Timely Probabilistic Data Preprocessing in Mobile Edge Computing

Peng Zou[1], Xianglin Wei[2], Omur Ozel[1], Tian Lan[1] and Suresh Subramaniam[1]
[1]ECE Department, George Washington University, Washington DC, 20052, USA
[2]The 63rd Research Institute, National University of Defense Technology, Nanjing 210007, China
pzou94, xianglinwei, ozel, tlan, suresh@gwu.edu

***Abstract*—A combination of mobile edge computing (MEC) and cloud computing paradigms has the potential to greatly alleviate the challenges facing Internet of Things (IoT). We consider a tiered IoT infrastructure in which data generated by an IoT sensor/device is delivered to a data center for processing through an intermediate MEC server. The MEC server can either directly transmit the data to the data center or pre-process the data and then transmit it to the data center over a shared channel. The goal is to maintain the freshness of the data delivered to the data center. In this paper, we assume a probabilistic model for pre-processing by the MEC server. Sensor data is assumed to be generated as a Poisson process and the transmission times over the two paths are assumed to have general distributions. We use Age of Information (AoI) as a measure of data freshness at the data center. We perform stationary distribution analysis in this system and obtain closed form expressions for average AoI and average peak AoI. We focus on selecting the offloading probabilities in conjunction with the mean service times for each server for optimal operation determined by average AoI and peak AoI. Our numerical results show the effect of path diversity in the selection of best offloading probability and service times.**

I. Introduction

As a burgeoning research area, Mobile Edge Computing (MEC) is known to alleviate the long latency caused by cloud computing. However, an MEC server is usually much less powerful than the servers in the cloud data center. Moreover, the energy budget at a MEC server is usually limited. Therefore, a combination of MEC and cloud computing is a much more promising computing paradigm in a tiered IoT infrastructure. In such a hybrid architecture, an IoT sensor (e.g., a camera) generates data that must be delivered to the data center for timely processing. The data may be either transmitted by the MEC server directly to the data center or pre-processed by the MEC server (which typically reduces the size of the data) and then sent to the data center. In this paper, we analyze the timeliness of such an application using Age of Information (AoI) metric. AoI is a new metric that has been found useful in and related to various Internet of Things (IoT) and cyber-physical system applications which require timely availability of information at the receiving end of a communication system. In the context of our system model, we use AoI to measure the time elapsed for sensor data delivered to the data center starting from its generation.

We provide a typical scenario shown in Fig. 1. In this scenario, an IoT sensor connects to an MEC server (e.g., through a wireless access point not represented in the figure), which also maintains a connection to the data center through the core network. The IoT sensor generates data and it sends them to the MEC server. Once the server gets the data, it can split them into two paths: With probability p, it can compute locally and send the outcome to the cloud as shown in server 1 (this path involves a combination of computation and transmission operations); with probability $(1-p)$, it can offload them to the cloud directly as shown in server 2 (this path involves directly transmitting). The channel between MEC server and data center has total resource W (thought as the bandwidth) and is shared by servers 1 and 2.

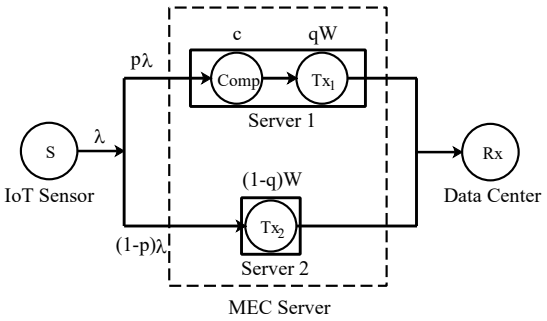

Fig. 1. System model for data pre-processing and offloading in Mobile Edge Computing.

We investigate average AoI and average peak AoI for the system in Fig. 1 with Poisson arrivals of rate λ and generally distributed service times. In line with many recent papers on AoI such as [1]–[4], we assume that there are no data buffers before servers 1 and 2, and therefore data is dropped if it finds the server in busy state. As a result, the servers form two independent M/GI/1/1 queues. The transmission channel of bandwidth W is shared by the two queues with ratio q; hence, servers 1 and 2 have average service rate of qW and $(1-q)W$, respectively. Our goal is to utilize the path diversity in timely transfer of the data from IoT sensor to the data center and determine the probability p and ratio q judiciously for this purpose. We evaluate average AoI and average peak AoI in this system model. In particular, we find integral expressions and calculate the outcome based on service distributions. Numerical results show the effects of path diversity due to different service distributions in selecting best offloading probability in conjunction with mean service times for optimizing average AoI and average peak AoI.

A. Related Work

Age of information (AoI) has found considerable attention in the recent literature as a measure of timeliness of

update information in IoT and edge computing applications. The pioneering work [5] analyzes AoI in queuing models motivated from vehicular networks and it motivated the use of AoI metric in the literature. Among others, [6] considers a general AoI analysis in preemptive and non-preemptive queuing disciplines. See also previous works such as [1]–[3]. The papers [7]–[9] consider AoI in energy harvesting communications. AoI analysis in multi-hop networks have been considered in [4], [10]–[13]. In particular, [4] considers multiple servers with preemption if there is no idle server. [14] considers scheduling data flows in vehicular communication networks. References [15], [16] consider AoI analysis with computing and communication queues. Our work [17], [18] address various computation-communication queues with single transmit server motivated by edge computing. Most recently, [19] considers AoI minimization for MEC under a given deadline. Our previous work in [20] and other works focusing on MEC without AoI metric such as those in [21] constitute motivation for our current work. With respect to existing literature, our study on general service distributions with non-preemptive parallel servers and our focus on peak AoI in relation to average AoI makes the current work unique.

II. System Model

We consider a tiered IoT infrastructure as shown in Figure 1. In this system, there is one IoT sensor representing the source, one MEC server represented by two parallel servers, and one data center at the receiving end. The IoT sensor transmits data to the MEC server through a short-range wireless link with negligible transmission time. At the MEC server, there are no data buffers available and data that finds any server in busy state is dropped.[1] Arriving data can be pre-processed locally at the MEC server and then transmitted to the data center (these two operations are together represented as server 1) or can be offloaded to the data center directly without pre-processing (represented as server 2). We model this system as two servers with different general service distributions and a single source split into two streams *probabilistically*. Additionally, we view the sum of average service rates of these servers constrained due to a common resource such as bandwidth. Our goal is to understand how to determine the splitting ratio judiciously with the service times of servers with general distribution with an objective to feed the receiving end with timely information. To this end, we will use Age of Information (AoI) metric.

A. Data Generation Model

In the IoT sensor, data are generated as a Poisson process with rate λ. This stream is split into two with probability p. The two parallel servers have no buffers. Consequently, we obtain independent M/GI/1/1 systems. Note that since the transmit channel is shared, when $q = 0$ or 1, the system becomes a single server M/GI/1/1 system. The data generated at the IoT sensor has the number of input bits s. Each packet will be pre-processed for a time period c in the MEC server. After computation, each input bit generates o output bits where $o = g(c)$ is a function of the computation time c. We use queue a to denote the queue system with server 1 and queue b to denote the queue system with server 2. Queue a has the arrival rate $p\lambda$ and bandwidth qW bps. Since server 1 is the combination of processor and transmitter, we assume the service time for packets in queue a will be identical, independent, generally distributed random variables with mean $\mathbb{E}[S^a] = c + \frac{so}{qW}$. Note that when $c = 0$, $o = 1$, data entering queue a will be transmitted to the data center without any processing. At the same time, we assume the service time for packets in queue b are identical, independent, exponentially distributed random variables with mean $\mathbb{E}[S^b] = \frac{s}{(1-q)W}$. Corresponding to the general distribution, we have $M_{S^a}(\gamma) \triangleq \mathbb{E}[e^{-\gamma S^a}]$ to denote the moment generating function of the datasize distribution at $-\gamma$ for $\gamma \geq 0$ and $M_{(S^a,1)}(\gamma)$ denotes its first derivative at $-\gamma$, $M_{(S^a,2)}(\gamma)$ denotes its second derivative at $-\gamma$. The main notations used in this paper are listed in Table I.

TABLE I NOTATIONS

Notation	Description
s	The number of input bits in an update
c	The pre-processing time for the update at server 1
o	Number of output bits in update after pre-processing at server 1
W	Bandwidth of the transmission channel
λ	Update generation rate at the IoT device sensor
p	The probability that the updates are pre-processed at MEC server
q	The fraction of transmission rate assigned to server 1.

We let t_i denote the time stamp of the event that packet i enters the system, and t'_i the time stamp of the event that the packet i is delivered to the data center. We index only those packets that enter either one of the servers and not count those that are discarded. The instantaneous AoI is the difference of current time and the time stamp of the packet at the data center:

$$\Delta(t) = t - u(t) \tag{1}$$

where $u(t)$ is the time stamp of the latest packet at the data center at time t.

Each arriving packet is routed probabilistically to either server a with probability p or server b with probability $(1-p)$. An arriving packet may find the server in Idle (Id) or Busy (B) state. If a packet finds the server in (Id), it is dropped; otherwise, its service starts right away. The service times for these queues are independent and they have different distributions. Therefore, a packet that is generated after a packet in service may reach the receiving end earlier and this will make the packet in service non-informative (i.e., will not decrease the age at the receiver). The packets are generated at sensor according to a Poisson process, and the intergeneration times have memoryless exponential distribution. We also define T_i as the effective system time for packet i starting from its entrance to the system until the time it is delivered to the receiver or when it becomes non-informative (whichever happens earlier).

[1]We focus on a zero-buffer system because previous research has shown that excessive queuing in large buffer systems can adversely impact AoI, while limited-buffer systems with packet management can improve AoI [1], [3].

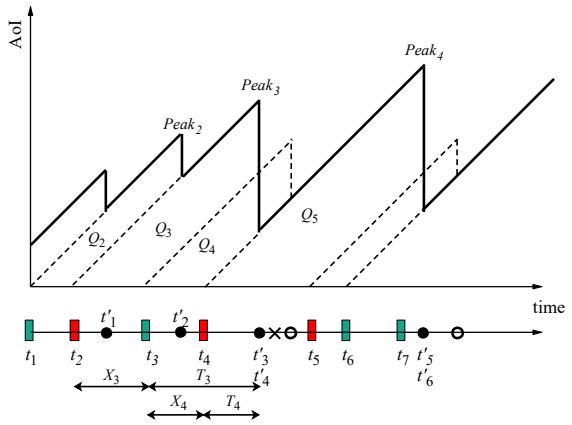

Fig. 2. The evolution of AoI in the system.

Fig. 2 shows a sample path of the AoI evolution. The first packet is generated at t_1 and enters queue a. The second packet is generated at t_2 and enters queue b. The service of first packet and second packet finish at t'_1 and t'_2, respectively. The third packet is generated at t_3 and enters queue a finding it idle. Even though its service finishes between t'_4 and t_5 shown as $\circ$ and any packet that comes in this interval is dropped; its effective service is finished at t'_4 as it becomes non-informative afterwards. Hence, corresponding effective service time is $T_3 = t'_4 - t_3$. Next effective packet is generated at t_5 and enters queue a. Similarly, packet 5 is a non-informative packet and its effective service time is $T_5 = t'_6 - t_5$. This approach is reminiscent of equivalent queues in earlier work [17], [18].

We define the areas Q_i under the triangular regions of the AoI curve as shown in Fig. 2. We then have average AoI:

$$\mathbb{E}[\Delta] = \lambda_e \mathbb{E}[Q] = \lambda_e \left(\mathbb{E}[X_e T] + \frac{\mathbb{E}[X_e^2]}{2} \right), \tag{2}$$

where λ_e is the effective arrival rate and X_e is the corresponding interarrival time. Here T is the effective system time.

III. Average AoI Analysis

Let us define the state of the MEC server as $K_i = (q^a, q^b)$, where $q^a, q^b \in \{(Id), (B)\}$ represent the state of queues a and b, respectively. We have the following stationary probabilities for each queue:

$$\Pr[q^a = (Id)] = \frac{1}{p\lambda D^a_{cycle}}, \Pr[q^a = (B)] = \frac{\mathbb{E}[S^a]}{D^a_{cycle}}, \tag{3}$$

$$\Pr[q^b = (Id)] = \frac{1}{(1-p)\lambda D^b_{cycle}}, \Pr[q^b = (B)] = \frac{\mathbb{E}[S^b]}{D^b_{cycle}}, \tag{4}$$

where $D^a_{cycle} = \frac{1}{p\lambda} + \mathbb{E}[S^a]$ and $D^b_{cycle} = \frac{1}{(1-p)\lambda} + \mathbb{E}[S^b]$ are the expected lengths of one renewal cycle in queue a and queue b, respectively. For convenience, we also denote X_a and X_b as the interarrival times for queues a and b, respectively. Note that X_a and X_b are independent Poisson with rates $p\lambda$ and $(1-p)\lambda$. We additionally denote with R_a and R_b the residual service time a packet observes for a packet in service for queues a and b. R_a and R_b are independent with probability density functions $f_a(r) = \frac{\Pr(S_a > r)}{\mathbb{E}[S_a]}$ and $f_b(r) = \frac{\Pr(S_b > r)}{\mathbb{E}[S_b]}$ for $r \in [0, \infty)$. Additionally,

$$M_{R^a}(\gamma) = \frac{1 - M_{S^a}(\gamma)}{\gamma \mathbb{E}[S^a]}. \tag{5}$$

More generally, the states of queues a and b are independent in this setting and we have:

$$\Pr[K_i = (x, y)] = \Pr[q^a = x] \Pr[q^b = y],$$

where $x, y \in \{(Id), (B)\}$. A packet enters the queue with probability p if $x = (Id)$ and with probability $(1-p)$ if $y = (Id)$. In this case, we have

$$\lambda_e = \lambda p \Pr[q^a = (Id)] + \lambda(1-p) \Pr[q^b = (Id)]$$

To get an expression for average AoI, we next evaluate $\mathbb{E}[Q]$ and put it in (2). For this purpose, we carefully consider conditions over which system variables take different values.

A. Conditioning on $K_{i-1} = ((Id), (Id))$

In this case, packet $i-1$ finds both queues in idle state. To evaluate $\mathbb{E}[Q|K_{i-1} = ((Id), (Id))]$, we evaluate the $\mathbb{E}[X_e T]$ and $\mathbb{E}[X_e^2]$ for the packet $i-1$ entered queue a with probability p. We calculate $\mathbb{E}[X_e T]$ and $\mathbb{E}[X_e^2]$ conditioned on various cases.

1) $X_b < S_a + X_a$*:* In this case, $X_e = X_b$, packet i enters server b and it spends $T = \min\{\tilde{S}_b, S_a + X_a - X_b + \tilde{S}_a\}$ where where $\tilde{S}_b$ is the service time spent in server b, X_a represents the next interarrival time to server a. $\tilde{S}_a$ is the independent service time in server a.

2) $X_b > S_a + X_a$*:* In this case, $X_e = X_a + S_a$, packet i enters server a, and it spends $T = \min\{\tilde{S}_a, X_b - X_a - S_a + \tilde{S}_b\}$ where $\tilde{S}_a$ is the independent service time spent in server a and $\tilde{S}_b$ is the independent service time spent in server b.

With probability $(1-p)$, $X_e = \min\{X_a, S_b + X_b\}$ and we just swap indices for queues a and b in the analysis above to get the other half of the expression. Hence, we have $\mathbb{E}[Q|K_{i-1} = ((Id), (Id))] = pE(a, b) + (1-p)E(b, a)$ where $E(a, b)$ is the corresponding expectation when packet i enters a first. For completeness, the expression for $E(a, b)$ is provided in (6) where integrations with respect to X_a, X_b and S_a are with respect to their probability density functions and the expectations are taken with respect to random variables represented as capital letters. To illustrate how we can use $E(a, b)$, we show the closed form expression for $\mathbb{E}[X_e^2]$ in Appendix A. The computation for the other parts is complex. Therefore, we evaluate the integrals numerically in other cases.

B. Conditioning on $K_{i-1} = ((Id), (B))$

In this case, packet $i-1$ finds the first queue in idle state and the second queue in busy state. We note that the effective packet is the packet entering queue a with probability p. Hence, we calculate $\mathbb{E}[Q]$ conditionally. In this case, the effective inter-arrival time X_e and the system time T behaves under different conditions as follows:

1) $R_b + X_b < S_a + X_a$*:* In this case, $X_e = R_b + X_b$, packet i enters server b and it spends $T = \min\{S_b, S_a + X_a - R_b - X_b + \tilde{S}_a\}$ where X_a is the next independent interarrival time to server a after the service for packet $i-1$ and $\tilde{S}_a$ is the corresponding independent service time.

$$E(a,b) \triangleq \int_0^\infty \int_0^\infty \int_0^{s_a+x_a} \mathbb{E}[x_b \min\{\tilde{S}_b, s_a + x_a - x_b + \tilde{S}_a\}] + \mathbb{E}[\frac{x_b^2}{2}] d(x_b) d(s_a) d(x_a)$$
$$+ \int_0^\infty \int_0^\infty \int_{s_a+x_a}^\infty \mathbb{E}[(x_a + s_a) \min\{\tilde{S}_a, x_b - s_a - x_a + \tilde{S}_b\}] + \mathbb{E}[\frac{(x_a+s_a)^2}{2}] d(x_b) d(s_a) d(x_a) \quad (6)$$

2) $R_b + X_b > S_a + X_a$: In this case, $X_e = X_a + S_a$, packet i enters server a and it spends $T = \min\{\tilde{S}_a, R_b + X_b - S_a - X_a + \tilde{S}_b\}$ where $\tilde{S}_a$ is the independent service time spent in server a, X_b represents the next interarrival time to server b and $\tilde{S}_b$ is the corresponding independent service time.

Finally, we have $\mathbb{E}[Q|K_{i-1} = ((Id),(B))] = pK(a,b)$ where $K(a,b)$ represents the queue entering queue a first and queue b is busy. For completeness, the expression for $K(a,b)$ is provided in (7) where integrations with respect to X_a, X_b, S_a and R_b are with respect to their probability density functions and the expectations are taken with respect to random variables represented as capital letters.

C. Conditioning on $K_{i-1} = ((B),(Id))$

In this case, packet $i-1$ finds the first queue in busy state and the second queue in idle state. We just swap the roles of queues a and b and p with $(1-p)$ to get the final expressions. To illustrate, we have $\mathbb{E}[Q] = (1-p)K(b,a)$. We finally combine all three cases to get

$$\begin{aligned}
&\mathbb{E}[Q](p\text{Pr}[q^a = (Id)] + (1-p)\text{Pr}[q^b = (Id)]) \\
&= \mathbb{E}[Q|K_{i-1} = ((Id),(Id))]\text{Pr}[q^a = (Id)]\text{Pr}[q^b = (Id)] \\
&\quad + \mathbb{E}[Q|K_{i-1} = ((Id),(B))]\text{Pr}[q^a = (Id)]\text{Pr}[q^b = (B)] \\
&\quad + \mathbb{E}[Q|K_{i-1} = ((B),(Id))]\text{Pr}[q^a = (B)]\text{Pr}[q^b = (Id)],
\end{aligned}$$

where we acknowledge that the incoming packets are dropped in all cases of K_{i-1} that remain unaccounted.

IV. Average Peak AoI Analysis

Peak AoI is the peak value of age at the instant right before the status is updated. Compared to the average AoI which represents the timeliness of status updates in the system, the average Peak AoI captures the key events when the status is successfully updated in the system. For the average peak AoI, we depart from the earlier definitions of interarrival and system times. In this case, we account for whether a packet that enters the system is informative or obsolete. We have

$$\mathbb{E}[\Delta^p] = \mathbb{E}[X^{(i)}] + \mathbb{E}[T^{(i)}], \quad (8)$$

where $X^{(i)}$ is the interarrival between two informative packets and $T^{(i)}$ is the system time for an informative packet. The rate of informative packets is

$$\lambda^{(i)} = \lambda P^{(i)},$$

where $P^{(i)}$ is the probability that a task is informative in the system. Let the event E_1 denote $S_a < X_b + S_b$, E_2 denote $S_a < R_b + X_b + S_b$, E_3 denote $S_b < X_a + S_a$ and E_4 denote $S_b < R_a + X_a + S_a$. We have:

$$\begin{aligned}
P^{(i)} &= p\text{Pr}[q^a = (Id)]\text{Pr}[q^b = (Id)]\text{Pr}[E_1] \\
&\quad + p\text{Pr}[q^a = (Id)]\text{Pr}[q^b = (B)]\text{Pr}[E_2] \\
&\quad + (1-p)\text{Pr}[q^b = (Id)]\text{Pr}[q^a = (Id)]\text{Pr}[E_3] \\
&\quad + (1-p)\text{Pr}[q^b = (Id)]\text{Pr}[q^a = (B)]\text{Pr}[E_4]. \quad (9)
\end{aligned}$$

We therefore get $\mathbb{E}[X^{(i)}] = \frac{1}{\lambda^{(i)}}$. Similarly, we have

$$\begin{aligned}
&\mathbb{E}[T^{(i)}]P^{(i)} = p\text{Pr}[q^a = (Id)]\text{Pr}[q^b = (Id)]\mathbb{E}[S_a|E_1]\text{Pr}[E_1] \\
&+ p\text{Pr}[q^a = (Id)]\text{Pr}[q^b = (B)]\mathbb{E}[S_a|E_2]\text{Pr}[E_2] \\
&+ (1-p)\text{Pr}[q^b = (Id)]\text{Pr}[q^a = (Id)]\mathbb{E}[S_b|E_3]\text{Pr}[E_3] \\
&+ (1-p)\text{Pr}[q^b = (Id)]\text{Pr}[q^a = (B)]\mathbb{E}[S_b|E_4]\text{Pr}[E_4]. \quad (10)
\end{aligned}$$

We therefore reach $\mathbb{E}[X^{(i)}]$ and $\mathbb{E}[T^{(i)}]$ to evaluate average peak AoI in (8).

V. Numerical Results

In this section, we provide numerical results for average AoI and average peak AoI. We performed packet-based queue simulations for 10^6 packets as verification of all numerical results. Our goal is to show how splitting probability and bandwidth are assigned to optimize the average AoI and average peak AoI in different settings. For simplicity, we use $d = \frac{s}{W}$ to denote the normalized transmission time and $o = e^{-c}$ relates computation time and datasize.

We use Gamma distributed service time for queue a with mean $\mathbb{E}[S^a]$. In particular, we use the probability density function $f_{S^a}(s_a) = \frac{k^\alpha}{\Gamma(\alpha)} s_a^{\alpha-1} e^{-k s_a}$ for $s_a \geq 0$ where $k = \frac{\alpha}{\mathbb{E}[S^a]}$ and $\alpha > 0$ determines the variance. The variance gets larger as α gets smaller. Indeed, this distribution converges to an impulse at $\mathbb{E}[S^a]$ as α grows large. We have the following closed form expressions for this Gamma distribution: $M_{S^a}(\mu) = \left(1 + \frac{\mu}{k}\right)^{-\alpha}$, $M_{S^{a,1}}(\mu) = \mathbb{E}[S^a]\left(1 + \frac{\mu}{k}\right)^{-\alpha-1}$.

We start with Fig. 3 where we compare the average AoI with respect to p under different bandwidth assignment. We assume $c = 0.2$ which means pre-processing is introduced in server a. We observe that optimal order of splitting probabilities matches with the order of bandwidth assignment. Note that

$$K(a,b) \triangleq \int_0^\infty \int_0^\infty \int_0^{s_a+x_a} \int_0^{s_a+x_a-x_b} \mathbb{E}[(x_b + r_b) \min\{\tilde{S}_b, s_a + x_a - x_b - r_b + \tilde{S}_a\} + \frac{(x_b+r_b)^2}{2}] d(r_b) d(x_b) d(s_a) d(x_a)$$
$$+ \int_0^\infty \int_0^\infty \int_0^{r_b+x_b} \int_0^{r_b+x_b-x_a} \mathbb{E}[(x_a + s_a) \min\{\tilde{S}_a, x_b + r_b - s_a - x_a + \tilde{S}_b\} + \frac{(x_a+s_a)^2}{2}] d(s_a) d(x_a) d(r_b) d(x_b) \quad (7)$$

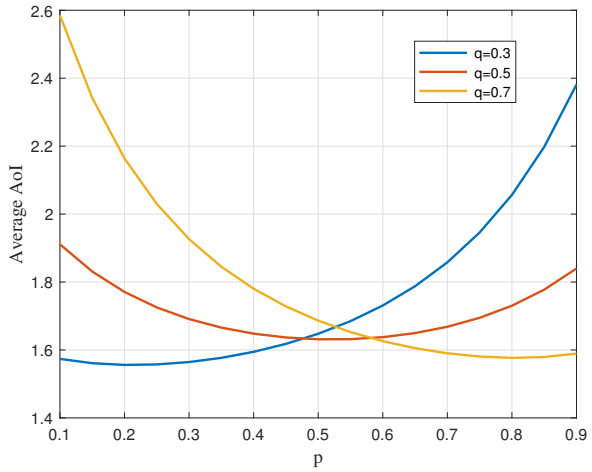

Fig. 3. *Average AoI with respect to p for fixed $\lambda = 2$, $d = 0.5$, $c = 0.2$, $\alpha = 2$.*

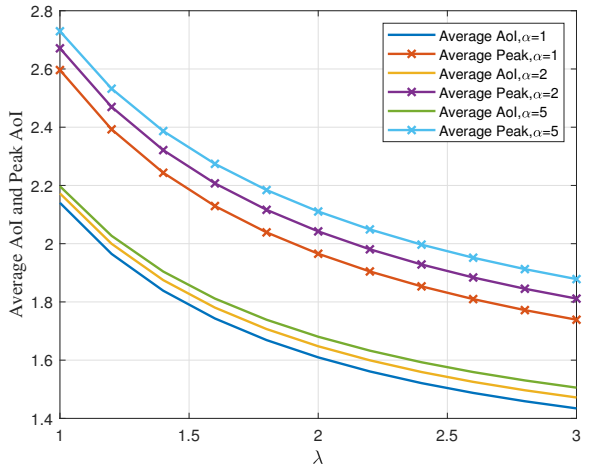

Fig. 4. *Average AoI and Peak AoI with respect to λ for fixed $p = 0.5$, $q = 0.3$, $d = 0.5$, $c = 0.2$.*

when $q = 0.5$, we observe that the curve of average AoI is asymmetric since the pre-processing makes the service rates be different between the two queues.

Next, in Fig. 4, we compare the average AoI and average Peak AoI with respect to λ under different α. We set $p = 0.5$, $q = 0.3$ and $c = 0.2$ here which means the two servers have identical arrival rates but different service rates. We observe that with arrival rate λ increasing, both average AoI and average Peak AoI are monotonic decreasing. Note that with α increasing, both average AoI and average Peak AoI are increasing due to the fact that with α increasing, the variance of service time in queue a is decreasing.

Then we compare the average AoI and average Peak AoI with respect to p under different α in Fig. 5. We set $q = 0.5$ and $c = 0$ here makes the mean service time to be identical for the two servers. Note when $\alpha = 1$, the service time of queue a will be exponentially distributed and we can see the optimal p is around $p = 0.5$ in this case which means when the two servers are identical, sending packets equally will be the optimal assignment for the packets. With α increasing, we

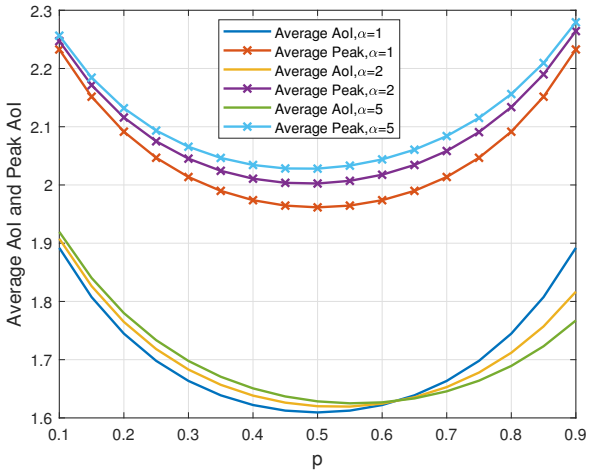

Fig. 5. *Average AoI and Peak AoI with respect to p for fixed $\lambda = 2$, $q = 0.5$, $d = 0.5$, $c = 0$.*

can observe that the optimal p increases, which means with same mean service time, the optimal strategy is to assign more packets to server a when the variance of service time in queue a decreases. On the other hand, we observe that the optimal point for Peak AoI does *not* change significantly due to the fact that the Peak AoI has high correspondence with mean service time. We also observe that with p increasing, more packets will be assigned to server a and $\alpha = 1$ will become the worst case for the value while it is optimal when p is small.

We observe that the optimal values of p could be quite different for average AoI and average peak AoI under fixed q. In general, a decrease in average AoI comes at the cost of increased average peak AoI. To understand the tradeoff between average AoI and average peak AoI, we optimize weighted sum of AoI and average peak AoI:

$$\min_{p \geq 0} \omega_1 \mathbb{E}[\Delta] + \omega_2 \mathbb{E}[PAoI] \tag{11}$$

In Fig 6, we plot the optimal tradeoff obtained by solving the weighted optimization in (11) with fixed $q = 0.5$ for differing service time variances. In particular, for each α, we solve (11) for all possible ω_1 and ω_2 and plot all possible operating points as tuples of average AoI and average peak AoI. This characterizes the optimal tradeoff between average AoI and average peak AoI. Note that when α is increasing, the variance of service time in server a is decreasing and we observe that this tradeoff becomes more apparent for smaller service time variances in queue a.

Finally, we show the average AoI with respect to c under different q in Fig. 7. We observe that when $q = 0.3$ or $q = 0.5$, with little bandwidth assigned to server a, we can optimize average AoI by pre-processing the data and decreasing the transmission time. Interestingly, when $q = 0.7$, the average AoI is a monotonic increasing function of computation time c. This is because if the bandwidth resource is plentiful, pre-processing becomes unnecessary and results in increased AoI.

VI. Conclusion

In this paper, we consider a tiered IoT infrastructure in which data generated by an IoT sensor is delivered to a data

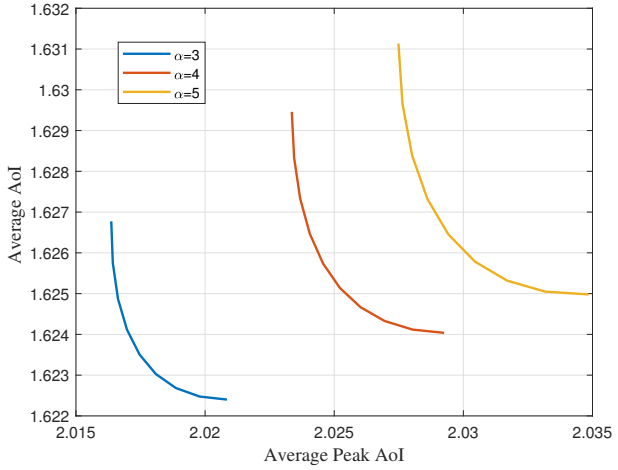

Fig. 6. *Optimal tradeoff curves for average AoI vs. average peak AoI with differing variances and fixed* $\lambda = 2$, $q = 0.5$, $d = 0.5$, $c = 0$.

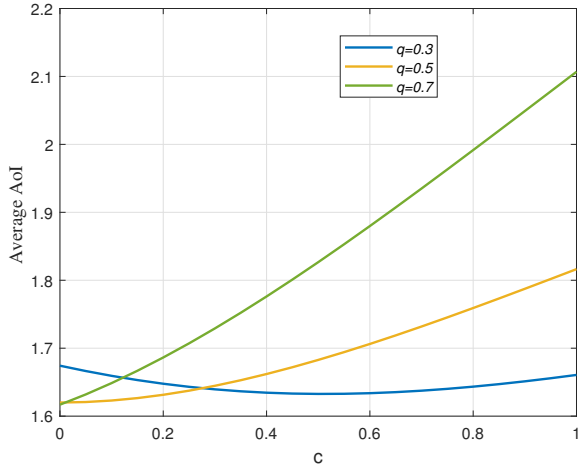

Fig. 7. *Average AoI with respect to* c *for fixed* $\lambda = 2$, $p = 0.5$, $\alpha = 2$, $d = 0.5$.

center for processing through an intermediate MEC server. The MEC server can either transmit the data to the data center or pre-process the data and then transmit it to the data center over a shared channel. The goal is to maintain the freshness of the data delivered to the data center. We assume a probabilistic model for pre-processing by the MEC server and perform stationary distribution analysis in this system to obtain closed form expressions for average AoI and average peak AoI. We focus on selecting the offloading probabilities for optimal operation determined by average AoI and peak AoI. Our numerical results show the effect of path diversity in the selection of best offloading probability and service times. Our future goal is to analyze and optimize the average AoI and average peak AoI in multi-hop and multi-server systems.

APPENDIX

A. $\mathbb{E}[X_e^2]$ *in* $E(a,b)$

We assume $\lambda_a = p\lambda$, $\lambda_b = (1-p)\lambda$ and $\mu_b = \frac{1}{\mathbb{E}[S^b]}$ for these exponential random variables X_a, X_b and S_b separately. Then we have R^b as an exponential random variable with μ_b. We have:

$$\mathbb{E}[X_e^2] \triangleq \frac{2}{\lambda_b^2} - \frac{\lambda_a}{\lambda_b+\lambda_a}\Bigg(\Big(\frac{2}{(\lambda_a+\lambda_b)^2} + \frac{2}{\lambda_b(\lambda_a+\lambda_b)} + \frac{2}{\lambda_b}\Big)M_{S^a}(\lambda_b) + 2\Big(\frac{1}{\lambda_a+\lambda_b} + \frac{1}{\lambda_b}\Big)M_{(S^a,1)}(\lambda_b) + M_{(S^a,2)}(\lambda_b)\Bigg) + \frac{\lambda_a}{\lambda_a+\lambda_b}\Big(\frac{2}{(\lambda_a+\lambda_b)^2}M_{S^a}(\lambda_b) + \frac{2}{\lambda_a+\lambda_b}M_{(S^a,1)}(\lambda_b) + M_{(S^a,2)}(\lambda_b)\Big).$$

REFERENCES

[1] M. Costa, M. Codreanu, and A. Ephremides. On the age of information in status update systems with packet management. *IEEE Transactions on Information Theory*, 62(4):1897–1910, 2016.

[2] C. Kam, S. Kompella, G.D. Nguyen, J.E. Wieselthier, and A. Ephremides. On the age of information with packet deadlines. *IEEE Transactions on Information Theory*, 2018.

[3] C. Kam, S. Kompella, G. D. Nguyen, J. E. Wieselthier, and A. Ephremides. Controlling the age of information: Buffer size, deadline, and packet replacement. In *IEEE MILCOM*, pages 301–306, 2016.

[4] R.D. Yates. Status updates through networks of parallel servers. In *2018 IEEE International Symposium on Information Theory (ISIT)*, pages 2281–2285. IEEE, 2018.

[5] S. Kaul, R. Yates, and M. Gruteser. Real-time status: How often should one update ? In *INFOCOM*, pages 2731–2735. IEEE, 2012.

[6] Y. Inoue, H. Masuyama, T. Takine, and T. Tanaka. A general formula for the stationary distribution of the age of information and its application to single-server queues. *arXiv preprint arXiv:1804.06139*, 2018.

[7] R. Yates. Lazy is timely: Status updates by an energy harvesting source. In *IEEE ISIT*, June 2015.

[8] S. Farazi, A.G. Klein, and D.R. Brown. Average age of information for status update systems with an energy harvesting server. In *IEEE INFOCOM WORKSHPS*, pages 112–117, 2018.

[9] A. Baknina, O. Ozel, J. Yang, S. Ulukus, and A. Yener. Sending information through status updates. In *IEEE ISIT*, 2018.

[10] A. M. Bedewy, Y. Sun, and N. B. Shroff. Age-optimal information updates in multihop networks. *Available at arXiv:1701.05711*, 2017.

[11] R. Talak, S. Karaman, and E. Modiano. Minimizing age-of-information in multi-hop wireless networks. In *Communication, Control, and Computing (Allerton), 2017 55th Annual Allerton Conference on*, pages 486–493. IEEE, 2017.

[12] R.D. Yates. The age of information in networks: Moments, distributions, and sampling. *arXiv preprint arXiv:1806.03487*, 2018.

[13] A. Maatouk, M. Assaad, and A. Ephremides. The age of updates in a simple relay network. *arXiv preprint arXiv:1805.11720*, 2018.

[14] A. Alabbasi and V. Aggarwal. Joint information freshness and completion time optimization for vehicular networks. *CoRR*, abs/1811.12924, 2018.

[15] C. Xu, H. H. Yang, X. Wang, and T.Q.S Quek. On peak age of information in data preprocessing enabled iot networks. *arXiv preprint arXiv:1901.09376*, 2019.

[16] A. Arafa, R.D. Yates, and H.V. Poor. Timely cloud computing: Preemption and waiting. In *Allerton Conference*, pages 528–535, 2019.

[17] P. Zou, O. Ozel, and S. Subramaniam. Trading off computation with transmission in status update systems. In *IEEE PIMRC 2019*, pages 1–6, 2019.

[18] P. Zou, O. Ozel, and S. Subramaniam. Optimizing information freshness through computation-transmission tradeoff and queue management in edge computing. *arXiv preprint arXiv:1912.02692*, 2019.

[19] J. Gong, Q. Kuang, and X. Chen. Joint transmission and computing scheduling for status update with mobile edge computing. *arXiv preprint arXiv:2002.09719*, 2020.

[20] X. Wei, C. Tang, J. Fan, and S. Subramaniam. Joint optimization of energy consumption and delay in cloud-to-thing continuum. *IEEE Internet of Things Journal*, 6(2):2325–2337, 2019.

[21] Y. Gu, Z. Chang, M. Pan, L. Song, and Z. Han. Joint radio and computational resource allocation in iot fog computing. *IEEE Transactions on Vehicular Technology*, 67(8):7475–7484, 2018.