Timely Offloading in Mobile Edge Cloud Systems

Nitya Sathyavageeswaran, Roy D. Yates, Anand D. Sarwate, Narayan Mandayam
Rutgers, The State University of New Jersey
Email: {nitya.s, anand.sarwate}@rutgers.edu, {ryates, narayan}@winlab.rutgers.edu

***Abstract*—Future real-time applications like smart cities will use complex Machine Learning (ML) models for a variety of tasks. Timely status information is required for these applications to be reliable. Offloading computation to a mobile edge cloud (MEC) can reduce the completion time of these tasks. However, using the MEC may come at a cost such as related to use of a cloud service or privacy. In this paper, we consider a source that generates time-stamped status updates for delivery to a monitor after processing by the mobile device or MEC. We study how a scheduler must forward these updates to achieve timely updates at the monitor but also limit MEC usage. We measure timeliness at the monitor using the age of information (AoI) metric. We formulate this problem as an infinite horizon Markov decision process (MDP) with an average cost criterion. We prove that an optimal scheduling policy has an age-threshold structure that depends on how long an update has been in service.**

I. Introduction

The Mobile Edge Cloud (MEC) is an emerging paradigm for computing in mobile environments. Mobile devices may be able to offload computation to a physically close edge cloud server for processing. This can help enable future real-time applications in smart cities [1], autonomous vehicles [2], healthcare [3], and industrial monitoring [4] which use complex machine learning (ML) models. For example, autonomous vehicles run ML algorithms to identify pedestrians, traffic signs, objects and other vehicles, and predict their behaviour to provide a safe driving experience. Low latency is required for these applications, so computing the output of the ML model should be done in a timely manner.

A ML model may compute an output more quickly for some inputs than for others. A mobile device therefore has a choice of whether to offload the evaluation of the model to a MEC or to process it locally. We propose a model for this decision process in which a source (the device) sends time-stamped status updates (the ML model outputs) to a monitor. We use the age of information (AoI) metric [5] to measure the timeliness of these updates. The device, or *scheduler* can choose to process the update locally or offload it, at a cost, to the MEC. Using the MEC can reduce the AoI, yielding a timeliness/cost tradeoff.

The scheduling problem in wireless systems from an AoI, privacy, and energy perspective has been extensively studied. Minimizing the long-run average age from a scheduling perspective for multiple users has been studied [6]–[8]. Finding optimal policies to minimize age using Markov decision theory has been investigated [9]–[12]. From a privacy perspective, He et al. [13] address the privacy concerns associated with using the MEC and provide privacy aware-task scheduling algorithms in MEC systems. Min et al. [14] propose offloading schemes to preserve privacy and save energy consumption of IoT devices. Task offloading with an energy constraint has been studied [15], [16]. There has also been a lot of work to find optimal policies to minimize AoI subject to an energy constraint under different system settings [17]–[22].

In this work, we study the tradeoff between the average age at the monitor and the frequency of MEC usage. We consider a generate-at-will source [17] that generates time-stamped system updates which are sent to the monitor after processing by either the local server or MEC. We formulate the problem as a Markov decision process (MDP) with an average cost criterion, and prove that an optimal policy has an age-threshold structure depending on the amount of time an update has been in service. We also do simulations to compare the optimal policy to a few heuristic polices.

II. System Model and Problem Formulation

Notation. Random variables will be denoted by capital letters and realizations by lower case letters. Let $\mathsf{Geom}(\mu)$ denote the geometric distribution with mean μ: if $X \sim \mathsf{Geom}(\mu)$ then $\mathrm{P}[X = k] = \mu(1-\mu)^{k-1}$, $k = 1, 2, \ldots$.

We study a discrete time system consisting of a source, scheduler, local server, mobile edge cloud (MEC) and a monitor as shown in Figure 1. Time passes in integer slots with slot $n \geq 0$ denoting the time interval $[n, n+1)$. Within a slot n, the following events happen in order:

1) The source may or may not generate an update. The variable $x_n = 1$ if there is an update in slot n and $x_n = 0$ otherwise.
2) The scheduler observes x_n. If $x_n = 1$, the scheduler decides whether to forward the update to the local server or the MEC. If the update is sent to the local server, it enters a first-in-first-out (FIFO) queue to await service. If the update is sent to the MEC, it enters service at the MEC and is serviced immediately.
3) The local server processes the head-of-line update in its queue according to its service distribution.
4) If the MEC or local server have finished servicing an update, it is sent to the monitor.
5) The monitor may or may not request a new update from the source.

In this paper, we study a specific instance of this more general setup. We consider a source that has information about the service facility state. To avoid queuing delays, we consider a generate-at-will source that can submit a fresh update only after the previous update is serviced and received at the monitor.

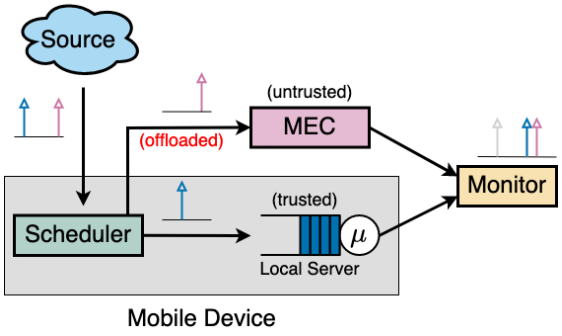

Fig. 1. System model for offloading: Source generates status updates to a monitor after processing by the local server or MEC.

The local server has service distribution $\mathsf{Geom}(\mu)$ and the MEC services an update in 1 time slot. We assume that the transmission time to submit a packet to the MEC is much smaller than the computational time at the MEC, and is thus neglected.

A. Scheduler-controlled source

We allow the scheduler to abort local processing and "pull" a fresh update from the source at the start of the time slot. Thus, if an update is being processed by the local server and has been in service for a while, the scheduler can terminate/abort the local processing and instead pull a fresh update and sent it to the MEC. We model the scheduler's actions over time as a sequence $\mathbf{u} = (u_1, u_2, \ldots)$, where

$$u_n = \begin{cases} 1 & \text{fresh update is sent to MEC} \\ 0 & \text{last update is processed by the local server.} \end{cases} \tag{1}$$

Once the local server or MEC finishes servicing an update, it is sent to the monitor.

B. Problem Formulation

We use AoI as a metric for timeliness. The AoI is defined as the amount of time that has elapsed since the generation of the most recent update that has been received at the monitor. Let $R_n = \max\{k \le n : \text{update } x_k = 1 \text{ received at monitor}\}$ so $A_n = n - R_n$ is the age at the monitor. Because we will model the time for local processing as a random variable, the update times and age at the monitor are random processes.

Our goal is to minimize the average age, defined as[1]

$$\Delta = \lim_{N\to\infty} \frac{1}{N} \sum_{n=1}^{N} (A_n + 1/2). \tag{2}$$

To disincentivize offloading every update, we impose a price for using the MEC. Since $u_n = 1$ when the scheduler offloads, the frequency of using the MEC is

$$\bar{p} = \lim_{N\to\infty} \frac{1}{N} \sum_{n=1}^{N} u_n. \tag{3}$$

[1]Actions occur slot by slot in discrete time but age is averaged continuously over the slot; when $A_n = a$, the average age over slot n is $a + 1/2$.

This price can represent a number of different factors: a privacy loss for revealing the update to the MEC, the bandwidth/energy for transmitting the update to the MEC, or a monetary cost if the MEC is owned by an external service.

Given these ingredients we can formulate our problem as a Markov decision process (MDP) [23]. If $A_n = a_n$ is the age at the monitor at the end of slot n and $Z_n = z_n$ is the amount of service an update has received in slot n, the age a_{n+1} and z_{n+1} in slot $n+1$ can be computed as a function of a_n and z_n. We therefore define the state of the system as $s_n = (a_n, z_n)$. The MDP consists of a tuple $(\mathcal{S}, \mathcal{U}, P, C)$: $s = (a, z) \in \mathcal{S}$, where $a \in \mathbb{N}$ and $z \in \mathbb{N}_0$ are the countably infinite set of states. $u \in \mathcal{U}$ is the action space defined in (1). The transition function P governing the probability of the transition $s_n \to s_{n+1}$ is:

$$s_{n+1} = \begin{cases} (1, 0) & \text{w.p. } 1 \text{ if } u_n = 1 \\ (z_n + 1, 0) & \text{w.p. } \mu \text{ if } u_n = 0 \\ (a_n + 1, z_n + 1) & \text{w.p. } 1 - \mu \text{ if } u_n = 0 \end{cases}. \tag{4}$$

We model the cost C of taking action u in state s as the sum of the age at the monitor and the MEC cost:

$$C(s_n, u_n) = a_n + 1/2 + \lambda u_n, \tag{5}$$

where $\lambda \in (0, \infty)$ is a relative weight. A policy f is a sequence of maps $\{f_n : n \in \mathbb{N}\}$, where $f_n : (a_n, z_n) \to u_n$. We want to find a policy f^* that minimizes the long term average cost:

$$V_f(s) = \limsup_{N\to\infty} \frac{1}{N} \mathrm{E}_f \left[\sum_{n=1}^{N} C(s_n, u_n) \middle| s_0 = s \right]. \tag{6}$$

III. Heuristic Policies

We first analyze a few heuristic policies to understand the age/cost tradeoff.

A. Local Server Only and MEC Only

If we ignore the MEC and use only the local server, $\bar{p} = 0$. This is a discrete time, geometric service zero-wait policy. Using the discrete time version of the age analysis by Yates [17], the average age at the monitor equals

$$\Delta = \frac{4 - \mu}{2\mu}. \tag{7}$$

In this policy, as the service rate μ approaches zero, the age at the monitor blows up. We thus need to look at other policies which use the MEC to reduce the age at the monitor.

A policy that only uses the MEC to process the updates is the same using the local server when $\mu = 1$. Hence it follows from (7) that the average age is $\Delta = 1.5$. For this policy $\bar{p} = 1$, so while it achieves the lowest possible age at the monitor, it may be undesirable if the MEC is costly.

B. Service Threshold Policy

This policy is defined by the service time threshold z^*. If the amount of time an update has spent in service at the start of the slot n is $Z_n = z^*$, then the existing update in service is dropped and the source immediately generates a new update

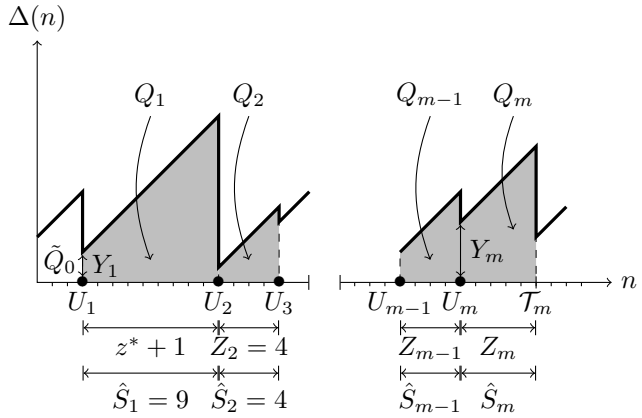

Fig. 2. Sample variation of the age process at the monitor with $z^* = 8$.

for the MEC to process. We find the average age at the monitor for this policy using the graphical approach [5]. Figure 2 shows a sample variation of the age process at the monitor for the service time threshold policy. The source submits updates at times $U_1, U_2, \ldots, U_m$. The age increases linearly with rate 1 in the absence of an update at the monitor and drops to a smaller value Y_i when an update finishes service and is received at the monitor. Y_i is the age at the monitor when an update completes service and is defined as follows:

$$Y_i = \begin{cases} Z_{i-1} & \text{if } Z_{i-1} \leq z^* \\ 1 & \text{if } Z_{i-1} > z^*. \end{cases} \tag{8}$$

Let Z_i denote the service times for the local server. Since either the local server or the MEC can service an update, we define the effective service times $\hat{S}_i$ as follows:

$$\hat{S}_i = \begin{cases} Z_i & \text{if } Z_i \leq z^* \\ z^*+1 & \text{if } Z_i > z^*. \end{cases} \tag{9}$$

Let m denote the number of updates that finish service in time $\mathcal{T}_m$. The average age at the monitor is the area under the saw tooth function in Figure 2 normalized by the total time of observation $\mathcal{T}_m$ given by

$$\mathcal{T}_m = \hat{S}_1 + \hat{S}_2 + \ldots + \hat{S}_m. \tag{10}$$

The average age at the monitor is given by

$$\Delta = \frac{\mathrm{E}[Q_i]}{\mathrm{E}[\hat{S}_i]} = \frac{\mathrm{E}[Y_i\hat{S}_i] + \frac{1}{2}\mathrm{E}[\hat{S}_i^2]}{\mathrm{E}[\hat{S}_i]}. \tag{11}$$

Since Y_i and $\hat{S}_i$ are independent,

$$\begin{aligned} \Delta &= \mathrm{E}[Y] + \frac{1}{2}\frac{\mathrm{E}[\hat{S}^2]}{\mathrm{E}[\hat{S}]} \qquad (12) \\ &= \frac{2\left(1-\bar{\mu}^{z^*}(\mu z^*+\bar{\mu}) - \bar{\mu}^{z^*+1} + \bar{\mu}^{2z^*+1}(\mu z^*+\bar{\mu})\right)}{2\mu\left(1-\bar{\mu}^{z^*+1}\right)} \\ &\quad + \frac{\mu^2\bar{\mu}^{z^*}(z^*+1) + \mu - \bar{\mu}^{z^*}z^* - \mu\bar{\mu}^{z^*}}{2\mu\left(1-\bar{\mu}^{z^*+1}\right)} \end{aligned}$$

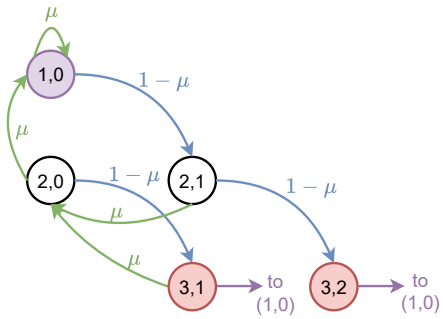

Fig. 3. Markov chain for the age threshold policy with $a^* = 3$.

$$+ \frac{2\bar{\mu} - \bar{\mu}^{z^*+1}(2+z^*\mu)}{2\mu\left(1-\bar{\mu}^{z^*+1}\right)}. \tag{13}$$

Computation of the expectation of Y, $\hat{S}$ and $\hat{S}^2$ is straightforward but tedious, and details on computing (11) and (12) are deferred to the extended version of this manuscript [24].

The frequency of MEC usage is

$$\bar{p} = \lim_{\mathcal{T}_m\to\infty} \frac{\sum_z \mathbb{1}\{Z_i > z^*\}}{\mathcal{T}_m} \tag{14}$$

$$= \lim_{\mathcal{T}_m\to\infty} \frac{\sum_z \mathbb{1}\{Z_i > z^*\}/m}{\mathcal{T}_m/m} = \frac{\mathrm{P}[Z_i > z^*]}{\mathrm{E}\left[\hat{S}\right]} \tag{15}$$

$$= \frac{\mu\bar{\mu}^{z^*}}{1-\bar{\mu}^{z^*+1}}. \tag{16}$$

C. Age Threshold Policy

This policy is defined by the age threshold a^*:

$$u_n = \begin{cases} 1 & \text{if } A_n = a^* \\ 0 & \text{otherwise.} \end{cases} \tag{17}$$

Thus when $A_n = a^*$, the existing update in service is dropped and the source immediately generates a new update for the MEC to process. The MEC processes this update in slot n so the age at the start of the next slot is $A_{n+1} = 1$.

We formulate the (A, Z) Markov chain for this policy to find Δ and $\bar{p}$. Figure 3 shows a Markov chain for the age threshold policy with $a^* = 3$. We find the stationary distribution $\pi(a, z)$ of the Markov chain numerically and use that to find $\mathrm{E}[A] = \sum_{a,z} a\pi(a, z)$. The average age at the monitor for the continuous age function is then given by

$$\Delta = \mathrm{E}[A] + 1/2. \tag{18}$$

Similarly we can also find the MEC frequency, $\bar{p} = \sum_z \pi(a^*, z)$ from the stationary distribution $\pi(a, z)$.

IV. Age Threshold Based Optimal Policy

We now turn to finding the optimal policy f^* which minimizes the long term average cost in (6). We first define the discounted cost $V_{f,\beta}(s)$ as follows:

$$V_{f,\beta}(s) = \mathrm{E}_f\left[\sum_{n=1}^{\infty} \beta^n C(s_n, u_n)\middle| s_0 = s\right], \tag{19}$$

where $s_0 = (1, 0)$ is the initial system state and $\beta \in (0, 1)$ is the discounting factor. We define:

$$V_\beta(s) = \inf_f V_{f,\beta}(s). \tag{20}$$

If the value function $V_\beta(s)$ is finite, we have that $V_\beta(s)$ satisfies the following discounted cost optimality equation:

$$V_\beta(s) = \min_{u \in \mathcal{U}} \Big(C(s, u) + \beta \sum_{s'} P_{ss'}(u) V_\beta(s') \Big), \tag{21}$$

for all $s \in \mathcal{S}$ where $P_{ss'}(u)$ is the transition probability from state s to s' under action u. We can define a value iteration $V_{\beta,n}(s)$ recursively as:

$$\begin{aligned} &V_{\beta,n}(s) \\ &= \min_u \Big(C(s, u) + \beta \sum_{s'} P_{ss'}(u) V_{\beta,n-1}(s') \Big) \qquad (22) \\ &= a + 1/2 + \min \Big(\lambda + \beta V_{\beta,n-1}(1, 0), \\ &\beta\mu V_{\beta,n-1}(z + 1, 0) + \beta(1 - \mu) V_{\beta,n-1}(a + 1, z + 1) \Big), \qquad (23) \end{aligned}$$

for any $n \geq 1$ and $V_{\beta,0}(s) = 0$ for all $s \in \mathcal{S}$ and $\beta \in (0, 1)$.

Sennot's results show that for a finite $V_\beta(s)$ satisfying certain conditions [25, Assumptions 1-3*] , there exists a differential cost function $h(s)$ and a stationary deterministic policy f^* that satisfies the optimality equation

$$g + h(s) = \min_u \Big(C(s, u) + \sum_{s'} P_{ss'}(u) h(s') \Big), \tag{24}$$

for all $s \in \mathcal{S}$, where g is the average cost by following policy f^*, and s' is the resulting state after taking action u. Details showing that Sennot's conditions, such as $V_\beta(s)$ being finite, hold for our problem are in the extended version of this manuscript [24].

A. Structure of the Optimal Policy

Definition 1. *A z-dependent age threshold policy is a stationary deterministic policy of the MDP in which if the action of the policy for state $s = (a, z)$ is to use the MEC, then the action for state $s' = (a + 1, z)$ is also to use the MEC.*

For a z-dependent age threshold policy, for each z there is an age threshold $\bar{a}_z$ given by

$$\bar{a}_z = \min\{a \colon u(a, z) = 1\}, \tag{25}$$

for which we use the MEC.

In the theorem below we show that the optimal policy f^* for the MDP has a z- dependent age threshold structure.

Theorem 1. *There exists a z-dependent age threshold policy of the MDP that minimizes the long term average cost.*

Proof. The proof technique is similar to that by Hsu et al. [26]. We start by showing the age-threshold policy is optimal for the discounted cost case given by (19) and generalize this for the average cost case.

Consider the action of using the MEC in state $s = (a, z)$ to be optimal. Then from (21) we have

$$\beta\mu V_\beta(z + 1, 0) + \beta(1 - \mu) V_\beta(a + 1, z + 1) \geq \lambda + \beta V_\beta(1, 0). \tag{26}$$

Now we need to show that the optimal action for state $s' = (a + 1, z)$ is also to use the MEC. Hence we need to show

$$\beta\mu V_\beta(z + 1, 0) + \beta(1 - \mu) V_\beta(a + 2, z + 1) \geq \lambda + \beta V_\beta(1, 0). \tag{27}$$

We have

$$\begin{aligned} &\beta\mu V_\beta(z + 1, 0) + \beta(1 - \mu) V_\beta(a + 2, z + 1) \\ &\quad \geq \beta\mu V_\beta(z + 1, 0) + \beta(1 - \mu) V_\beta(a + 1, z + 1) \qquad (28) \\ &\quad \geq \lambda + \beta V_\beta(1, 0), \qquad (29) \end{aligned}$$

where (28) follows from the monotonicity property of the value function [24], and (29) follows from (26). This concludes the proof for the discounted case.

We now need to generalize the structure of the optimal policy for the average cost case. Let $\{\beta_k\}_{k=1}^{\infty}$ be a sequence of discount factors converging to 1. Following Sennot [25], the optimal policy for minimizing the total β_k-discounted cost converges to the policy f_k^* for the average cost case. □

Lemma 1. *If the action of the policy for state $s = (a, z)$ is to use the MEC, then the action for state $s = (a, z + 1)$ is also to use the MEC. Also, if the action of the policy for state $s = (a, z)$ is to use the local server, then the action for state $s = (a, z - 1)$ is also to use the local server.*

The proof is similar to that of Theorem 1 and is in an extended version of this manuscript [24]. From Lemma 1 we have

$$\bar{a}_0 \geq \bar{a}_1 \geq \bar{a}_2 \geq \dots \tag{30}$$

This is intuitive because as z increases, we get a better age reduction by using the MEC.

B. Relative Value Iteration

To find the optimal policy f^*, we want to apply the relative value iteration (RVI) method [23]. But our state space is countably infinite and the cost is unbounded. We therefore truncate our state space as follows: if the age of an update goes above some specified tolerance level $a_{\max}$, the scheduler then requests a new update from the source and sends it to the MEC. The state of the system then goes from $(a_{\max}, z)$ to $(1, 0)$. The optimal policy for the truncated MDP and the original MDP are asymptotically identical when $a_{\max} \to \infty$ if the truncated MDP is *unichain* [23]. Our truncated MDP is unichain since any stationary policy f for that chain has only one recurrent class, as the state $(1, 0)$ (recurrent state) can be reached from all other states (a, z). Under any policy, two consecutive service completions has non zero probability following which we are at state $(1, 0)$.

We therefore apply the RVI algorithm to the finite state approximation of our problem. In each iteration of the RVI

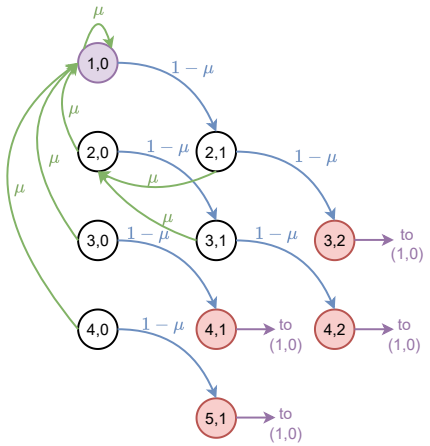

Fig. 4. Markov chain of the optimal policy for $\mu = 0.5$ and $\lambda = 3$. The green lines indicate local service completion. The blue lines indicate that the update is still in service in the local server. The red circles indicate MEC service. $\bar{a}_1 = 4$, $\bar{a}_2 = 3$.

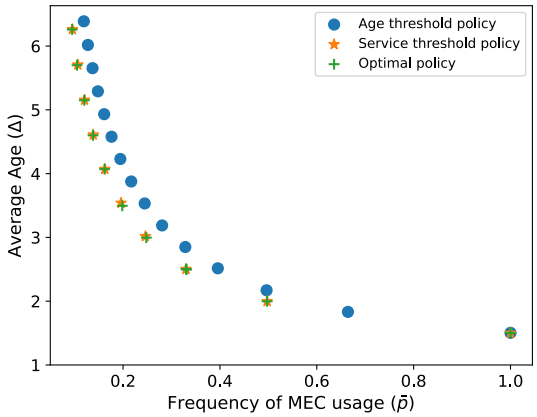

Fig. 5. Age vs. frequency of using the MEC for $\mu = 0.01$. a^* varies from 1 to 15 for the age threshold policy, and z^* varies from 0 to 9 for the service threshold policy. For the optimal policy, $\lambda \in (0, 50)$.

algorithm we compute the value function for each state as follows:

$$V_n(s) = \min_u \left[C(s,u) + \sum_{s'} P_{ss'}(u) V_{n-1}(s') - V_{n-1}(1,0) \right] \tag{31}$$

where, $V_0(s) = 0$ for all $s \in \mathcal{S}$.

Algorithm 1 RVI algorithm given μ and λ

Input: μ, λ
Initialize: $V(s) \leftarrow 0$ for all states $s \in \mathcal{S}$;
while 1 **do**
  for all $s \in \mathcal{S}$ **do**
    if $a = a_{\max}$ **then**
      $u \leftarrow 1$;
    else
      $u \leftarrow \arg\min_{u \in \mathcal{U}} (C(s,u) + \mathrm{E}[V(s')])$;
    end if
    $V_{\text{temp}}(s) \leftarrow \min_u (C(s,u) + \mathrm{E}[V(s')]) - V(1,0)$;
  end for
  $V_{\text{temp}}(s) \leftarrow V(s)$ for all $s \in \mathcal{S}$.
end while

Figure 4 illustrates the optimal policy when the local server service rate, $\mu = 0.5$, and $\lambda = 3$. From Figure 4 we see that the local processor keeps working on an update till the age reaches $\bar{a}_z$. Once the age equals $\bar{a}_z$, the MEC takes over and services a new update that it receives from the source immediately. We see that the age threshold is monotonically non-decreasing in z. In Figure 4, $\bar{a}_1 = 4$ and $\bar{a}_2 = 3$.

C. Simulations

Figure 5 shows the variation of age with the frequency of MEC usage when local service rate, $\mu = 0.01$. For the service threshold and age threshold policies, as z^* or a^* decreases respectively, we use the MEC more often, and the age decreases. As a^*, z^*, λ approach infinity, the age blows up for each of the policies and is equivalent to only using the local server. For $(\bar{p}, \Delta) = (1, 1.5)$ all policies are equivalent to only using the MEC in every slot. The service threshold policy gives us a better age- MEC frequency tradeoff than the age threshold policy. In fact, the service threshold policy is very close to the optimal policy. This is a useful observation since the service threshold policy is easier to implement in practice than the optimal policy, as we need to compute the relative values for each state (a, z) recursively to find the optimal policy.

V. Conclusions and Future Work

In this paper we looked at a Mobile Edge Cloud System in which the MEC helps a local device process updates in a timely way, but with a cost. We provided structural results for the optimal policy that minimizes the long term average cost: this policy applies an age threshold dependent on the amount of time the update has spent in service. Proving that the service threshold policy is a close approximation to the optimal policy is an interesting open question.

Additional future work includes: analyzing systems with multiple users, finding optimal policies (or the structure thereof) for more realistic modelling of the latency of the MEC which accounts for features in real systems such as round-trip time (RTT), multiple edge servers, or network issues such as contention. Ultimately, we hope to add quality of experience (QoE) considerations into our model for specific applications.

Acknowledgements

This material is based upon work supported by the National Science Foundation under grant CNS-2148104 and is supported in part by funds from federal agency and industry partners as specified in the Resilient & Intelligent NextG Systems (RINGS) program.

REFERENCES

[1] M. Mohammadi, A. Al-Fuqaha, M. Guizani, and J.-S. Oh, "Semisupervised deep reinforcement learning in support of iot and smart city services," *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 624–635, 2018.

[2] A. I. Maqueda, A. Loquercio, G. Gallego, N. García, and D. Scaramuzza, "Event-based vision meets deep learning on steering prediction for self-driving cars," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 5419–5427.

[3] R. Miotto, F. Wang, S. Wang, X. Jiang, and J. T. Dudley, "Deep learning for healthcare: review, opportunities and challenges," *Briefings in bioinformatics*, vol. 19, no. 6, pp. 1236–1246, 2018.

[4] V. Nasir and F. Sassani, "A review on deep learning in machining and tool monitoring: Methods, opportunities, and challenges," *The International Journal of Advanced Manufacturing Technology*, vol. 115, no. 9, pp. 2683–2709, 2021.

[5] 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.

[6] I. Kadota, A. Sinha, and E. Modiano, "Optimizing age of information in wireless networks with throughput constraints," in *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*. IEEE, 2018, pp. 1844–1852.

[7] Y. Sun, E. Uysal-Biyikoglu, and S. Kompella, "Age-optimal updates of multiple information flows," in *IEEE INFOCOM 2018-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 2018, pp. 136–141.

[8] Y.-P. Hsu, E. Modiano, and L. Duan, "Scheduling algorithms for minimizing age of information in wireless broadcast networks with random arrivals," *IEEE Transactions on Mobile Computing*, vol. 19, no. 12, pp. 2903–2915, 2020.

[9] Y.-P. Hsu, "Age of information: Whittle index for scheduling stochastic arrivals," in *2018 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2018, pp. 2634–2638.

[10] E. T. Ceran, D. Gündüz, and A. György, "Average age of information with hybrid arq under a resource constraint," *IEEE Transactions on Wireless Communications*, vol. 18, no. 3, pp. 1900–1913, 2019.

[11] H. Tang, J. Wang, L. Song, and J. Song, "Minimizing age of information with power constraints: Multi-user opportunistic scheduling in multi-state time-varying channels," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 5, pp. 854–868, 2020.

[12] A. M. Bedewy, Y. Sun, S. Kompella, and N. B. Shroff, "Optimal sampling and scheduling for timely status updates in multi-source networks," *IEEE Transactions on Information Theory*, vol. 67, no. 6, pp. 4019–4034, 2021.

[13] X. He, J. Liu, R. Jin, and H. Dai, "Privacy-aware offloading in mobile-edge computing," in *GLOBECOM 2017-2017 IEEE Global Communications Conference*. IEEE, 2017, pp. 1–6.

[14] M. Min, X. Wan, L. Xiao, Y. Chen, M. Xia, D. Wu, and H. Dai, "Learning-based privacy-aware offloading for healthcare iot with energy harvesting," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4307–4316, 2018.

[15] Y. Mao, J. Zhang, and K. B. Letaief, "Dynamic computation offloading for mobile-edge computing with energy harvesting devices," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 12, pp. 3590–3605, 2016.

[16] J. Xu and S. Ren, "Online learning for offloading and autoscaling in renewable-powered mobile edge computing," in *2016 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2016, pp. 1–6.

[17] R. D. Yates, "Lazy is timely: Status updates by an energy harvesting source," in *2015 IEEE International Symposium on Information Theory (ISIT)*, 2015, pp. 3008–3012.

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

[19] X. Wu, J. Yang, and J. Wu, "Optimal status update for age of information minimization with an energy harvesting source," *IEEE Transactions on Green Communications and Networking*, vol. 2, no. 1, pp. 193–204, 2018.

[20] S. Feng and J. Yang, "Age of information minimization for an energy harvesting source with updating erasures: Without and with feedback," *IEEE Transactions on Communications*, vol. 69, no. 8, pp. 5091–5105, 2021.

[21] M. A. Abd-Elmagid, H. S. Dhillon, and N. Pappas, "Aoi-optimal joint sampling and updating for wireless powered communication systems," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 11, pp. 14 110–14 115, 2020.

[22] G. Stamatakis, N. Pappas, and A. Traganitis, "Control of status updates for energy harvesting devices that monitor processes with alarms," in *2019 IEEE Globecom Workshops (GC Wkshps)*. IEEE, 2019, pp. 1–6.

[23] M. L. Puterman, "Markov decision processes: Discrete stochastic dynamic programming," 1994.

[24] N. Sathyavageeswaran, R. D. Yates, A. D. Sarwate, and N. Mandayam, "Timely offloading in mobile edge cloud systems," *arXiv preprint arXiv:2405.07274*, 2024.

[25] L. I. Sennott, "Average cost optimal stationary policies in infinite state markov decision processes with unbounded costs," *Operations Research*, vol. 37, no. 4, pp. 626–633, 1989.

[26] Y.-P. Hsu, E. Modiano, and L. Duan, "Age of information: Design and analysis of optimal scheduling algorithms," in *2017 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2017, pp. 561–565.