Updates with Multiple Service Classes

Roy D. Yates, Jing Zhong, Wuyang Zhang
WINLAB, Department of Electrical and Computer Engineering
Rutgers University
{ryates, jzhong, wuyang}@winlab.rutgers.edu

***Abstract*—A source submits status update jobs to a service facility for processing and delivery to a monitor. The status updates belong to service classes with different service requirements. We model the service requirements using a hyperexponential service time model. To avoid class-specific bias in the service process, the system implements an M/G/1/1 blocking queue; new arrivals are discarded if the server is busy. Using an age-of-information (AoI) metric to characterize timeliness of the updates, a stochastic hybrid system (SHS) approach is employed to derive the overall average AoI and the average AoI for each service class. We observe that both the overall AoI and class-specific AoI share a common penalty that is a function of the second moment of the average service time and they differ chiefly because of their different arrival rates. We show that each high-probability service class has an associated age-optimal update arrival rate while low-probability service classes incur an average age that is always decreasing in the update arrival rate.**

I. Introduction

Consider a system in which time-stamped raw updates are processed by a service facility for delivery to a monitor. The updates belong to different service time classes – some updates can be processed quickly, while others require longer service times. Perhaps an update occasionally needs a very long service time.

One such example is an augmented reality (AR) system in which images are processed and analyzed, and an update in the form of an image augmentation is returned to the user. Object recognition is typically a key step in a broad range of augmented reality applications. To find a particular object in a given image input, the system extracts key feature points from the image input, and then matches all the feature points with those of the particular object. With a high matching ratio, it assumes the object has been detected [1], [2]. However, when there are a large number of objects in the input, there will be numerous feature points and this will increase the matching complexity and thus the processing time.

In the AR system, time-stamped images are jobs that are the input to a processing system. When an input job is processed, the output, namely an image augmentation, represents an update. The time-stamp of the update is the time-stamp of the image from which it was derived. As timeliness is essential in an AR system, we use the Age-of-Information (AoI) metric [3] for performance evaluation. Specifically, if the newest processed update at time t has time-stamp $u(t)$, the age at the monitor is $\Delta(t) = t - u(t)$.

To model such systems, we adopt a hyperexponential service time model [4] in which the service time belongs to one of c classes such that the service time is exponential (μ_i) with probability p_i, $i = 1, \dots, c$. Without loss of generality, we assume the service rates are ordered such that $\mu_1 \geq \cdots \geq \mu_c$. While service times in a class are memoryless, the composition of the service time classes yields a service time with memory; the longer a service time lasts, the more likely it is that the service time was chosen from a class i with small μ_i.

We note that the hyperexponential model has been used to model long-tailed distributions [4]; specifically, a Weibull distribution was closely approximated by a hyperexponential PDF with $c = 20$ service classes and corresponding service rates that varied over 11 orders of magnitude. While such a range of service times seems likely to be incompatible with a timely updating system, this example serves to demonstrate the flexibility of the hyperexponential model.

In prior work on updates with non-memoryless service times [5]–[7], it has been observed that average AoI can be reduced substantially by a simple preemption-in-service mechanism. Preemption can replace an update that becomes stale while in service with a fresh update and this can substantially reduce the AoI. However, in the context of multiple classes of updates, it is unclear whether this is a desirable approach. Specifically, class i updates and class j updates may be poor substitutes for one another. In the AR system example, a larger processing time would be consistent with a complex image with a large number of objects to classify. In this case, more images to classify would suggest the job is more important.

Based on these considerations, we believe that reducing AoI via preemption in service may be inappropriate for some applications. In particular, preemption in service will be biased in that updates with longer service times are more likely to be preempted. On the other hand, queueing of updates also remains undesirable. Timeliness is improved when the system avoids processing updates that have become stale in a queue. Hence, this work focuses on an M/G/1/1 queueing model with blocking. If the server is busy, new arrivals are blocked and discarded. This mechanism avoids queueing but also avoids a bias against jobs with long service times. Whether an arriving job goes into service (or is blocked) is independent of its service class. Moreover, once a job goes into service, it is guaranteed to finish processing, independent of its class.

A. Related Work

Prior work [8] on the AoI analysis of multi-class queueing systems has examined peak AoI (PAoI) in multiclass M/G/1 and M/G/1/1 queues. Each traffic class is described by its arrival rate, and the first and second moments of its service

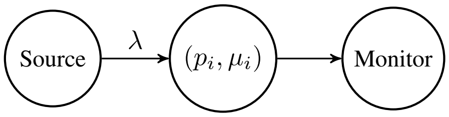

Fig. 1. The updating system with hyperexponetial service .

time, and arrival rates are optimized to minimize $\max_i C_i(A_i)$, where $C_i(A_i)$ is the cost of stream i having PAoI of A_i. In a study of the average AoI for multiple streams arriving at an M/G/1/1 queue with preemption [9], all streams have the same general service time and it is shown that increasing the arrival rate for one class can reduce its AoI, but at the expense of increased AoI for other customers.

This work differs from these prior M/G/1/1 studies in that updates belong to different service classes but they all originate from the same source. The overall update rate λ is a controllable input but the probability p_i that an arriving job is class i is a property of the application scenario. In the AR example, the service rates of the classes would depend on the complexity and variety of the scenario-specific images. Thus we assume that the p_i and μ_i parameters of the hyperexponential model characterize the application scenario. However, the updating system does have the flexibility to adapt the overall job submission rate λ and this specifies the arrival rate $\lambda_i = \lambda p_i$ of class i jobs.

B. Paper Summary

The hyperexponential model permits us to employ the method of stochastic hybrid systems (SHS) for the analysis of age. While AoI analysis of the M/G/1/1 blocking queue has previously appeared [7], SHS enables analysis of the average AoI for each update class.

In Section II, we provide a short introduction to the SHS method. SHS analysis of the hyperexponential service system appears in Section III with derivations of the overall age in Section III-A and the class-specific age in Section III-B. The optimal update arrival rate is studied in Section IV and the paper concludes with a discussion of open issues in Section V.

II. SHS for AoI: Background

A stochastic hybrid system (SHS) [10] has state $[q(t), \mathbf{x}(t)]$ such that $q(t) \in \mathcal{Q} = \{0, \ldots, m\}$ is a continuous-time finite-state Markov chain and $\mathbf{x}(t) \in \mathbb{R}^n$ is a real-valued non-negative row vector that describes the continuous-time evolution of a collection of age-related processes. We will refer to $\mathbf{x}(t)$ as the age vector or AoI process.

For AoI analysis, the SHS approach was introduced in [11], where it was shown that age tracking can be implemented as a simplified SHS with non-negative linear reset maps in which the continuous state is a piecewise linear process [12], a special case of piecewise deterministic processes [13], [14]. In this case, the SHS approach yielded a system of first order ordinary differential equations describing the temporal evolution of the expected value of the age process. For finite-state systems, this led to a set of age balance equations and simple conditions [11, Theorem 4] under which $\mathrm{E}[\mathbf{x}(t)]$ converges to a fixed point.

In this work, we follow [11], but with a small modification from [15] in which the age vector tracks the ages of monitors at specified system locations. We now summarize the basics of this simplified SHS. For the continuous state $\mathbf{x}(t)$, the jth component $x_j(t)$ is the age at an observer/monitor that sees time-stamped updates that pass through a position j in the system or network. In short, x_j is the age of the freshest update observed at position j. An observer sees new (fresher) updates arrive in transitions of the discrete state. In the absence of a fresher arriving update, the age $x_j(t)$ at each observer grows at unit rate. Thus, in each discrete state $q(t) = q$, the continuous state evolves according to $\dot{\mathbf{x}}(t) = \mathbf{1} = [1\ 1\ \cdots\ 1]$.

In the graphical representation of the Markov chain $q(t)$, each state $q \in \mathcal{Q}$ is a node and each transition l is a directed edge (q_l, q'_l) with transition rate $\lambda^{(l)}\delta_{q_l,q(t)}$. The Kronecker delta function $\delta_{q_l,q}$ ensures that transition l occurs only in state q_l. For each transition l, there is transition reset mapping that can induce a discontinuous jump in the continuous state $\mathbf{x}(t)$. For AoI analysis, we employ a linear mapping of the form $\mathbf{x}' = \mathbf{x}\mathbf{A}_l$. That is, transition l causes the system to jump to discrete state q'_l and resets the continuous state from $\mathbf{x}$ to $\mathbf{x}' = \mathbf{x}\mathbf{A}_l$. For tracking of the age process, the transition reset maps are binary: $\mathbf{A}_l \in \{0,1\}^{n\times n}$. The linear mappings $\mathbf{A}_l$ will depend on the specific network system and the definition of the age vector $\mathbf{x}(t)$.

The transition rates $\lambda^{(l)}$ describe the continuous-time Markov chain for $q(t)$ but there are some differences. Unlike an ordinary continuous-time Markov chain, the SHS may include self-transitions in which the discrete state is unchanged because a reset occurs in the continuous state. Furthermore, for a given pair of states $i, j \in \mathcal{Q}$, there may be multiple transitions l and l' in which the discrete state jumps from i to j but the transition maps $\mathbf{A}_l$ and $\mathbf{A}_{l'}$ are different.

It will be sufficient for average age analysis to define for all $\hat{q} \in \mathcal{Q}$,

$$\pi_{\hat{q}}(t) = \mathrm{E}\big[\delta_{\hat{q},q(t)}\big], \tag{1a}$$

$$v_{\hat{q}j}(t) = \mathrm{E}\big[x_j(t)\delta_{\hat{q},q(t)}\big], \quad 1 \le j \le n, \tag{1b}$$

and the vector functions

$$\mathbf{v}_{\hat{q}}(t) = [v_{\hat{q}1}(t), \ldots, v_{\hat{q}n}(t)] = \mathrm{E}\big[\mathbf{x}(t)\delta_{\hat{q},q(t)}\big]. \tag{1c}$$

We note that $\pi_{\hat{q}}(t) = \mathrm{E}\big[\delta_{\hat{q},q(t)}\big] = \mathrm{P}[q(t) = \hat{q}]$ is simply the probability of the discrete Markov state $\hat{q}$.

We assume the Markov chain $q(t)$ is ergodic since time-average age analysis otherwise makes little sense. Under this assumption, the state probability vector $\boldsymbol{\pi}(t) = [\pi_0(t)\ \cdots\ \pi_m(t)]$ always converges to the unique stationary vector $\bar{\boldsymbol{\pi}} = [\bar{\pi}_0\ \cdots\ \bar{\pi}_m]$ satisfying

$$\bar{\pi}_{\bar{q}} \sum_{l\in\mathcal{L}_{\bar{q}}} \lambda^{(l)} = \sum_{l\in\mathcal{L}'_{\bar{q}}} \lambda^{(l)}\bar{\pi}_{q_l}, \quad \bar{q} \in \mathcal{Q}, \tag{2a}$$

$$\sum_{\bar{q}\in\mathcal{Q}} \bar{\pi}_{\bar{q}} = 1. \tag{2b}$$

If $\boldsymbol{\pi}(t) = \bar{\boldsymbol{\pi}}$, it is shown [11] that $\mathbf{v}(t) = [\mathbf{v}_0(t)\ \cdots\ \mathbf{v}_m(t)]$ obeys a system of first order differential equations. When this system is stable, each $\mathbf{v}_{\bar{q}}(t) = \mathrm{E}\big[\mathbf{x}(t)\delta_{\bar{q},q(t)}\big]$ converges to a limit $\bar{\mathbf{v}}_{\bar{q}}$ as $t \to \infty$. In this case,

$$\mathrm{E}[\mathbf{x}] \equiv \lim_{t\to\infty}\mathrm{E}[\mathbf{x}(t)] = \lim_{t\to\infty}\sum_{\bar{q}\in\mathcal{Q}}\mathrm{E}\big[\mathbf{x}(t)\delta_{\bar{q},q(t)}\big] = \sum_{\bar{q}\in\mathcal{Q}}\bar{\mathbf{v}}_{\bar{q}} \quad (3)$$

is the vector of average ages at the set of observers. With

$$\mathcal{L}'_{\bar{q}} = \{l \in \mathcal{L} : q'_l = \bar{q}\}, \qquad \mathcal{L}_{\bar{q}} = \{l \in \mathcal{L} : q_l = \bar{q}\} \quad (4)$$

denoting the respective sets of incoming and outgoing transitions for each state $\bar{q}$, the following theorem provides a simple way to calculate this average age vector.

Theorem 1: [11, Theorem 4] If the discrete-state Markov chain $q(t)$ is ergodic with stationary distribution $\bar{\boldsymbol{\pi}}$ and we can find a non-negative solution $\bar{\mathbf{v}} = [\bar{\mathbf{v}}_0 \ \cdots \bar{\mathbf{v}}_m]$ such that

$$\bar{\mathbf{v}}_{\bar{q}}\sum_{l\in\mathcal{L}_{\bar{q}}}\lambda^{(l)} = \mathbf{1}\bar{\pi}_{\bar{q}} + \sum_{l\in\mathcal{L}'_{\bar{q}}}\lambda^{(l)}\bar{\mathbf{v}}_{q_l}\mathbf{A}_l, \quad \bar{q}\in\mathcal{Q}, \quad (5a)$$

then the vector of average ages is given by

$$\mathrm{E}[\mathbf{x}] = \sum_{\bar{q}\in\mathcal{Q}}\bar{\mathbf{v}}_{\bar{q}}. \quad (5b)$$

In the next section, we use Theorem 1 to find both the overall age and the class-specific ages for the updating system with hyperexponential service times.

III. SHS Analysis

The SHS's for tracking the overall age or the class-specific age have much in common. We start by describing these common elements. In both cases, we define $x_1(t)$ as the age of an observer that sees fresh updates that enter service and $x_2(t)$ as the age at the monitor that sees processed jobs that complete service. Thus $n = 2$ and the continuous state is $\mathbf{x} = [x_1 \ x_2]$.

The discrete state space is $\mathcal{Q} = \{0, \ldots, c\}$ such that the server is idle in state 0 and a class i job is in service in states $i \in \{1, \ldots, c\}$. The service is non-preemptive but when the server is busy, new arriving jobs are discarded to prevent queueing from causing jobs to become stale.

Associated with each state $i > 0$ is an incoming arrival transition $l = 2i-1$ from state 0 of rate $\lambda_i = p_i\lambda$ that marks a class i job going into service. Similarly, state i also has a rate μ_i departure transition, with index $l = 2i$, back to the idle state for the corresponding service completion. For non-preemptive processing with discarding of new arrivals when the server is busy, the SHS Markov chain is shown in Figure 2. When a transition l occurs, the continuous state jumps from $\mathbf{x}$ to $\mathbf{x}' = \mathbf{x}\mathbf{A}_l$. The SHS's for the overall age and the class-specific age differ in how $\mathbf{x}(t)$ is defined. We now describe these differences.

A. Overall Age

For the overall average age, every processed job, no matter what class, yields an update that reduces the age. These transitions are shown in Table I. We see from the table that $\mathbf{A}_l$ is the same matrix $\mathbf{A}$ for each arrival transition l. In the mapping $\mathbf{x}' = \mathbf{x}\mathbf{A}$, the age at the input to the service facility is set to $x'_1 = 0$ since the new update is fresh. However, x'_2 is unchanged since no arrival completes service in the transition. Similarly, $\mathbf{A}_l$ is the same matrix $\mathbf{D}$ for each departure transition. Specifically, the transition $\mathbf{x}' = \mathbf{x}\mathbf{D}$ leaves $x'_1 = x_1$ unchanged but resets $x'_2 = x_1$ because the age x_1 update is delivered to the monitor. To summarize,

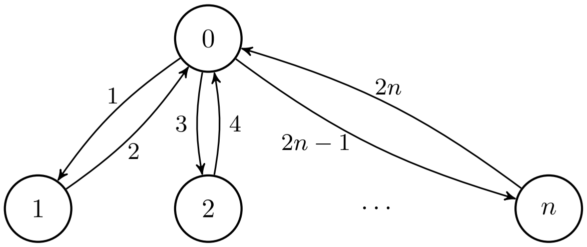

Fig. 2. SHS Markov chain for the M/G/1/1 blocking system with hyperexponential service times.

$$\mathbf{A}_l = \begin{cases}\mathbf{A} = \left[\begin{smallmatrix}0 & 0\\ 0 & 1\end{smallmatrix}\right] & l = 2i-1,\\ \mathbf{D} = \left[\begin{smallmatrix}1 & 1\\ 0 & 0\end{smallmatrix}\right] & l = 2i.\end{cases} \quad (6)$$

To employ Theorem 1, we first observe that $\bar{\pi}_i\mu_i = \bar{\pi}_0\lambda_i$, implying $\bar{\pi}_i = \bar{\pi}_0\rho_i$, where $\rho_i = \lambda_i/\mu_i$ is the offered load of class i jobs. In terms of the total offered load

$$\rho = \sum_{i=1}^{c}\rho_i, \quad (7)$$

(2b) implies the stationary state probabilities are

$$\bar{\pi}_0 = \frac{1}{1+\rho}, \quad \bar{\pi}_i = \frac{\rho_i}{1+\rho}, \quad i > 0. \quad (8)$$

From (5a), we have at $\bar{q} = 0$ and at $\bar{q} = i \in \{1, \ldots, c\}$ that

$$\bar{\mathbf{v}}_0\lambda = \bar{\pi}_0\mathbf{1} + \sum_{i=1}^{c}\mu_i\bar{\mathbf{v}}_i\mathbf{D}, \quad (9)$$

$$\bar{\mathbf{v}}_i\mu_i = \bar{\pi}_i\mathbf{1} + \lambda_i\bar{\mathbf{v}}_0\mathbf{A}. \quad (10)$$

Substituting (10) into (9) and observing that $\mathbf{1}\mathbf{D} = \mathbf{1}$ and $\mathbf{A}\mathbf{D} = \mathbf{0}$, it follows from (8) that

$$\bar{\mathbf{v}}_0 = \frac{1}{\lambda}\left(\frac{\mathbf{1}}{1+\rho} + \sum_{i=1}^{c}\frac{\rho_i\mathbf{1}}{1+\rho}\right) = \frac{\mathbf{1}}{\lambda}. \quad (11)$$

Since $\mathbf{1}\mathbf{A} = [0 \ 1]$, it follows from (10) and (11) that

$$\bar{\mathbf{v}}_i = \frac{1}{\mu_i}\left(\frac{\rho_i\mathbf{1}}{1+\rho} + \frac{\lambda_i}{\lambda}[0 \ 1]\right). \quad (12)$$

From Theorem 1, the average ages of observers at the input and output of the service facility are

$$\mathrm{E}[\mathbf{x}] = \sum_{i=0}^{c}\bar{\mathbf{v}}_i = \frac{\mathbf{1}}{\lambda} + \frac{\mathbf{1}}{1+\rho}\sum_{i=1}^{c}\frac{\rho_i}{\mu_i} + \frac{\rho}{\lambda}[0 \ 1]. \quad (13)$$

Thus, the average age at the monitor is

$$\Delta = \mathrm{E}[x_2] = \frac{1+\rho}{\lambda} + \sum_{i=1}^{c}\frac{\rho_i}{1+\rho}\frac{1}{\mu_i}. \quad (14)$$

l	$q_l \to q'_l$	$\lambda^{(l)}$	$\mathbf{x}\mathbf{A}_l$	$\mathbf{A}_l$	$\mathbf{v}_{q_l}\mathbf{A}_l$
1	$0 \to 1$	λ_1	$[0 \;\; x_2]$	$\left[\begin{smallmatrix}0&0\\0&1\end{smallmatrix}\right]$	$[0 \;\; v_{02}]$
2	$1 \to 0$	μ_1	$[x_1 \;\; x_1]$	$\left[\begin{smallmatrix}1&1\\0&0\end{smallmatrix}\right]$	$[v_{11} \;\; v_{11}]$
$\vdots$	$\vdots$	$\vdots$	$\vdots$		
$2n-1$	$0 \to n$	λ_n	$[0 \;\; x_2]$	$\left[\begin{smallmatrix}0&0\\0&1\end{smallmatrix}\right]$	$[0 \;\; v_{02}]$
$2n$	$n \to 0$	μ_n	$[x_1 \;\; x_1]$	$\left[\begin{smallmatrix}1&1\\0&0\end{smallmatrix}\right]$	$[v_{n1} \;\; v_{n1}]$

TABLE I
SHS TRANSITIONS FOR TRACKING THE OVERALL AGE IN THE MARKOV CHAIN OF FIG. 2.

l	$q_l \to q'_l$	$\lambda^{(l)}$	$\mathbf{x}\mathbf{A}_l$	$\mathbf{A}_l$	$\mathbf{v}_{q_l}\mathbf{A}_l$
$2i-1$	$0 \to i$	λ_i	$[x_1 \;\; x_2]$	$\left[\begin{smallmatrix}1&0\\0&1\end{smallmatrix}\right]$	$[v_{01} \;\; v_{02}]$
$2i$	$i \to 0$	μ_1	$[x_1 \;\; x_2]$	$\left[\begin{smallmatrix}1&0\\0&1\end{smallmatrix}\right]$	$[v_{i1} \;\; v_{i2}]$
- - -	- - -	- - -	- - -	- - -	- - -
$2k-1$	$0 \to k$	λ_k	$[0 \;\; x_2]$	$\left[\begin{smallmatrix}0&0\\0&1\end{smallmatrix}\right]$	$[0 \;\; v_{02}]$
$2k$	$k \to 0$	μ_k	$[x_1 \;\; x_1]$	$\left[\begin{smallmatrix}1&1\\0&0\end{smallmatrix}\right]$	$[v_{k1} \;\; v_{k1}]$

TABLE II
SHS TRANSITIONS FOR TRACKING THE CLASS k AGE IN THE MARKOV CHAIN OF FIG. 2. NOTE THAT $i \in B_{-k}$ REFERS TO ANY CLASS $i \neq k$.

We see in (14) that job class n with the smallest service rate μ_n can dominate the average age. Even though there is no queueing induced by the "slow truck" effect, an occasional very long service time can have an outsize effect on the average age. For comparison, we now analyze the class-specific AoI.

B. Class-specific Age

Under class-specific age tracking, the continuous state is used to track only class k jobs. Thus $x_1(t)$ is now the age of an observer that sees fresh class k jobs that enter service and $x_2(t)$ is the age at the monitor that sees processed class k jobs that complete service. The discrete Markov chain is unchanged; in state 0, the server is idle and in state $i > 0$ a class i job is in service. The non-preemptive blocking queue service model is unchanged.

The transition reset maps $\mathbf{A}_l$ have changed because the age process we track has changed. The new transition reset maps $\mathbf{A}_l$ are now shown in Table II. These transitions are shown in Table II. Using $B_{-k} = \{1, \dots, n\} \backslash \{k\}$ to denote the set of states in which the server is busy with a job not in class k, we see that continuous state $\mathbf{x}$ does not change for transitions into or out of states $\bar{q} \in B_{-k}$ because these state transitions involve updates other than those in the targeted class k. However, for class k arrivals, the transitions into state k reset x_1 to $x'_1 = 0$ since it marks a fresh class k arrival. Similarly a departure from state k marks a delivery of a class k update to the monitor and thus x_2 is reset to $x'_2 = x_1$, the age of the just delivered update. Thus,

$$\mathbf{A}_l = \begin{cases} \mathbf{A} & = \left[\begin{smallmatrix}0&0\\0&1\end{smallmatrix}\right], \quad l = 2k-1, \\ \mathbf{D} & = \left[\begin{smallmatrix}1&1\\0&0\end{smallmatrix}\right], \quad l = 2k, \\ \mathbf{I} & = \left[\begin{smallmatrix}1&0\\0&1\end{smallmatrix}\right], \quad l = 2i-1, i \in B_{-k}, \\ \mathbf{I} & = \left[\begin{smallmatrix}1&0\\0&1\end{smallmatrix}\right], \quad l = 2i, i \in B_{-k}. \end{cases} \tag{15}$$

To employ Theorem 1, we first observe that the state probabilities $\bar{\pi}_i$ are unchanged and given by (8). We note that appying (5a) to (15) at $\bar{q} = 0$ yields

$$\bar{\mathbf{v}}_0 \lambda = \bar{\pi}_0 \mathbf{1} + \mu_k \bar{\mathbf{v}}_k \mathbf{D} + \sum_{i \in B_{-k}} \mu_i \bar{\mathbf{v}}_i \mathbf{I}. \tag{16}$$

At $\bar{q} = k$ and $\bar{q} = i \in B_{-k}$, (5a) and (15) imply

$$\bar{\mathbf{v}}_k \mu_k = \bar{\pi}_k \mathbf{1} + \lambda_k \bar{\mathbf{v}}_0 \mathbf{A}, \tag{17a}$$

$$\bar{\mathbf{v}}_i \mu_i = \bar{\pi}_i \mathbf{1} + \lambda_i \bar{\mathbf{v}}_0 \mathbf{I}, \qquad i \in B_{-k}. \tag{17b}$$

Since $\mathbf{A}\mathbf{D} = \mathbf{0}$, it follows from applying (17) to (16) that

$$\bar{\mathbf{v}}_0 \lambda = \bar{\pi}_0 \mathbf{1} + \bar{\pi}_k \mathbf{1} \mathbf{D} + \sum_{i \in B_{-k}} (\bar{\pi}_i \mathbf{1} + \lambda_i \bar{\mathbf{v}}_0). \tag{18}$$

Since $\bar{\pi}_k \mathbf{1} \mathbf{D} = \bar{\pi}_k \mathbf{1}$, it follows from (8) that

$$\bar{\mathbf{v}}_0 = \frac{\mathbf{1}}{\lambda - \sum_{i \in B_{-k}} \lambda_i} = \frac{\mathbf{1}}{\lambda_k}. \tag{19}$$

Since $\mathbf{1}\mathbf{A} = [0 \;\; 1]$, applying (8) and (19) to (17) yields

$$\bar{\mathbf{v}}_k = \frac{1}{\mu_k} \left[\frac{\rho_k \mathbf{1}}{1+\rho} + [0 \;\; 1] \right] \tag{20a}$$

$$\bar{\mathbf{v}}_i = \frac{1}{\mu_i} \frac{\rho_i \mathbf{1}}{1+\rho} + \frac{\rho_i}{\lambda_k} \mathbf{1}. \tag{20b}$$

The average age of class k updates at the monitor is

$$\Delta_k = \mathrm{E}[x_2] = \sum_{i=0}^{c} \bar{\mathbf{v}}_{i2} \;\; = \frac{1+\rho}{\lambda_k} + \sum_{i=1}^{c} \frac{\rho_i}{1+\rho} \frac{1}{\mu_i}. \tag{21}$$

Just as we saw for the overall average age (14), see in (21) that job class n with the smallest service rate μ_n can still dominate the average age. We also note that if $p_1 = \cdots = p_n$, then the average age Δ_k for each class k will be the same. This may be surprising inasmuch as the classes can have very different service times.

However, what is perhaps most striking is the similarity of the overall age Δ in (14) and the class specific age Δ_k in (21). In both cases, there are two terms; the second term is a common age penalty function while the first depends only on the update arrival rate.

IV. OPTIMIZING THE UPDATE ARRIVAL RATE

We now examine how the the arrival rate λ affects the both the overall and class-specific ages. We start by observing that

$$\delta_1 = \sum_{i=1}^{c} \frac{p_i}{\mu_i}, \qquad \delta_2 = 2 \sum_{i=1}^{c} \frac{p_i}{\mu_i^2} \tag{22}$$

are the first and second moments of the hyperexponential service time. With this notation, the overall age in (14) is

$$\Delta = \delta_1 \left[\frac{1 + \lambda\delta_1}{\lambda\delta_1} + \frac{1}{2} \frac{\lambda\delta_1}{1 + \lambda\delta_1} \frac{\delta_2}{\delta_1^2} \right]. \tag{23}$$

Defining $x = \lambda\delta_1/(1+\lambda\delta_1)$, we observe that all $x \in [0, 1)$ are feasible by selection of $\lambda \geq 0$. Setting $\mathrm{d}\Delta/\mathrm{d}x = 0$ at $x = x^*$, we have

$$x^* = \sqrt{2\delta_1^2/\delta_2}. \tag{24}$$

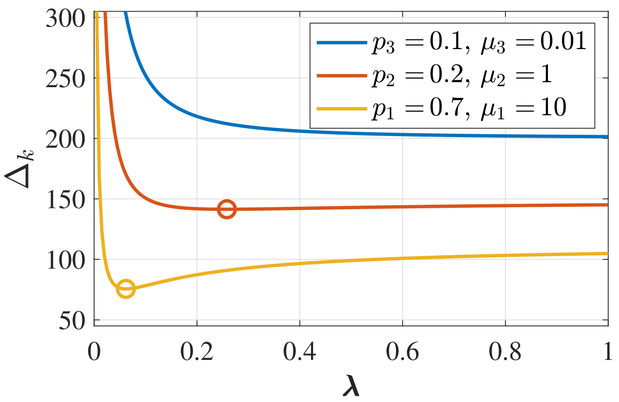

Fig. 3. Class-specific average ages Δ_k as a function of the arrival rate λ. The optimal λ for a specific class k is marked with $\circ$ if it exists.

If $\delta_2 > 2\delta_1^2$, then $x^* < 1$ and the age-minimizing arrival rate λ^* exists. In short, there is an age-minimizing arrival rate when the coefficient of variation of the service time is sufficiently large. On the other hand, if $\delta_2 \leq 2\delta_1^2$, then the overall age is a decreasing function λ and is minimized as $\lambda \to \infty$. In this limit, the system effectively operates as a just-in-time system in which a fresh update goes into service immediately after a service completion.

We note that this result has been previously reported in [7] in the general M/G/1/1 blocking queue.[1] Nevertheless, we repeat this earlier result in order to contrast it to the class-specific scenario. The class-specific age Δ_k in (21) can be written as

$$\Delta_k = \delta_1 \left[\frac{1 + \lambda\delta_1}{\lambda\delta_1} \frac{1}{p_k} + \frac{1}{2} \frac{\lambda\delta_1}{1 + \lambda\delta_1} \frac{\delta_2}{\delta_1^2} \right]. \tag{25}$$

In this case, setting $\mathrm{d}\Delta_k/\mathrm{d}x = 0$ at $x = x_k^*$ yields

$$x_k^* = \sqrt{\frac{2\delta_1^2}{\delta_2 p_k}} = \frac{x^*}{\sqrt{p_k}}. \tag{26}$$

For class k updates, there is an age-optimal finite arrival rate if $p_k\delta_2 > 2\delta_1^2$ and this optimal rate increases as the probability p_k decreases. However, this rate will be suboptimal for other classes. Moreover, an update class k with $p_k \leq 2\delta_1^2/\delta_2$ will prefer λ to be as large as possible.

An example of these tradeoffs appears in Fig. 3. The age-optimal arrival rate λ is finite for classes 1 and 2. For class 1 with the highest probability $p_1 = 0.7$, the optimal arrival rate is $\lambda^* \approx 0.07$. This is suboptimal for class 3 with the smallest probability $p_3 = 0.1$ as the age Δ_3 is a decreasing function of λ.

V. Conclusion

This work reports on AoI analysis of an update processing system in which updates belonging to multiple service classes arrive as a rate λ Poisson process and new jobs are discarded when the server is busy. With a hyperexponential service time model to describe these classes, we were able to use SHS analysis to describe the overall average age and the class-specific average age.

We observed there is a tension in the system in that low probability service classes desire the overall arrival rate to be as high as possible while higher probability classes benefit from class-specific tuning of the update rate. While this is a somewhat ambiguous conclusion, it highlights the need for experimental characterization of updating applications. For example, if an application enables arriving updates to be tagged by service class or to be identified by initial pre-processing, the service facility could implement class-specific policies for admission and preemption. However, the efficacy of these policies will depend on properties of that application.

[1] The optimality of finite λ is analagous to results in [16], [17] which found that waiting before submitting an update to a non-preemptive system could outperform the just-in-time policy, particularly when short service times have high probability.

References

[1] Liang-Chi Chiu, Tian-Sheuan Chang, Jiun-Yen Chen, Nelson Yen-Chung Chang, et al. Fast SIFT design for real-time visual feature extraction. *IEEE Transactions on Image Processing*, 22(8):3158–3167, 2013.

[2] Qi Zhang, Yurong Chen, Yimin Zhang, and Yinlong Xu. SIFT implementation and optimization for multi-core systems. In *IEEE International Symposium on Parallel and Distributed Processing*, pages 1–8. IEEE, 2008.

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

[4] Anja Feldmann and Ward Whitt. Fitting mixtures of exponentials to long-tail distributions to analyze network performance models. *Performance Evaluation*, 31(3):245 – 279, 1998.

[5] A. M. Bedewy, Y. Sun, and N. B. Shroff. The age of information in multihop networks. *CoRR*, abs/1712.10061, 2017.

[6] Yoshiaki Inoue, Hiroyuki Masuyama, Tetsuya Takine, and Toshiyuki Tanaka. A general formula for the stationary distribution of the age of information and its application to single-server queues. *CoRR*, abs/1804.06139, 2018.

[7] E. Najm, R.D. Yates, and E. Soljanin. Status updates through M/G/1/1 queues with HARQ. In *Proc. IEEE Int'l. Symp. Info. Theory (ISIT)*, pages 131–135, June 2017.

[8] L. Huang and E. Modiano. Optimizing age-of-information in a multi-class queueing system. In *Proc. IEEE Int'l. Symp. Info. Theory (ISIT)*, June 2015.

[9] E. Najm and E. Telatar. Status updates in a multi-stream M/G/1/1 preemptive queue. In *IEEE Conference on Computer Communications (INFOCOM) Workshops*, pages 124–129, April 2018.

[10] J.P. Hespanha. Modelling and analysis of stochastic hybrid systems. *IEE Proceedings-Control Theory and Applications*, 153(5):520–535, 2006.

[11] R. D. Yates and S. K. Kaul. The age of information: Real-time status updating by multiple sources. *IEEE Transactions on Information Theory*, 65(3):1807–1827, 2018.

[12] D. Vermes. Optimal dynamic control of a useful class of randomly jumping processes. Technical Report PP-80-015, International Institute for Applied Systems Analysis, 1980.

[13] M. H. A. Davis. Piecewise-deterministic Markov processes: a general class of nondiffusion stochastic models. *J. Roy. Statist. Soc.*, 46:353–388, 1984.

[14] Lee DeVille, Sairaj Dhople, Alejandro D Domínguez-García, and Jiangmeng Zhang. Moment closure and finite-time blowup for piecewise deterministic Markov processes. *SIAM Journal on Applied Dynamical Systems*, 15(1):526–556, 2016.

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

[16] R.D. Yates. Lazy is timely: Status updates by an energy harvesting source. In *Proc. IEEE Int'l. Symp. Info. Theory (ISIT)*, pages 3008–3012, June 2015.

[17] 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 Trans. Info. Theory*, 63(11):7492–7508, November 2017.