Optimizing Peak Age of Information in Mobile Edge Computing

Jianhang Zhu, Jie Gong
Guangdong Key Laboratory of Information Security Technology,
School of Computer Science and Engineering, *Sun Yat-sen University*, Guangzhou, 510006, China
Email: zhujh26@mail2.sysu.edu.cn, gongj26@mail.sysu.edu.cn

***Abstract*—Recently, information freshness in real-time monitoring systems has received increasing attention. Age of Information (AoI) has emerged as a metric for measuring the information freshness. In many applications, the information embedded in an update packet needs to be computed before delivering to a destination. Mobile edge computing (MEC) can efficiently accomplish the computing process. In the MEC system, transmission process and computation process are coupled together, jointly affecting freshness. In this paper, we consider minimizing the average peak AoI (PAoI) in an MEC system, where each update is received and computed by an edge server before delivering to the destination. We consider the generate-at-will source model and study when to generate a new update. We prove that the fixed threshold policy is optimal for arbitrary transmission time and computation time distributions. Our numerical simulation results not only validate the theoretical findings, but also demonstrate the behaviors of the average PAoI versus the mean transmission time and the mean computation time.**

***Index Terms*—Age of information, transmission-computation trade-off, mobile edge computing**

I. Introduction

Recently, there has been a trend in mobile computing towards shifting cloud functions to network edges, such as base stations and access points, in order to utilize the vast amount of idle computation power and storage space available there for computation-intensive and latency-critical tasks of mobile devices. This trend is known as Mobile Edge Computing (MEC) [1]. MEC provides low-latency services, and combined with the extensive data collection capabilities of the Internet of Things (IoT) [2], it enables various real-time applications such as remote monitoring and control, phase packet update in smart grids, and environment monitoring for autonomous driving. The performance of these systems is closely tied to the freshness of the information they provide. The concept of Age of Information (AoI), defined as the time elapsed since the generation of the last received update, was introduced in [3] to quantify this freshness. In MEC systems, in addition to transmission latency and update frequency, the impact of computation on AoI should also be taken into account, which is still an open problem for AoI optimization.

This research was funded by the National Natural Science Foundation of China grant number 62171481, the National Key Research and Development Program of China grant number 2019YFE0114000, the Natural Science Foundation of Guangdong Province grant number 2021A1515011124, and the Science and Technology Program of Guangzhou under Grant 202201011577.
Corresponding author: Jie Gong.

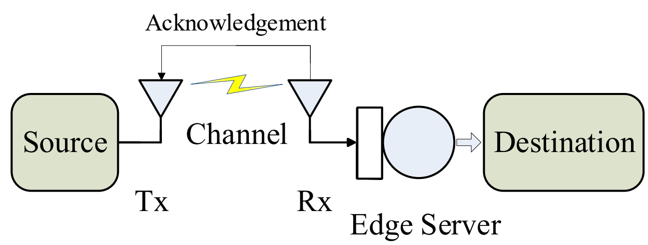

Fig. 1: Status update system with MEC.

To address this issue, we study a state update system incorporating MEC, as illustrated in Fig. 1, where a source node generates update packets and transmits them to an edge server via a channel. The server then computes the packets and sends the results to a destination. There is a queue at the server to buffer the received packets when necessary. Both transmission time and computation time are assumed to be random and unknown apriori. Computation affects AoI performance in two ways. First, computation delay at the edge server directly increases the age of updates. Second, updates may wait in the server queue due to limited computing resources at the edge server and hence, become stale. When computation time is taken into account, the source should choose to generate and transmit a new update either before or after current update's computation is complete. The former option utilizes the channel and the edge server simultaneously, which may benefit AoI performance, but may also result in new data becoming stale in the queue. The latter can completely avoid queueing at the edge server, but increases the inter-generation time. Thus, the status update process must be carefully designed to minimize AoI by avoiding queueing in the server while reducing the inter-generation time.

As mentioned before, transmitting a new update before the completion of the current update's computation can potentially reduce the AoI, as it better utilizes the transmission resource of the channel and the computing resource of the edge server. However, we inevitably encounter queueing at the edge server which is harmful to AoI performance. To tackle the problem, this paper investigates the transmission-computation trade-off in a single-source MEC system with a non-preemptive server, where the newly arrived packet will wait in a queue until the current one completes computing, and a generate-at-will source, which can generate updates at any time as long as the channel is available[1].

A. Related Work

The Age of Information has become an important metric in various status update systems, leading to studies aiming at minimizing it. These studies can be classified into two categories based on the type of source. The first category focuses on uncontrollable sources, as seen in works such as [3]–[6]. In these works, minimizing AoI typically involves optimizing the service rate and packet management strategy in the queues. For instance, [3] analyzed the average AoI in the First-Come-First-Serve (FCFS) system for different types of queues and found that there exists an optimal offered load for each queue to minimize the average AoI. Similarly, [7] discussed the Last-Come-First-Serve (LCFS) queuing system and the impact of preemptive servers on reducing AoI. In [6], a new simplified technique for evaluating AoI in finite-state continuous-time queuing systems was derived for the multi-source LCFS system with Poisson arrivals and exponential service time. The second category considers controllable sources, also known as the generate-at-will source model, as seen in [8]–[10]. For example, [8] considered a generate-at-will source model where updates can be generated at any time according to a scheduling policy and proved that a threshold-based policy is optimal when considering independent and identically distributed transmission times. However, these works did not consider the impact of computation on AoI.

The impact of computing on AoI has garnered increasing attention [11]–[16], as the information contained in a status update packet is often not revealed until it has been processed. Ref. [11] explored the impact of computing on AoI by scheduling computing tasks in the central cloud. The scheduling policy for update cloud computing, ignoring transmission time, was studied in [12]. In [13], the trade-off between computing and transmission was analyzed, where each packet was pre-processed before being transmitted. In [14] and [15], the average AoI with exponential transmission time and service time was analyzed for the single-user case when MEC was considered. However, these works did not address the fundamental question of what the optimal scheduling policy is to achieve the minimum AoI in an MEC system that considers both transmission and computation times. Recently, the authors in [17] investigated the optimal scheduling policy in a single-source single-server system with a fixed request delay. The requests initiated by the monitor arrive at the source after a fixed delay, and then the source generates an update. Transmission time is not taken into account, so there is no challenge posed by the transmission-computation coupling problem. In our research, we consider both transmission and computation time to address this fundamental question.

[1]In general systems, the source has a queue to store new updates when the channel is unavailable. However, generating a new update when the channel is unavailable is obviously sub-optimal when considering the freshness of information. Therefore, in this paper, we consider the source has no queue and only generates updates when the channel is available.

B. Main Results

In this paper, we aim to minimize the average peak AoI (PAoI) in MEC systems while considering transmission delay and computation time. PAoI is a new AoI metric that reflects the average value of the maximum AoI, introduced in [18]. We choose this metric because, on the one hand, the average PAoI is more meaningful for scenarios that require reducing AoI violation probability, rather than the average AoI. On the other hand, optimizing average AoI in systems with generate-at-will sources is usually challenging [8], and only heuristic strategies can be proposed in continuous-time systems [16]. Even in a cloud computing system without considering transmission time, minimizing average AoI for general service time is still an open problem [12]. Typically, PAoI minimization leads to well-structured solutions [17]. Therefore, we consider transmission time and computation time to be independently and identically distributed (i.i.d.) and seek the optimal scheduling policy to minimize the average PAoI. The key contributions of this paper are:

- We formulate the average PAoI minimization problem. The feasible policy space consists of all causal policies, where the control decisions depend on the history and current information of the system. We first prove that the optimal policy is a continuous working policy (Lemma 1). Then, we prove that a randomized threshold policy is optimal (Theorem 1). Subsequently, we prove that a fixed threshold policy is optimal (Theorem 2), and derive the conditions that the optimal threshold must satisfy (Proposition 1).
- We discuss two special cases. With exponentially distributed computation time, the optimal threshold is given in closed-form. With exponentially distributed transmission time, a piece-wise bisection method is proposed to find the optimal threshold.
- In the numerical results, we study the performance of the proposed policies. We use exponentially distributed transmission time and two computation time distributions, namely exponential and Pareto, to study the minimum PAoI and optimal thresholds. Our experimental results validate the analytical conclusions, and demonstrate that as the ratio of the mean transmission time to the mean computation time increases, the optimal threshold monotonically increases, while the minimum PAoI first decreases and then increases.

The remainder of the paper is structured as follows. In Section II, we introduce the MEC system model and formulate the problem of minimizing the average PAoI. Section III focuses on the optimal policy and two special cases. In Section IV, we present numerical results, and in Section V, we draw conclusions based on our findings.

II. System Model And Problem Formulation

A. System Model

We consider a status update system as shown in Fig. 1, where a source generates real-time status packets and sends them to an edge server through a channel. The source generates and submits packets to the channel at times $S_0, S_1, \ldots$. Each packet arrives at the edge server after a random transmission time is served with a random computation time, and then delivered to the destination. Packets arriving while the edge server is busy can be stored in a queue. The source is aware of the idle/busy state of the channel and the edge server through the feedback link between the source and the edge server. For example, the edge server sends 0 when a new packet arrives and 1 when the packet finishes computing. We assume the transmission time of the feedback signal is negligible.

Intuitively, discarding updates in the queue when a new update arrives at the server can lead to a lower AoI. However, this makes a policy of sending as much as possible potentially optimal, which results in a large number of packets being dropped. Therefore, in this paper, we consider an FCFS queue and the source can only submit new data when the queue is empty. Furthermore, we consider a unit-sized buffer for two primary reasons: 1) Simplicity and Analytical Convenience: Under the assumption of a unit-sized buffer, the information source sends new data packets when the buffer is empty to prevent data loss. This simplification allows for a more straightforward analysis of the impact of sampling strategies on peak age without the complexities introduced by the presence of data in the buffer. 2) Performance Advantage: Another critical rationale for considering a unit-sized buffer relates to performance. In scenarios where the buffer contains a substantial amount of data, transmitting new data may become suboptimal. This is because new data may need to wait for each data item in the buffer to be serviced individually, leading to a decrease in the freshness of the new data. In contrast, a unit-sized buffer mitigates this issue by limiting the amount of data that can be stored, ensuring timely transmission of new data.

In order to fully utilize the transmission resource and hence potentially reduce the AoI, the source can submit a new packet when the channel is idle and the queue is empty, even if the edge server is busy. Unfortunately, the server may still be busy when a new packet arrives since the transmission time is random. The newly arrived packet is stored into the unit-sized queue if the edge server is busy and waits until the server becomes idle. During the waiting period, the source does not generate new packets. Thus, all packets will be successfully delivered.

Suppose the update k is generated and submitted at time S_k, its transmission time is T_k, and its computation time is C_k. Assume the transmission times of the packets are i.i.d. with a positive finite mean $0 < E[T] < \infty$, and make the same assumptions for the computation times. Let $F_T(\cdot), f_T(\cdot)$ and $F_C(\cdot), f_C(\cdot)$ denote the cumulative distribution function and probability density function of the transmission time and computation time, respectively. Since both T and C are positive, we have $f_T(t) = f_C(t) = 0$ for all $t \leq 0$. Denote W_k as the waiting time of packet k in the queue. Hence, packet k

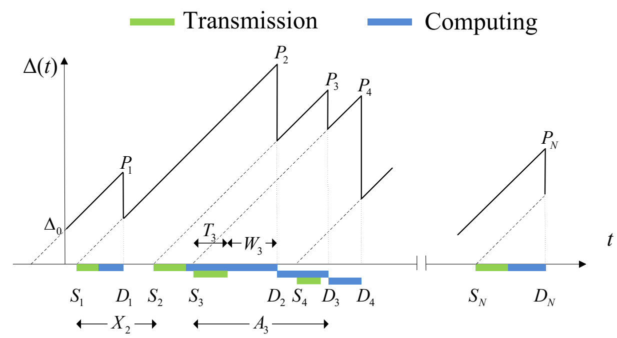

Fig. 2: Age curve in MEC systems.

is delivered at time

$$D_k = S_k + T_k + W_k + C_k. \tag{1}$$

We assume that packet 0 is submitted at time $S_0 = -T_0 - C_0$ and is delivered at $D_0 = 0$. The inter-generation time between packet k and packet $k-1$, denoted by X_k, is given by

$$X_k = S_k - S_{k-1}. \tag{2}$$

Since the source can only submit packets when the channel is idle and the queue is empty, we have

$$X_k \geq T_{k-1} + W_{k-1}. \tag{3}$$

A scheduling policy can be written as $\pi = \{X_k, k \geq 1\}$.

At time t, the AoI at the destination, denoted by $\Delta(t)$, is given by

$$\Delta(t) = t - \max_{k \in \mathbb{N}} \{S_k | D_k \leq t\}. \tag{4}$$

The initial AoI $\Delta(0) = T_0 + C_0$. To calculate the PAoI, we denote

$$A_k = D_k - S_k \tag{5}$$

to be the system time, which is the time spent by the k-th packet in the channel and the edge server. Then, the PAoI can be calculated by

$$P_k = X_k + A_k. \tag{6}$$

The evolution of the instantaneous age is given in Fig. 2.

B. Problem Formulation

Under a given scheduling policy π, the average PAoI is defined as

$$\mathcal{P}_\pi = \lim_{K \to \infty} \frac{1}{K} \mathbb{E} \left[\sum_{k=1}^{K} P_k \right]. \tag{7}$$

As we focus on the set of policies where the source can only submit packets when the channel is idle and the queue is empty, i.e., $\Pi = \{\pi = \{X_k, k >= 0\} | X_{k+1} >= T_k + W_k\}$, the problem of minimizing the average PAoI can be formulated as

$$\mathcal{P}^* = \min_{\pi \in \Pi} \quad \mathcal{P}_\pi, \tag{8}$$

Our goal is to find the optimal policy in Π, denoted by π^*, that achieves the minimum average PAoI $\mathcal{P}^*$. In the following, we consider the optimal policy step by step.

III. The Optimal Scheduling Policy

In this section, we solve problem (8) in three steps. First, we prove that a *continuously working* policy is optimal for this problem. Second, we prove that a *randomized threshold* policy is optimal, and third, we show that each randomized threshold policy is outperformed by a *fixed threshold* policy. Finally, we develop a low-complexity algorithm to find the optimal fixed threshold policy.

A. The Optimality of Continuous Working Policies

Based on definitions in Sec. II-A, we have $P_k = X_k + A_k = X_k + T_k + W_k + C_k$. The PAoI benefits from reducing X_k and W_k. Intuitively, reducing the inter-generation time X_k increases the waiting time W_k and thus may degrade the PAoI performance. However, if an idle time exists between the delivery of the last packet D_{k-1} and the generation of a new packet S_k, it is always beneficial to reduce this idle time by reducing X_k. We express this idea in the following lemma.

Lemma 1. *In the optimal policy, the source submits a new packet before or immediately when the old packet is delivered to the destination, which means $S_k \le D_{k-1}$ for all $k = 1, 2, \cdots$.*

Proof. The details are provided in Appendix A □

Lemma 1 shows that idle time in the system will increase the average PAoI. This conclusion shows the difference between PAoI and AoI as indicators of information freshness: appropriate idle time may reduce the average AoI [8] but only increases the PAoI. By substituting (1) and (2) into $S_k \le D_{k-1}$, we have the following result.

Corollary 1. *The optimal policy for the problem* (8) *satisfies $X_{k+1} \le T_k + W_k + C_k$.*

Based on Corollary 1, we focus on the following special policies.

Definition 1 (Continuous Working Policy)**.** *A policy is said to be a Continuous Working Policy, if it satisfies $X_{k+1} \le T_k + W_k + C_k$ for all k.*

Let Π_{CW} ($\Pi_{\text{CW}} \subseteq \Pi$) denote the set of continuous working policies. In the following, we focus on a special policy in Π_{CW}.

B. The Optimality of Randomized Threshold Policies

A policy $\pi \in \Pi_{\text{CW}}$ is said to be a *randomized threshold* policy if it chooses a threshold Θ_k from an invariant distribution $f_\Theta(\cdot)$ and then determines X_{k+1} according to $X_{k+1} = T_k + W_k + \min\{\Theta_k, C_k\}$. This means that the new update is generated at most Θ_k times later than the previous one starts to be computed. We use Π_{RT} ($\Pi_{\text{RT}} \subseteq \Pi_{\text{CW}}$) to represent the set of randomized threshold policies. The first key result of this paper is the following theorem.

Theorem 1. *Given the distribution of transmission time and computation time, a randomized threshold policy in Π_{RT} is optimal for Problem* (8)*.*

Proof. The details are provided in Appendix B. □

Based on Theorem 1, we can reformulate Problem (8) as follows:

$$\begin{aligned} P_{k+1} &= X_{k+1} + A_{k+1} \\ &= T_k + W_k + \min\{\Theta, C_k\} + T_{k+1} + W_{k+1} + C_{k+1} \end{aligned}$$

where $W_k = \max\{0, D_{k-1} - S_k - T_k\} = \max\{0, C_{k-1} - \Theta_{k-1} - T_k\}$. Since T_k's ,C_k's and Θ_k's are i.i.d., we have that W_k's are also i.i.d. Thus, by dropping the subscripts from all random variables and replacing numerical average with expectation, Problem (8) can be reformulated as the following functional optimization problem:

$$\begin{aligned} \mathcal{P}^* = \min_{f_\Theta} & \mathbb{E}\left[\min\{\Theta, C\} + 2W\right] + 2\mathbb{E}\left[T\right] + \mathbb{E}\left[C\right] \qquad (9) \\ s.t. \quad & W = \max\{0, \widetilde{C} - \Theta - T\}. \end{aligned}$$

where $\widetilde{C}$ represents the computation time of the last packet.

Theorem 1 shows that we can optimize average PAoI without considering historical information, such as transmission times, computation times of old packets and inter-generation times. Next, we consider the optimal policy in Π_{RT}.

C. The Optimality of Fixed Threshold Policies

A policy $\pi \in \Pi_{\text{RT}}$ is said to be a *fixed threshold* policy, if Θ degrades to a constant value, denoted by θ. In this case, $X_{k+1} = T_k + W_k + \min\{\theta, C_k\}$ for all $k = 1, 2, \cdots$. Let Π_{FT} ($\Pi_{\text{FT}} \subseteq \Pi_{\text{RT}}$) denote the set of fixed threshold policies.

Theorem 2. *Given the distribution of transmission time and computation time, a fixed threshold policy in Π_{FT} is optimal for Problem* (9)*.*

Proof. The details are provided in Appendix C. □

Theorem 2 shows that the source can apply a simple policy to minimize the average PAoI: when a packet starts to be processed, the source sends the new packet after waiting for a constant time θ or after the data is processed. In the following, we study the value of the minimum average PAoI.

Denote $\mathcal{P}^*(\theta)$ as the average PAoI in the fixed threshold policy with parameter θ. We have

$$\mathcal{P}^*(\theta) = \mathbb{E}\left[\min\{\theta, C\} + 2W\right] + 2\mathbb{E}\left[T\right] + \mathbb{E}\left[C\right], \qquad (10)$$

where

$$\mathbb{E}\left[\min\{\theta, C\}\right] = \theta \int_\theta^\infty f_C(x)\, dx + \int_0^\theta f_C(x) x\, dx, \qquad (11)$$

$$\mathbb{E}\left[W\right] = 2\int_0^\infty f_T(x) \int_{x+\theta}^\infty f_C(y)(y - \theta - x)\, dy\, dx. \qquad (12)$$

Based on Theorems 1 and 2, the Problem (8) can be reformulated as

$$\mathcal{P}^* = \min_{\theta \in [0,\infty) \cup \{\infty\}} \mathcal{P}^*(\theta). \qquad (13)$$

Assume that f_C is continuous[2], Problem (13) is a one-dimensional optimization of a continuous, differentiable function. Thus, the minimum point is either a boundary point (0 or ∞) or a local minimum. In particular, two special fixed threshold policies are defined: if $\theta = 0$, the source sends a new packet when the old packet begins to be computed; if $\theta = \infty$, the source sends a new packet when the computing process of the old one is finished. By substituting $\theta = 0$ and $\theta = \infty$ into (11) and (12), we have

$$\mathcal{P}^*(0) = 2\int_0^\infty f_T(x)\int_x^\infty f_C(y)(y-x)\,dy\,dx + 2\mathbb{E}[T] + \mathbb{E}[C] \tag{14}$$

$$\mathcal{P}^*(\infty) = 2\mathbb{E}[T] + 2\mathbb{E}[C] \tag{15}$$

The local minimum may exist in multiple, and they all satisfy the following condition.

Proposition 1. *The local minimum $\theta^\dagger$ of Problem* (13) *satisfy*

$$2\mathbb{E}_T\left[F_C(x+\theta^\dagger)\right] = F_C(\theta^\dagger) + 1, \tag{16}$$

and

$$\mathbb{E}_T\left[f_C(T+\theta^\dagger)\right] \geq \frac{f_C(\theta^\dagger)}{2}. \tag{17}$$

Proof. The equations are directly derived based on the first-order condition $\frac{d\mathcal{P}^*(\theta)}{d\theta} = 0$ and the second-order condition $\frac{d^2\mathcal{P}^*(\theta)}{d\theta^2} \geq 0$. □

For any given distributions of T and C, the local minimum $\theta^\dagger$ can be found by solving (16) and checking if (17) is satisfied. The optimal threshold θ^* is the value among $0, \theta^\dagger, \infty$ that achieves the minimum average PAoI $\mathcal{P}^* = \min\{\mathcal{P}^*(0), \mathcal{P}^*(\theta^\dagger), \mathcal{P}^*(\infty)\}$.

D. Special Cases

In the following, we focus on two special cases: 1) when the computation time follows an exponential distribution, the optimal policy θ^* is always either 0 or ∞; 2) when the transmission time follows an exponential distribution, we can calculate θ^* using a low complexity algorithm developed based on equations (16) and (17).

1) Exponentially Distributed Computation Time: If the computation time follows an exponential distribution with parameter μ, which means $f_C(x) = \mu e^{-\mu x}, x > 0$. From (11) and (12), we have $\mathcal{P}^*(\theta)$ is first-order differentiable. Denote $P'(\theta)$ as the first derivative of $\mathcal{P}^*(\theta)$, then we have

$$\begin{aligned} P'(\theta) &= \int_\theta^\infty f_C(x)\,dx - 2\int_0^\infty f_T(x)\int_{x+\theta}^\infty f_C(y)\,dy\,dx \\ &= e^{-\mu\theta}(1 - 2\int_0^\infty f_T(x)e^{-\mu x}\,dx) \\ &= e^{-\mu\theta}(1 - 2\mathcal{L}_\mu), \end{aligned}$$

where $\mathcal{L}_\mu$ is the Laplace transform of the transmission time distribution. It can be easily seen that $P'(\theta) \neq 0$ for any θ if $\mathcal{L}_\mu \neq 1/2$. Therefore, the minimum must be achieved at the boundaries. In particular, $\theta^* = 0$ if $\mathcal{L}_\mu \leq \frac{1}{2}$ and $\theta^* = \infty$ if $\mathcal{L}_\mu > \frac{1}{2}$.

2) Exponentially Distributed Transmission Time: Assume that the transmission time follows an exponential distribution with parameter λ, which means $f_T(x) = \lambda e^{-\lambda x}, x > 0$. We denote the second derivative of $\mathcal{P}^*$ by $P''(\theta)$. In general, there is no closed-form expression for θ^* for a continuous computation time distribution. However, we can first numerically calculate $\theta^\dagger$ and then compare $\mathcal{P}^*(\theta^\dagger), \mathcal{P}^*(0)$, and $\mathcal{P}^*(\infty)$ to obtain θ^*.

To calculate $\theta^\dagger$, we need to find all $\theta \in (0,\infty)$ that satisfy (16) and (17). To achieve this, we define

$$P_z''(\theta) = \lambda(1 - F_C(\theta)) - f_C(\theta). \tag{18}$$

Then, we have the following lemma.

Lemma 2. *When the transmission time follows an exponential distribution, for any θ that satisfies* (16)*, we have*

$$P''(\theta) = P_z''(\theta). \tag{19}$$

Proof. The details are provided in Appendix D. □

Since $P_z''(\theta) = 0$ is generally easier to solve compared with (16), we can calculate the local minimum based on $P_z''(\theta)$ as shown in the following proposition.

Proposition 2. *When the transmission time follows an exponential distribution and $f_C(x)$ is continuous, suppose $0 \leq \theta_1 < \theta_2 < \infty$, we have*

- *If $P_z''(\theta) > 0$ for all $\theta_1 < \theta < \theta_2$, there is at most one $\theta \in (\theta_1, \theta_2)$ that satisfies* (16)*, and if such a θ exists, it also satisfies* (17)*.*
- *If $P_z''(\theta) < 0$ for all $\theta_1 < \theta < \theta_2$, there is no $\theta \in [\theta_1, \theta_2]$ that satisfies* (16)*.*

Proof. The details are provided in Appendix E. □

Based on Proposition 2, we can calculate the local minimum as follows. Denote $\theta_1, \theta_2, \cdots$ as all zero points of $P_z''(\theta)$. These zero points divide $(0,\infty)$ into multiple sub-intervals. On each sub-interval, if $P_z''(\theta) > 0$, we can use the bisection method to calculate θ which satisfies (16) and (17). Then, by comparing the corresponding average PAoI of these θ's, we can obtain $\theta^\dagger$, and finally obtain θ^*. The procedure is shown in Algorithm 1.

IV. Numerical Results

In this section, we examine the performance of the optimal policies when the transmission time is exponentially distributed and the computation time is exponential and Pareto distributed. We set $\mathbb{E}[T + C] = 1$ and change the ratio between $\mathbb{E}[T]$ and $\mathbb{E}[C]$ in the simulations. For comparison, we adopt the zero-wait policy, mean-threshold policy and long-wait policy as baselines, which are all special fixed threshold policies that use 0, the mean of the computation time, and ∞ as the threshold θ, respectively.

[2]For the case of non-continuous or discrete distribution, a similar analysis can be launched by introducing the Dirac impulse function.

Algorithm 1 Piece-wise Bisection Method for Solving Problem (13) for Exponentially Distributed Transmission Time

given $\theta_1, \cdots, \theta_M$
$\mathbb{S} \leftarrow \emptyset$
for $(l, r) \leftarrow \{(0, \theta_1), \cdots, (\theta_M, \infty)\}$
    if $P_z''(\theta) > 0, \theta \in (l, r)$
        find θ by bisection method such that (16) holds
        $\mathbb{S} \leftarrow \mathbb{S} \cup \{\mathcal{P}^*(\theta)\}$
$\theta^\dagger \leftarrow \arg\min_\theta \mathbb{S}$
$\theta^* \leftarrow \arg\min_\theta \{\mathcal{P}^*(\theta^\dagger), \mathcal{P}^*(0), \mathcal{P}^*(\infty)\}$
return θ^*

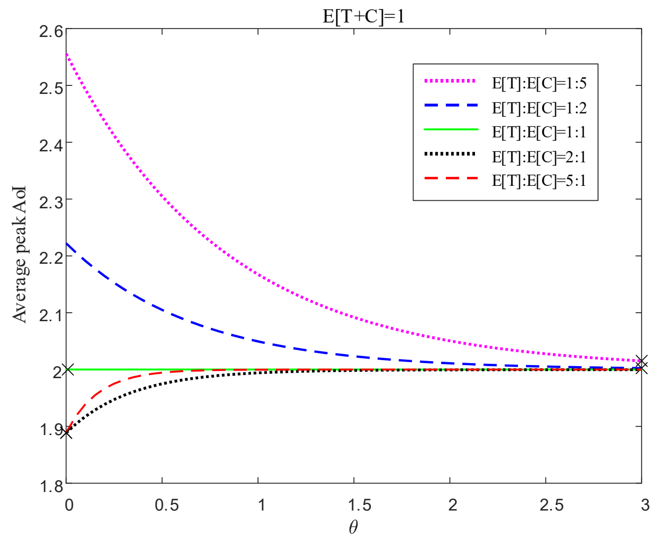

Fig. 3: Average PAoI vs. θ under the fixed-threshold policy with exponential transmission and computation time for different $\mathbb{E}[T] : \mathbb{E}[C]$.

A. Exponential Transmission Time and Computation Time

In this example, we consider both transmission and computation times to be exponentially distributed with parameters λ and μ, respectively. The mean transmission time and computation time are given by $\mathbb{E}[T] = \frac{1}{\lambda}$ and $\mathbb{E}[C] = \frac{1}{\mu}$, respectively. In Fig. 3, we present the average PAoI under the fixed-threshold policy for different $\frac{\mathbb{E}[T]}{\mathbb{E}[C]}$ values. The optimal threshold θ^* that achieves the minimum PAoI is indicated by black cross signs.

As discussed in Sec.III-D, when $\frac{\mathbb{E}[T]}{\mathbb{E}[C]} > 1$, we have $\lambda : \mu < 1$, and then the Laplace transform $\mathcal{L}_\mu = \frac{\lambda}{\lambda+\mu} < \frac{1}{2}$. Consequently, the average PAoI performance of the optimal policy strictly decreases with increasing θ, and the optimal threshold is $\theta^* = \infty$. Conversely, when $\frac{\mathbb{E}[T]}{\mathbb{E}[C]} < 1$, as shown in Fig. 3, the average PAoI is increasing and $\theta^* = 0$. When $\frac{\mathbb{E}[T]}{\mathbb{E}[C]} = 1$, the average PAoI is constant for all threshold values.

The average PAoI curve exhibits a starkly different trend between $\frac{\mathbb{E}[T]}{\mathbb{E}[C]} > 1$ and $\frac{\mathbb{E}[T]}{\mathbb{E}[C]} < 1$. When $\frac{\mathbb{E}[T]}{\mathbb{E}[C]} < 1$, the average PAoI decreases as the threshold increases, indicating that the waiting time of packets caused by the computation process is the main factor that causes the data packets to become stale. Conversely, when $\frac{\mathbb{E}[T]}{\mathbb{E}[C]} > 1$, the average PAoI increases as the threshold increases, indicating that transmission delay is the main factor in this case. Therefore, the PAoI decreases as the transmission interval decreases when the threshold is small.

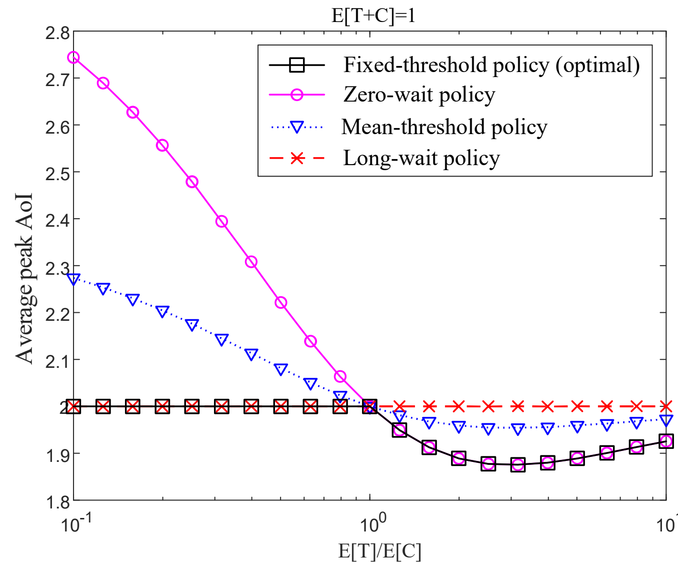

Fig. 4: Average PAoI achieved by different policies under exponential transmission and computation time with varying $\frac{\mathbb{E}[T]}{\mathbb{E}[C]}$.

In Fig. 4, we compare the performance of the fixed-threshold policy, the zero-wait policy, the mean-threshold policy, and the long-wait policy by varying $\frac{\mathbb{E}[T]}{\mathbb{E}[C]}$. The experimental results validate our analysis in Sec. III-D1: the optimal threshold θ^* takes 0 or ∞ when the computation time follows the exponential distribution. It can be observed that as $\frac{\mathbb{E}[T]}{\mathbb{E}[C]}$ increases, the average PAoI of the optimal policy shows a trend of decreasing first and then increasing. When $\frac{\mathbb{E}[T]}{\mathbb{E}[C]} \approx 3$, the average PAoI reaches its minimum value.

B. Exponential Transmission Time and Pareto Computation Time

In this example, we assume the computation time to follow a Pareto distribution, which is characterized by the parameters (x_m, α). x_m and α are the scale and shape parameters, respectively. The Pareto distribution is a heavy-tailed distribution, where the smaller the α, the heavier the tail. This distribution is chosen because update computation times are often related to the length of the update, which frequently follows a Pareto distribution. The probability density function of the Pareto distribution is given by $f(x) = \frac{\alpha x_m^\alpha}{x^{\alpha+1}}, x > x_m$. We set $\alpha = 2$ and vary $\mathbb{E}[C]$ by changing x_m, where $\mathbb{E}[C] = \frac{\alpha x_m}{\alpha - 1}$.

Fig. 5 shows the average PAoI under the optimal policy for different values of $\frac{\mathbb{E}[T]}{\mathbb{E}[C]}$. We observe that when $\frac{\mathbb{E}[T]}{\mathbb{E}[C]}$ is large, the PAoI monotonically increases. However, when $\frac{\mathbb{E}[T]}{\mathbb{E}[C]}$ is small, the PAoI curve is not monotonic, and has multiple local minima. The optimal threshold θ^*, still marked by a black cross, decreases as $\frac{\mathbb{E}[T]}{\mathbb{E}[C]}$ increases. The same phenomenon

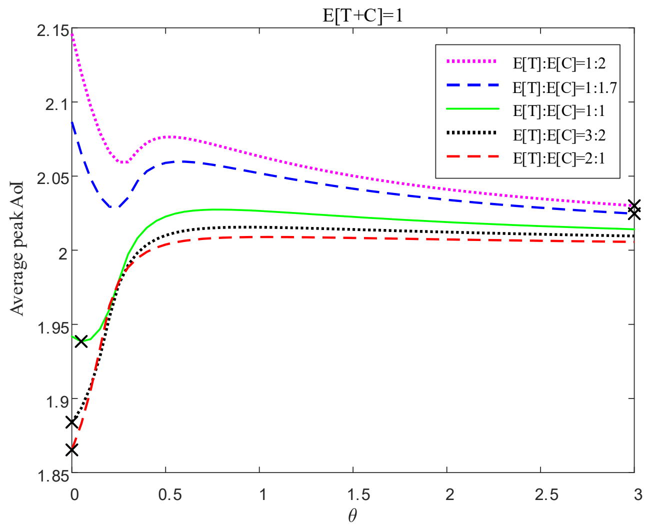

Fig. 5: Average PAoI vs. θ under exponential transmission and Pareto computation time for different $E[T] : E[C]$.

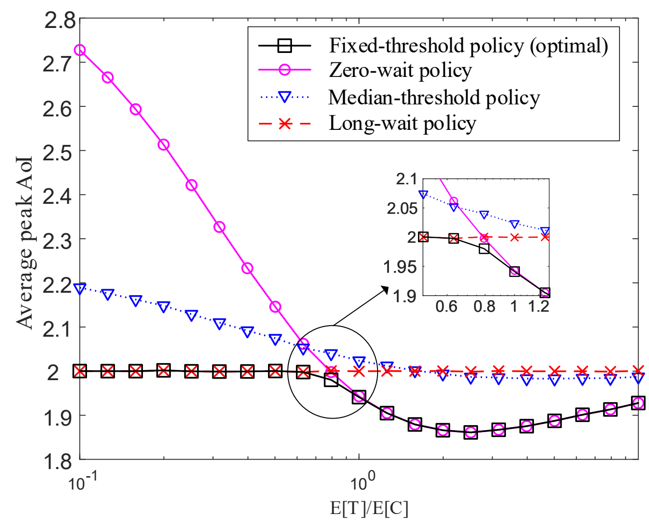

Fig. 6: Average PAoI achieved by different policies under Pareto transmission and exponential computation time with varying $\frac{\mathbb{E}[T]}{\mathbb{E}[C]}$ values.

holds true in the first experiment. This phenomenon is intuitive in MEC systems, because the update frequency of the source should be decreased when the computation time of the edge server increases to avoid an increase in age caused by updates piling up in the queue.

In Fig. 6, we compare the performance of the fixed threshold policy and other policies by varying $\frac{\mathbb{E}[T]}{\mathbb{E}[C]}$. We observe that when $\frac{\mathbb{E}[T]}{\mathbb{E}[C]}$ is very small or very large, the PAoI performance of all policies is close to the results in the first experiment. However, when $\frac{\mathbb{E}[T]}{\mathbb{E}[C]} \approx 0.8$, the fixed-threshold policy outperforms other policies, indicating that the optimal threshold is between 0 and ∞. This verifies our conclusion in Fig. 5: the optimal threshold decreases as $\frac{\mathbb{E}[T]}{\mathbb{E}[C]}$ increases. In this case, as $\frac{\mathbb{E}[T]}{\mathbb{E}[C]}$ increases, the PAoI still decreases first and then increases.

V. Conclusions

In this paper, we investigate the problem of minimizing the PAoI in MEC systems with transmission delay and computation time. We show that a fixed threshold policy can achieve the minimum PAoI. We provide the optimal threshold directly or develop a low-complexity algorithm to compute it when the transmission or computation time follows the exponential distribution. To validate our theoretical analysis, we conduct experiments and observe that the optimal threshold increases as the ratio of the mean transmission delay to the mean service time grows. Our analytical findings can provide valuable insights for designing scheduling algorithms in MEC systems. Future research can focus on optimizing AoI, multi-source systems, or multi-user MEC systems.

Appendix

A. Proof of Lemma 1

We prove this lemma by contradiction. Suppose that under the optimal policy $\pi \in \Pi$, packet $k+1$ is submitted G_k seconds after packet k is delivered, which means $S_{k+1} = D_k + G_k$. using $X_k = S_k - S_{k-1}$, we can obtain the following expression:

$$\begin{aligned} P_k &= X_k + T_k + W_k + C_k \\ &= S_k - S_{k-1} + T_k + W_k + C_k. \end{aligned}$$

Now we construct a new policy π' that satisfies $S_j^{'} = S_j$ for all $j < k$ and $S_j^{'} = S_j - G_k$ for all $j \geq k$. In this policy, $G_k^{'} = 0$. The new policy has no impact on packets before packet k, thus $P_j^{'} = P_j$ for all $j < k$. The new policy π' submits packet k and all subsequent packets in advance. Since packet k is still submitted when the queue is empty and the edge server is idle in policy π', the waiting time of these packets remains unchanged. Therefore, we have $W_j^{'} = W_j$ for all $j \geq k$. Then we have $P_k^{'} = P_k - G_k$ and $P_j^{'} = P_j$ for $j > k$.

According to the definition of the average PAoI in (7), policy π' achieves a lower average PAoI than policy π. This contradicts the assumption that π is optimal. Therefore, the optimal policy must satisfy $S_k \leq D_{k-1}$ for all $k = 1, 2, \cdots$ in the optimal policy.

B. Proof of Theorem 1

We prove this theorem by showing that any continuous working policy can be outperformed by a randomized threshold policy. Our approach is inspired by [19], but we make adjustments and adaptations to their method to account for the different optimization objectives and models we consider.

Under a continuous working policy, the action of packet k is $X_{k+1} = T_k + W_k + \min\{\Xi_k, C_k\}$, where Ξ_k is a random value determined by all history information $\{S_0, T_0, C_0, W_0, \cdots, S_{k-1}, T_{k-1}, C_{k-1}, W_{k-1}\}$ before packet k, denoted by $\mathcal{H}_k$, packet k's transmission time duration T_k and packet k's waiting time W_k. Specifically, the

$$\lim_{N\to\infty}\frac{1}{N}\mathbb{E}\left[\sum_{k=0}^{N-1}P_{k+1}\right]=\lim_{N\to\infty}\frac{\sum_{k=0}^{N-1}\mathbb{E}\left[P_{k+1}\right]}{N} \tag{20}$$

$$=\lim_{N\to\infty}\frac{\sum_{k=0}^{N-1}\mathbb{E}\left[T_k+W_k+\min\{\Xi_k,C_k\}+T_{k+1}+W_{k+1}+C_{k+1}\right]}{N} \tag{21}$$

$$=\lim_{N\to\infty}\frac{\sum_{k=0}^{N-1}\mathbb{E}\left[\min\{\Xi_k,C_k\}+2W_{k+1}\right]}{N}+2\mathbb{E}\left[T\right]+\mathbb{E}\left[C\right]+\lim_{N\to\infty}\frac{\mathbb{E}\left[T_0+W_0-T_N-W_N\right]}{N} \tag{22}$$

$$=\lim_{N\to\infty}\frac{\sum_{k=0}^{N-1}\mathbb{E}\left[\min\{\Xi_k,C_k\}+2W_{k+1}\right]}{N}+2\mathbb{E}\left[T\right]+\mathbb{E}\left[C\right] \tag{23}$$

$$=\lim_{N\to\infty}\frac{\sum_{k=0}^{N-1}\mathbb{E}_{\mathcal{H}_k}\left[\mathbb{E}_{T_k}\left[\mathbb{E}_{W_k}\left[\widehat{R}_k(T_k,W_k,\mathcal{H}_k)|W_k\right]|T_k\right]|\mathcal{H}_k\right]}{N}+2\mathbb{E}\left[T\right]+\mathbb{E}\left[C\right] \tag{24}$$

$$\geq\lim_{N\to\infty}\frac{\sum_{k=0}^{N-1}\mathbb{E}_{\mathcal{H}_k}\left[R^*(\mathcal{H}_k)|\mathcal{H}_k\right]}{N}+2\mathbb{E}\left[T\right]+\mathbb{E}\left[C\right] \tag{25}$$

$$\geq\lim_{N\to\infty}\frac{\sum_{k=0}^{N-1}R_{\min}}{N}+2\mathbb{E}\left[T\right]+\mathbb{E}\left[C\right] \tag{26}$$

$$\geq R_{\min}+2\mathbb{E}\left[T\right]+\mathbb{E}\left[C\right] \tag{27}$$

---

source first observes $\mathcal{H}_k$, T_k and W_k, and then chooses Ξ_k by a conditional probability $p(\gamma,\eta,\boldsymbol{h})\triangleq Pr(\Xi_k|T_k=\gamma,W_k=\eta,\mathcal{H}_k=\boldsymbol{h})$. By the definition of W_k we have

$$W_k=\max\{0,D_{k-1}-S_k-T_k\} \tag{28}$$

$$=\max\{0,C_{k-1}-\Xi_{k-1}-T_k\}. \tag{29}$$

Since T_k's and C_k's are independent of the scheduling policy, a policy can be equivalently written as $\pi=\{\Xi_k,i\geq 1\}$. The $k+1$-th peak AoI is

$$P_{k+1}=X_{k+1}+T_{k+1}+W_{k+1}+C_{k+1} \tag{30}$$

$$=T_k+W_k+\min\{\Xi_k,C_k\}+T_{k+1}+W_{k+1}+C_{k+1}. \tag{31}$$

To facilitate the analysis, we define a special value for packet k as

$$R_k\triangleq\min\{\Xi_k,C_k\}+2W_{k+1}. \tag{32}$$

Based on (29), T_{k+1} also affects R_k. For packet k and a fixed history $\mathcal{H}_k$, we group all the status updating sample paths that have the same T_k,W_k and perform a statistical averaging over all of them to get the following average R_k:

$$\widehat{R}_k(\gamma,\eta,\boldsymbol{h})\triangleq\mathbb{E}\left[R_k|T_k=\gamma,W_k=\eta,\mathcal{H}_k=\boldsymbol{h}\right], \tag{33}$$

where the expectation is taken for Ξ_k,C_k and T_{k+1}.

Then, we derive a lower bound of the average PAoI as shown in (20)-(27). There, (20) follows by (7); (21) follows by (31); (22) follows by combining the random variables with the same subscript together and the fact that T_k's and C_k's are i.i.d.; (23) follows that: by (29) we have $T_k+W_k\leq C_k$, and then we have

$$\lim_{N\to\infty}\frac{\mathbb{E}\left[T_0+W_0-T_N-W_N\right]}{N}$$
$$\leq\lim_{N\to\infty}\frac{\mathbb{E}\left[T_0+W_0\right]}{N}\leq\lim_{N\to\infty}\frac{\mathbb{E}\left[C_0\right]}{N}=0,$$

and

$$\lim_{N\to\infty}\frac{\mathbb{E}\left[T_0+W_0-T_N-W_N\right]}{N}$$
$$\geq-\lim_{N\to\infty}\frac{\mathbb{E}\left[T_N+W_N\right]}{N}\geq-\lim_{N\to\infty}\frac{\mathbb{E}\left[C_N\right]}{N}=0,$$

; (24) follows by (32); $R^*(\mathcal{H}_k)$ in (25) denotes the minimum value of $\widehat{R}_k(\gamma,\eta,h)$ over all possible T_k and W_k; $R_{\min}$ in (27) denotes the minimum value of $R^*(\mathcal{H}_k)$ over all packets and their corresponding histories, i.e, the minimum over all k and $\mathcal{H}_k$.

Note that in the continuous working policy achieving $R_{\min}$, Ξ_k is determined by a certain distribution $p(\gamma,\eta,\boldsymbol{h})=Pr(\Xi_k|T_k=\gamma,W_k=\eta,\mathcal{H}_k=\boldsymbol{h})$, for a fixed condition, i.e., fixed values of T_k, W_k, and $\mathcal{H}_k$. Now observe that T_i's are i.i.d.. Therefore, if we directly apply the distribution that achieves $R_{\min}$ over all packets without considering the history information, all inequalities in (20)-(27) become equations. The new policy achieves the lower bound and is a fixed threshold policy. This completes the proof.

C. Proof of Theorem 2

For any randomized threshold policy, the average of PAoI as shown in (9) is

$$\begin{aligned}\mathbb{E}[P] =& 2\mathbb{E}\left[T\right]+\mathbb{E}\left[C\right]\\ &+\int_0^\infty f_\Theta(\theta)\bigg(\theta\int_\theta^\infty f_C(x)\,dx+\int_0^\theta xf_C(x)\,dx\\ &+2\int_0^\infty f_T(x)\int_{x+\theta}^\infty f_C(y)(y-\theta-x)\,dy\,dx\bigg)\,d\theta\\ \geq& 2\mathbb{E}\left[T\right]+\mathbb{E}\left[C\right]\\ &+\min_\theta\bigg(\theta\int_\theta^\infty f_C(x)\,dx+\int_0^\theta xf_C(x)\,dx\\ &+2\int_0^\infty f_T(x)\int_{x+\theta}^\infty f_C(y)(y-\theta-x)\,dy\,dx\bigg)\end{aligned}$$

Suppose θ^* achieves the minimum of the above inequality. Then, construct a new policy π' satisfying $Pr(\Theta'=\theta^*)=1$. Note that policy π^* is a fixed threshold policy defined in Sec. III-C. Therefore, any randomized threshold policy is outperformed by a fixed threshold policy. This completes the proof.

D. Proof of Lemma 2

Based on (11) and (12) we have

$$P''(\theta)=2\int_0^\infty f_C(x+\theta)\lambda e^{-\lambda x}\,dx-f_C(\theta) \tag{34}$$

For any θ that satisfies (16), by submitting $f_T(x)=\lambda e^{-\lambda x}, x>0$ into (16), we have

$$\begin{aligned}F_C(\theta)+1&=2\int_0^\infty f_T(x)F_C(x+\theta)\,dx\\ &=2\int_\theta^\infty f_C(y)\int_0^{y-\theta}f_T(x)\,dx\,dy\\ &=2\int_\theta^\infty f_C(y)(1-e^{-\lambda(y-\theta)})\,dy\\ &=2-2\int_\theta^\infty f_C(y)e^{-\lambda(y-\theta)}\,dy\\ &=2-2\int_0^\infty f_C(x+\theta)e^{-\lambda x}\,dx\end{aligned}$$

Finally, we can obtain $P''(\theta)=\lambda(1-F_C(\theta))-f_C(\theta)$ by submitting the above last equation into (34).

E. Proof of Proposition 2

We prove the proposition by contradiction. In the first case, if $P_z''(\theta)>0$ for all $\theta_1<\theta<\theta_2$, we suppose that θ_a,θ_b ($\theta_1<\theta_a<\theta_b<\theta_2$) satisfy (16), and there is no other $\theta\in(\theta_a,\theta_b)$ that satisfies (16). Since $P'(\theta)$ is continuous, there must exist $\theta_c\in(\theta_a,\theta_b)$ such that $P'(\theta_c)=0$. Based on our assumption, we have $P''(\theta_c)<0$. Then, we have $P_z''(\theta_c)=P''(\theta_c)<0$, which contradicts $P_z''(\theta)>0$ for all $\theta_1<\theta<\theta_2$. Thus, if $P_z''(\theta)>0$ for all $\theta_1<\theta<\theta_2$, there is at most one $\theta\in(\theta_1,\theta_2)$ that satisfies (16). If $\theta\in(\theta_1,\theta_2)$ satisfies (16), we have $P''(\theta)=P_z''(\theta)>0$, which means θ also satisfies (17). We complete the proof of the first case.

In the second case, if $P_z''(\theta)>0$ for all $\theta_1<\theta<\theta_2$, we suppose that θ_d ($\theta_1<\theta_d<\theta_2$) satisfies (16) and (17). Then, we can obtain $P_z''(\theta_d)=P''(\theta_d)>0$, which also contradicts $P_z''(\theta)>0$ for all $\theta_1<\theta<\theta_2$. Therefore, if $P_z''(\theta)<0$ for all $\theta_1<\theta<\theta_2$, there is no $\theta\in[\theta_1,\theta_2]$ that satisfies (16). We complete the proof of the second case.

References

[1] Y. Mao, C. You, J. Zhang, K. Huang, and K. Letaief, "A survey on mobile edge computing: The communication perspective," *IEEE Communications Surveys Tutorials*, vol. 19, no. 4, pp. 2322–2358, 2017.

[2] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of things: A survey on enabling technologies, protocols, and applications," *IEEE Communications Surveys Tutorials*, vol. 17, no. 4, pp. 2347–2376, 2015.

[3] S. Kaul, R. Yates, and M. Gruteser, "Real-time status: How often should one update?" in *2012 Proceedings IEEE INFOCOM*. IEEE, 2012, pp. 2731–2735.

[4] B. Bacinoglu, E. Ceran, and E. Uysal-Biyikoglu, "Age of information under energy replenishment constraints," in *2015 Information Theory and Applications Workshop (ITA)*. IEEE, 2015, pp. 25–31.

[5] A. Bedewy, Y. Sun, and N. Shroff, "The age of information in multihop networks," *IEEE/ACM Transactions on Networking*, vol. 27, no. 3, pp. 1248–1257, 2019.

[6] R. D. Yates and S. K. Kaul, "The age of information: Real-time status updating by multiple sources," *IEEE Transactions on Information Theory*, vol. 65, no. 3, pp. 1807–1827, March 2019.

[7] S. K. Kaul, R. D. Yates, and M. Gruteser, "Status updates through queues," in *2012 46th Annual Conference on Information Sciences and Systems (CISS)*, Princeton, NJ, USA, March 2012, pp. 1–6.

[8] Y. Sun, E. Uysal-Biyikoglu, R. D. Yates, C. E. Koksal, and N. B. Shroff, "Update or wait: How to keep your data fresh," *IEEE Transactions on Information Theory*, vol. 63, no. 11, pp. 7492–7508, Nov. 2017.

[9] A. Arafa, J. Yang, S. Ulukus, and H. V. Poor, "Age-minimal transmission for energy harvesting sensors with finite batteries: Online policies," *IEEE Transactions on Information Theory*, vol. 66, no. 1, pp. 534–556, Jan 2020.

[10] J. Gong, J. Zhu, X. Chen, and X. Ma, "Sleep, sense or transmit: Energy-age tradeoff for status update with two-threshold optimal policy," *IEEE Transactions on Wireless Communications*, vol. 21, no. 3, pp. 1751–1765, March 2022.

[11] A. Alabbasi and V. Aggarwal, "Joint information freshness and completion time optimization for vehicular networks," *IEEE Transactions on Services Computing*, vol. 15, no. 2, pp. 1118–1129, 2022.

[12] A. Arafa, R. D. Yates, and H. V. Poor, "Timely cloud computing: Preemption and waiting," in *2019 57th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, 2019, pp. 528–535.

[13] P. Zou, O. Ozel, and S. Subramaniam, "Optimizing information freshness through computation–transmission tradeoff and queue management in edge computing," *IEEE/ACM Transactions on Networking*, vol. 29, no. 2, pp. 949–963, Apr. 2021.

[14] Q. Kuang, J. Gong, X. Chen, and X. Ma, "Analysis on computation-intensive status update in mobile edge computing," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 4, pp. 4353–4366, 2020.

[15] J. Gong, Q. Kuang, X. Chen, and X. Ma, "Reducing age-of-information for computation-intensive messages via packet replacement," in *The 11th Int. Conf. Wireless Commun. Signal Processing (WCSP)*, Oct. 2019.

[16] J. Zhu and J. Gong, "Online scheduling of transmission and processing for aoi minimization with edge computing," in *IEEE INFOCOM 2022 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2022, pp. 1–6.

[17] J. P. Champati, R. R. Avula, T. J. Oechtering, and J. Gross, "Minimum achievable peak age of information under service preemptions and request delay," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 5, pp. 1365–1379, May 2021.

[18] M. Costa, M. Codreanu, and A. Ephremides, "On the age of information in status update systems with packet management," *IEEE Transactions on Information Theory*, vol. 62, no. 4, pp. 1897–1910, Apr. 2016.

[19] A. Arafa, J. Yang, S. Ulukus, and H. V. Poor, "Age-minimal transmission for energy harvesting sensors with finite batteries: Online policies," *IEEE Transactions on Information Theory*, vol. 66, no. 1, pp. 534–556, Jan 2020.