathrm{d}}}{f_{\mathrm{e}}}\right)\left(\frac{1-\beta}{1-\beta_0}\right) + \frac{\beta L}{t^{\mathrm{u}}r^{\mathrm{u}}} + \frac{L}{t^{\mathrm{d}}r^{\mathrm{d}}}. \end{aligned} \tag{6}$$

Scheme 3 (Edge Compression and Destination Decompression): In this scheme, the source node directly uploads the raw data to the BS. Next, the edge server compresses the original data and then transmits the results to the destination node, which eventually decompresses the received data. This scheme corresponds to the scenario with powerful edge server but poor downlink capacity. Correspondingly, the end-to-end latency of this scheme can be expressed as

$$\begin{aligned} T_3 &= \frac{L}{t^{\mathrm{u}}r^{\mathrm{u}}} + \frac{N_{\mathrm{c}}(\beta)}{f_{\mathrm{e}}} + \frac{\beta L}{t^{\mathrm{d}}r^{\mathrm{d}}} + \frac{N_{\mathrm{d}}(\beta)}{f_{\mathrm{d}}} \\ &= \left(\frac{N_{\mathrm{c}}}{f_{\mathrm{e}}}+\frac{N_{\mathrm{d}}}{f_{\mathrm{d}}}\right)\left(\frac{1-\beta}{1-\beta_0}\right) + \frac{L}{t^{\mathrm{u}}r^{\mathrm{u}}} + \frac{\beta L}{t^{\mathrm{d}}r^{\mathrm{d}}}. \end{aligned} \tag{7}$$

Note that in each scheme, the special case of $\beta = 1$ corresponds to the same scenario for direct data transmission without any compression.

D. Problem Formulation

The aim of this letter is to achieve the minimum end-to-end latency by jointly optimizing the data compression strategy and wireless resource allocation. Therefore, the latency optimization problem can be formulated as

$$\mathcal{P}_1: \quad \min_{\{t^{\mathrm{u}},t^{\mathrm{d}},\beta\}} \min_i T_i, \tag{8a}$$

$$\text{s.t. } t^{\mathrm{u}} + t^{\mathrm{d}} \le 1, \quad t^{\mathrm{u}}, t^{\mathrm{d}} \ge 0, \tag{8b}$$

$$\beta_0 \le \beta \le 1, \tag{8c}$$

where (8b) indicates the communication resource limitation and T_1, T_2, and T_3 are given in (5), (6), and (7), respectively.

III. Optimal Algorithm

To solve problem $\mathcal{P}_1$, we first fix the value of β and derive the optimal time-slot allocation strategy for each scheme. It is easy to prove that the latency expressions of all three schemes are convex and the constraints are all affine. Therefore, given the value of β, the optimal end-to-end latency for each scheme can be easily derived as

$$\begin{cases} T_1^*(\beta) = (1-\beta)T_{\mathrm{sd}} + \beta\left(\sqrt{T^{\mathrm{u}}}+\sqrt{T^{\mathrm{d}}}\right)^2, \\ T_2^*(\beta) = (1-\beta)T_{\mathrm{se}} + \left(\sqrt{\beta T^{\mathrm{u}}}+\sqrt{T^{\mathrm{d}}}\right)^2, \\ T_3^*(\beta) = (1-\beta)T_{\mathrm{ed}} + \left(\sqrt{T^{\mathrm{u}}}+\sqrt{\beta T^{\mathrm{d}}}\right)^2, \end{cases} \tag{9}$$

where $T^{\mathrm{u}} = \frac{L}{r^{\mathrm{u}}}$, $T^{\mathrm{d}} = \frac{L}{r^{\mathrm{d}}}$, $T_{\mathrm{sd}} = \left(\frac{N_{\mathrm{c}}}{f_{\mathrm{s}}}+\frac{N_{\mathrm{d}}}{f_{\mathrm{d}}}\right)\left(\frac{1}{1-\beta_0}\right)$, $T_{\mathrm{se}} = \left(\frac{N_{\mathrm{c}}}{f_{\mathrm{s}}}+\frac{N_{\mathrm{d}}}{f_{\mathrm{e}}}\right)\left(\frac{1}{1-\beta_0}\right)$, and $T_{\mathrm{ed}} = \left(\frac{N_{\mathrm{c}}}{f_{\mathrm{e}}}+\frac{N_{\mathrm{d}}}{f_{\mathrm{d}}}\right)\left(\frac{1}{1-\beta_0}\right)$.

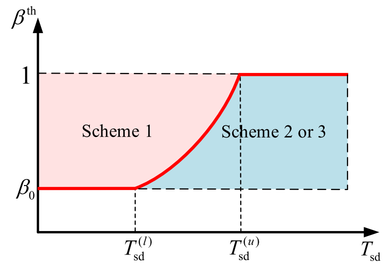

Fig. 1. Illustration of Lemma 2.

Based on the above results, we now make a comparison among the three schemes. Due to the symmetric structures between scheme 2 and scheme 3, we can directly derive the difference between $T_2^*(\beta)$ and $T_3^*(\beta)$ as $T_2^*(\beta) - T_3^*(\beta) = (1-\beta)(T_{\text{se}} + T^{\text{d}} - T_{\text{ed}} - T^{\text{u}})$. Therefore, the optimal choice between scheme 2 and scheme 3 can be obtained as follows.

Lemma 1 (Comparison Between Scheme 2 and Scheme 3): Given the value of β, the selection strategy between scheme 2 and scheme 3 is given by

(1) If $T_{\text{se}} + T^{\text{d}} \le T_{\text{ed}} + T^{\text{u}}$, choose scheme 2.
(2) If $T_{\text{se}} + T^{\text{d}} > T_{\text{ed}} + T^{\text{u}}$, choose scheme 3.

From Lemma 1, we have obtained the attainable latency performance when utilizing edge computing. For ease of notation, we define $T_0 = \min\{T_{\text{se}}+T^{\text{d}}, T_{\text{ed}}+T^{\text{u}}\}$. Then we can derive the optimal selection strategy among all three schemes, as presented in Lemma 2 and also illustrated in Fig. 1.

Lemma 2 (Edge Computing or Not?): Given the value of β, the optimal selection among the three schemes can be summarized as

(1) If $\beta_0 \le \beta < \beta^{\text{th}}$, choose scheme 2 or scheme 3 based on Lemma 1.
(2) If $\beta^{\text{th}} \le \beta \le 1$, choose scheme 1.

The threshold β^{th} is defined as

$$\beta^{\text{th}} = \begin{cases} \beta_0, & \text{if } T_{\text{sd}} \le T_{\text{sd}}^{(l)}, \\ \dfrac{1}{\left(\frac{2\sqrt{T^{\text{u}}T^{\text{d}}}}{T_{\text{sd}}-T_0} - 1\right)^2}, & \text{if } T_{\text{sd}}^{(l)} < T_{\text{sd}} < T_{\text{sd}}^{(u)}, \\ 1, & \text{if } T_{\text{sd}} \ge T_{\text{sd}}^{(u)}, \end{cases} \tag{10}$$

where $T_{\text{sd}}^{(u)} = T_0 + \sqrt{T^{\text{u}}T^{\text{d}}}$ and $T_{\text{sd}}^{(l)} = T_0 + \frac{2\sqrt{\beta_0 T^{\text{u}}T^{\text{d}}}}{1+\sqrt{\beta_0}}$.

Proof: Please refer to Appendix. ■

Remark 1: Lemma 2 reveals that the optimal scheme selection strategy has a threshold-based structure, which mainly depends on T_{sd} and β. In case that $T_{\text{sd}} < T_{\text{sd}}^{(l)}$, the communication capacity is the main bottleneck so we should compress the whole data before transmission. The result becomes contrary when $T_{\text{sd}} > T_{\text{sd}}^{(u)}$ since the computation capacity is the main bottleneck in this case. Moreover, when $T_{\text{sd}}^{(l)} < T_{\text{sd}} < T_{\text{sd}}^{(u)}$, the optimal scheme is also determined by β. Specifically, if β exceeds the threshold β^{th}, the source node should compress the task locally since a large β would lead to a small compression workload, making the transmission latency dominate the overall end-to-end latency. Therefore, choosing scheme 1 can reduce the transmission data to the maximum extent. Otherwise, edge computing should be utilized.

Thus far, we have obtained the optimal scheme selection strategy for fixed β. In what follows, we will optimize the compression ratio β for further performance improvement. It can be observed that the expressions $T_1^*(\beta)$, $T_2^*(\beta)$, and $T_3^*(\beta)$ are all concave over β. Therefore $\min\{T_1^*(\beta), T_2^*(\beta), T_3^*(\beta)\}$ is also concave over β. As a result, the optimal compression ratio must be taken at the endpoints, i.e., β_0 and 1. With simple mathematical calculation, we can derive the optimal compression ratio adaptation strategy as follows.

Theorem 1 (Binary Compression Ratio Adaptation): The optimal compression ratio β^* is given by

(1) If $T_{\text{sd}} \ge T_{\text{sd}}^{\text{th}}$ and $T_0 \ge T_0^{\text{th}}$, then $\beta^* = 1$,
(2) If $T_{\text{sd}} < T_{\text{sd}}^{\text{th}}$ or $T_0 < T_0^{\text{th}}$, then $\beta^* = \beta_0$,

where the thresholds $T_{\text{sd}}^{\text{th}} = T^{\text{u}} + T^{\text{d}} + 2\sqrt{T^{\text{u}}T^{\text{d}}}$ and $T_0^{\text{th}} = T^{\text{u}} + T^{\text{d}} + \frac{2\sqrt{T^{\text{u}}T^{\text{d}}}}{1+\sqrt{\beta_0}}$.

Remark 2: Theorem 1 implies that the optimal compression ratio β^* depends on two terms, i.e., T_{sd} and T_0. Recall that T_{sd} represents the overall computation delay of the source node and the edge node, while T_0 reflects the achievable communication and computation delay when utilizing edge computing. Therefore, when both of them exceed their respective thresholds, i.e., $T_{\text{sd}}^{\text{th}}$ and T_0^{th}, the raw data should be directly transmitted to the destination node through the BS without compression. This case happens when the computation capacities of the source node, edge server, and destination node are all poor. However, as long as one of these two items is under the corresponding threshold, the whole data needs to be compressed, i.e., $\beta^* = \beta_0$, and the scheme selection is determined by Lemma 1 and Lemma 2.

IV. Simulation Results

In this section, simulations are conducted to confirm our theoretical analysis and show the latency performance of the proposed schemes. We consider a small-cell cellular network whose radius is 300 m. The channel gains of both the source-to-edge link and the edge-to-destination link are calculated following the path-loss model: PL [dB] $= 128.1 + 37.6\log d$ [km], and the small-scale fading is Rayleigh distributed with uniform variance [8]. The system bandwidth $B = 10$ MHz. For the computation task, the total data size $L = 100$ Kbits, and the workload, i.e., the number of CPU cycles for compressing and decompressing one-byte task data are $n_{\text{c}} = 330$ and $n_{\text{d}} = 165$ per byte [9].

Fig. 2(a) depicts the end-to-end latency of scheme 2 and scheme 3 versus the source computation capacity, where $f_{\text{d}} = 2 \times 10^9$ CPU cycle/s and $p^{\text{d}} = 24$ dBm. It shows that when the source computation capacity equals the destination computation capacity, scheme 3 outperforms scheme 2. The reason is that the workload for data compression is heavier than that for decompression, i.e., $n_{\text{c}} > n_{\text{d}}$, therefore offloading the compression task to the powerful edge server performs better. However, the result becomes contrary when the source computation capacity exceeds a threshold, which demonstrates the accuracy and applicability of Lemma 1.

Fig. 2(b) shows the end-to-end latency of three schemes versus the compression ratio, where $f_{\text{s}} = 2 \times 10^9$ and $f_{\text{e}} = 8 \times 10^9$ CPU cycle/s, respectively. It is seen that all

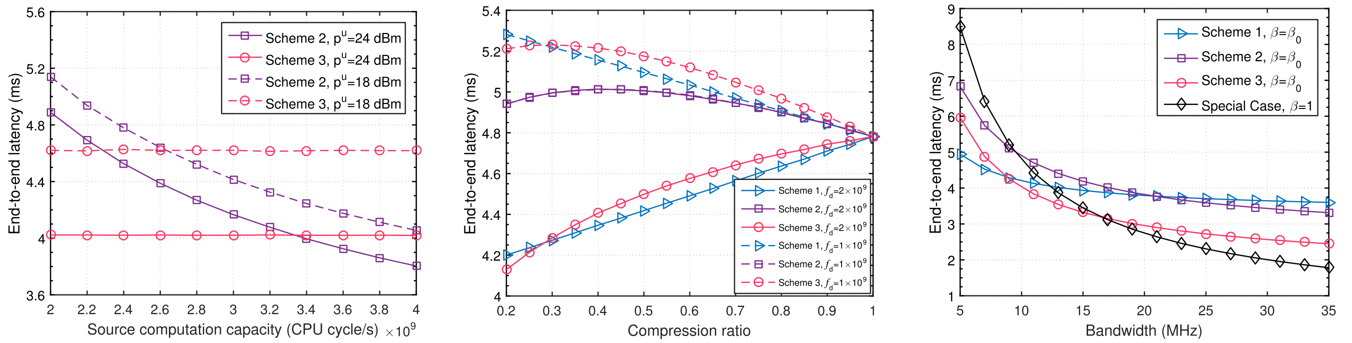

Fig. 2. Simulation results. (a) End-to-end latency vs source computation capacity (b) End-to-end latency vs compression ratio (c) End-to-end latency vs channel bandwidth.

schemes have identical latency performance when $\beta = 1$, which corresponds to the special case without compression or decompression. Besides, in case that the destination capacity is rather poor, i.e., $f_{\rm d} = 1\times 10^9$ CPU cycle/s, scheme 2 performs the best among all schemes because it excludes the limited destination computing and fully utilizes the source and edge computation resources. This result becomes contrary when the destination computation capacity is adequate. Specifically when $f_{\rm d} = 2\times 10^9$ CPU cycle/s, the latency of scheme 1 and scheme 3 increase with β. It is because the communication capacity becomes the main bottleneck in this case, and a larger β leads to a larger end-to-end latency.

Fig. 2(c) presents the end-to-end latency of three schemes with compression ratio $\beta = \beta_0$ and the special case of $\beta = 1$, where $f_{\rm s} = f_{\rm d} = 2 \times 10^9$ and $f_{\rm e} = 1 \times 10^{10}$ CPU cycle/s. From the figure, we can observe the end-to-end latency of all schemes decrease with the system bandwidth. It is rather intuitive that a larger bandwidth can provide stronger communication capacity and result in smaller transmission latency. Specifically, when the system bandwidth is small, scheme 1 outperforms the special case, demonstrating that compression before transmission is better than direct transmission in current communication-limited systems. However, as the bandwidth increases, the special case achieves the best performance among all schemes, indicating that raw data no longer needs to be compressed in future communication-sufficient systems.

V. Conclusion

Taking into account the compression/decompression delay, this letter has revisited the multimedia compression strategy in an MEC system to minimize the end-to-end latency. We have analyzed the latency performance of three different compression schemes and presented analytical results to compare them. We have further proved that the optimal compression ratio to minimize the end-to-end latency has a binary structure, depending on the relative values of computation delay and communication delay. Our initial work in this letter sheds some light on the joint optimization of compression and communication strategies for further latency performance improvement. Nevertheless, some interesting issues, such as the energy-latency tradeoff, non-linear compression model, and multi-hop transmission also deserve further investigation in the future work.

Appendix

The difference between $T_1^*(\beta)$ and $\min\{T_2^*(\beta), T_3^*(\beta)\}$ is

$$\begin{aligned}\Delta T &= T_1^*(\beta) - \min\{T_2^*(\beta), T_3^*(\beta)\} \\ &= (1-\beta)\left(T_{\rm sd} - T_0 - \frac{2\sqrt{\beta T^{\rm u} T^{\rm d}}}{1+\sqrt{\beta}}\right). \end{aligned} \tag{11}$$

Note that $\frac{2\sqrt{\beta}}{1+\sqrt{\beta}}$ increases with $\beta \in [\beta_0, 1]$. Denote $T_{\rm sd}^{(l)} = T_0 + \frac{2\sqrt{\beta_0 \ T^{\rm u} T^{\rm d}}}{1+\sqrt{\beta_0}}$ and $T_{\rm sd}^{(u)} = T_0 + \sqrt{T^{\rm u} T^{\rm d}}$. Therefore, when $T_{\rm sd} \le T_{\rm sd}^{(l)}$, $\Delta T \le 0$. Otherwise when $T_{\rm sd} \ge T_{\rm sd}^{(u)}$, $\Delta T > 0$. Moreover, when $T_{\rm sd} \in \left(T_{\rm sd}^{(l)}, T_{\rm sd}^{(u)}\right)$, if $\beta \le \frac{1}{\left(\frac{2\sqrt{T^{\rm u}T^{\rm d}}}{T_{\rm sd}-T_0} - 1\right)^2}$, $\Delta T \ge 0$, otherwise $\Delta T < 0$. In summary, we can define a threshold function $\beta^{\rm th}$ as in (10). Then, when $\beta_0 \le \beta < \beta^{\rm th}$, we have $\Delta T > 0$ and scheme 2 or scheme 3 should be chosen based on Lemma 1. Otherwise, when $\beta^{\rm th} \le \beta \le 1$, we have $\Delta T \le 0$ and scheme 1 should be chosen.

References

[1] M. Agiwal, A. Roy, and N. Saxena, “Next generation 5G wireless networks: A comprehensive survey,” *IEEE Commun. Surveys Tuts.*, vol. 18, no. 3, pp. 1617–1655, 3rd Quart., 2016.

[2] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, “A survey on mobile edge computing: The communication perspective,” *IEEE Commun. Surveys Tuts.*, vol. 19, no. 4, pp. 2322–2358, 4th Quart., 2017.

[3] S. Sardellitti, G. Scutari, and S. Barbarossa, “Joint optimization of radio and computational resources for multicell mobile-edge computing,” *IEEE Trans. Signal Inf. Process. Over Netw.*, vol. 1, no. 2, pp. 89–103, Jun. 2015.

[4] X. Li, C. You, S. Andreev, Y. Gong, and K. Huang, “Wirelessly powered crowd sensing: Joint power transfer, sensing, compression, and transmission,” *IEEE J. Sel. Areas Commun.*, vol. 37, no. 2, pp. 391–406, Feb. 2019.

[5] E. Belyaev, A. Turlikov, and A. Ukhanova, “Low-latency video transmission over high-speed WPANs based on low-power compression,” in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Sydney, NSW, Australia, Apr. 2010, pp. 1–6.

[6] C. You, K. Huang, H. Chae, and B.-H. Kim, “Energy-efficient resource allocation for mobile-edge computation offloading,” *IEEE Trans. Wireless Commun.*, vol. 16, no. 3, pp. 1397–1411, Mar. 2017.

[7] J. Ren, G. Yu, Y. Cai, and Y. He, “Latency optimization for resource allocation in mobile-edge computation offloading,” *IEEE Trans. Wireless Commun.*, vol. 17, no. 8, pp. 5506–5519, Aug. 2018.

[8] *Third Generation Partnership Project; Technical Specification Group Radio Access Network; Evolved Universal Terrestrial Radio Access (E-UTRA): Further Enhancements to LTE Time Division Duplex (TDD) for Downlink-Uplink (DL-UL) Interference Management and Traffic Adaptation (Release 11)*, document TR 36.828 V11.0.0, 3GPP, 2012.

[9] A. P. Miettinen and J. K. Nurminen, “Energy efficiency of mobile clients in cloud computing,” in *Proc. USENIX Conf. Hot Topics Cloud Comput. (HotCloud)*, Boston, MA, USA, Jun. 2010, pp. 1–7.