

# Age and Value of Information Optimization for Systems with Multi-Class Updates

Ahmed Arafa<sup>1</sup> and Roy D. Yates<sup>2</sup>

<sup>1</sup>Department of Electrical and Computer Engineering, University of North Carolina at Charlotte

<sup>2</sup>Department of Electrical and Computer Engineering, Rutgers University

**Abstract**—Received samples of a stochastic process are processed by a server for delivery as updates to a monitor. Each sample belongs to a class that specifies a distribution for its processing time and a function that describes how the value of the processed update decays with age at the monitor. The class of a sample is identified when the processed update is delivered. The server implements a form of M/G/1/1 blocking queue; samples arriving at a busy server are discarded and samples arriving at an idle server are subject to an admission policy that depends on the age and class of the prior delivered update. For the delivered updates, we characterize the average age of information (AoI) and average value of information (VoI). We derive the optimal stationary policy that minimizes the convex combination of the AoI and (negative) VoI. It is shown that the policy has a threshold structure, in which a new sample is allowed to arrive to the server only if the previous update's age and value difference surpasses a certain threshold that depends on the specifics of the value function and system statistics.

## I. INTRODUCTION

Consider a system in which time-stamped raw data samples are processed into updates for a monitor. The server and the monitor are co-located, and each sample incurs a *processing time* before being delivered as an update to the monitor. Some updates can be processed quickly, while others require longer service times. Updates with long processing times may be unusually important or valuable at the monitor. Occasionally, an update may have a very long service time or be of exceptional importance.

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 delivered to the monitor. In the AR system, time-stamped images are samples that arrive at the input of 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.

For a second example, consider a camera system monitoring an urban crosswalk. The video frame updates require processing to generate bounding boxes that correspond to pedestrians in the crosswalk. As the number of pedestrians increases, the video frames require more processing time to identify bounding boxes. However, in the context of pedestrian safety, images with more pedestrians are more important and have greater value.

This work was supported by the U.S. National Science Foundation under Grants CNS 21-14537 and ECCS 21-46099.

Object recognition is typically a key step in these image processing 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]. However, when there is 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. The value of an image is likely to be increasing with its number of classified objects.

To model such situations, samples are categorized into  $M$  different classes. A sample belongs to class  $i$  with probability  $p_i$ . Each class  $i$  sample has a class-dependent processing time and carries a *value*  $v_i(t)$  at time  $t$ . Such value is only revealed at the monitor after being processed by the server.

Both timeliness and value of the updates are important. We evaluate the performance of the updating system using both age of information (AoI) [2] and value of information (VoI) [3] metrics. In particular, we develop a class of server policies that minimize a convex combination of AoI and (negative) VoI. This enables us to compare the characteristics of AoI and VoI minimization as well as tradeoffs between AoI and VoI.

In prior work on updates with non-memoryless service times [4]–[6], 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, if updates with long processing times are the important high-value updates, preemption will result in low VoI at the monitor. Moreover, VoI alters how we should process updates. Specifically, the system is earning a reward over time for update  $k - 1$  following its delivery. In that same time period, the system is either processing update  $k$  or waiting to begin the processing of update  $k$ . The key question is

*how long should the system accrue value from update  $k - 1$  before initiating the process to deliver update  $k$ ?*

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 [7] 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 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 [8], with all streams having the same general service time, it is shown that increasing the arrival rate for one stream/class can reduce its AoI, but at the expense of increased AoI for other classes.

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  $\lambda$  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.

### B. Notation

We define  $[x]^+ = \max(x, 0)$ . A random variable  $R$  has probability density function  $f_R(r)$ , expected value  $\mathbb{E}[R]$  and moment generating function (MGF)  $\Phi_R(s) = \mathbb{E}[e^{sR}]$ .

## II. SYSTEM MODEL

At the processor, a received sample belongs to class  $i$  with probability  $p_i$ . A class  $i$  sample consumes time  $\hat{Y}_i$  in processing. Let  $Y_k$  denote the processing time of the  $k$ th sample. For instance, if the  $\hat{Y}_i$ 's are exponential ( $\lambda_i$ ), then  $Y_k$  would be hyperexponential. We assume that processing times and class identities are independent and identically distributed (i.i.d.) across samples.

Let  $S_k$  and  $D_k$  denote the sampling and delivery time of the  $k$ th sample/update, respectively, and let  $i_k$  indicate its class, i.e.,  $i_k = i$  if sample  $k$  belongs to class  $i$ . Hence,

$$D_k = S_k + Y_k. \quad (1)$$

At the monitor, two quantities are observed over time: 1) the age-of-information (AoI),

$$\Delta(t) \triangleq t - S_k, \quad D_k \leq t < D_{k+1}; \quad (2)$$

and 2) the value-of-information (VoI),

$$V(t) = \nu_{i_k}(t - S_k), \quad D_k \leq t < D_{k+1}. \quad (3)$$

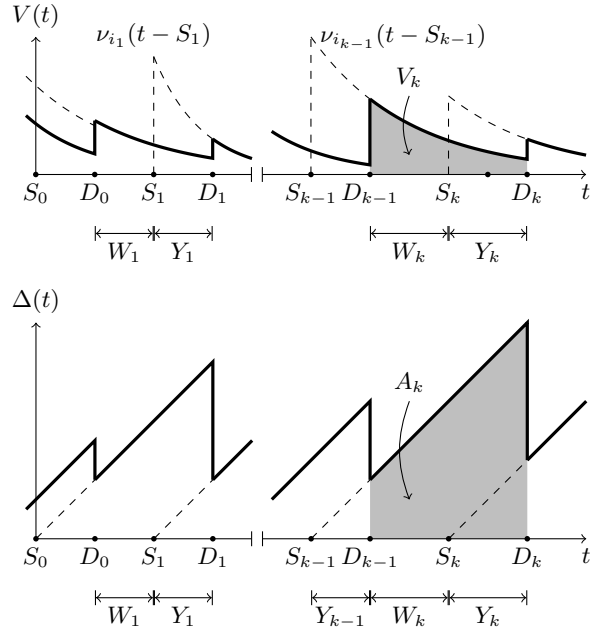


Fig. 1. Sample paths of age  $\Delta(t)$  and exponentially decaying value  $V(t)$ .

That is,  $\nu_{i_k}(\tau)$  denotes the value carried by the  $k$ th update when that update has age  $\tau$ . Note that  $\nu_{i_k}(\tau) = \nu_i(\tau)$  in case update  $k$  belongs to class  $i$ .

While our formulation allows for general forms of value functions  $\nu_i(\tau)$ , we will focus on *exponentially-decaying* value functions given by

$$\nu_i(\tau) = \nu_i e^{-\alpha_i \tau}, \quad \tau \geq 0, \quad (4)$$

for some  $\alpha_i \geq 0$ . Thus the class  $i$  of an update specifies its initial value  $\nu_i$  (which we will often refer to as the value of a class  $i$  update) and its value decay rate  $\alpha_i$ . When  $\alpha_i = 0$ , we obtain the special case of *fixed* value functions.

An example sample path of AoI and VoI versus time is shown in Figure 1.

Following the processing and delivery of update  $k - 1$ , let  $W_k$  denote the waiting time before the server starts processing the  $k$ th sample. We consider an *external arrivals* model, in which samples arrive as an exogenous rate  $\lambda$  Poisson process. In this model, our choice is to either block/discard an arriving sample or to admit and process that sample. Consequently, the waiting time  $W_k$  between the delivery of update  $k - 1$  and the start of processing of sample  $k$  is

$$W_k = W'_k + X_k, \quad (5)$$

where  $W'_k$  is a controlled waiting time during which arriving samples are blocked and  $X_k$  is the uncontrolled random exponential ( $\lambda$ ) time until a new exogenous sample arrives following the end of the controlled waiting/blocking period. After a sample gets admitted, any arriving sample while the server is busy processing will be blocked/discarded.

In the limit as  $\lambda \rightarrow \infty$ , the uncontrolled wait  $X_k \rightarrow 0$  and the external arrivals model becomes equivalent to a *generate-at-will* model in which a fresh arrival is admitted following

the controlled waiting time  $W_k = W'_k$ . In this limiting case, the sampling times  $\{S_k\}$  are fully controlled.

Our goal is to design sampling policies, i.e., the controlled waiting times  $\{W'_k\}$ , to optimize a weighted sum of the long-term average AoI and VoI. The long-term average AoI is defined as

$$\overline{\text{AoI}} \triangleq \limsup_{N \rightarrow \infty} \frac{\sum_{k=1}^N \mathbb{E} \left[ \int_{D_{k-1}}^{D_{k-1} + W_k + Y_k} \Delta(t) dt \right]}{\sum_{k=1}^N \mathbb{E} [D_k - D_{k-1}]}, \quad (6)$$

which, from Figure 1, can be more explicitly expressed as

$$\overline{\text{AoI}} = \limsup_{N \rightarrow \infty} \frac{\sum_{k=1}^N \mathbb{E} \left[ Y_{k-1} (W_k + Y_k) + \frac{1}{2} (W_k + Y_k)^2 \right]}{\sum_{k=1}^N \mathbb{E} [D_k - D_{k-1}]}. \quad (7)$$

The long-term average VoI on the other hand is defined as

$$\overline{\text{VoI}} \triangleq \limsup_{N \rightarrow \infty} \frac{\sum_{k=1}^N \mathbb{E} \left[ \int_{D_{k-1}}^{D_{k-1} + W_k + Y_k} V(t) dt \right]}{\sum_{k=1}^N \mathbb{E} [D_k - D_{k-1}]}. \quad (8)$$

Next, we discuss the optimization problem in detail.

### III. PROBLEM FORMULATION AND OBJECTIVE

Our main goal is to choose the waiting times to minimize a convex combination of the long-term average AoI and long-term average negative VoI:

$$\min_{\{W'_k \geq 0\}} (1 - \beta) \overline{\text{AoI}} - \beta \overline{\text{VoI}}, \quad (9)$$

for a weighting factor  $\beta$  satisfying  $0 \leq \beta \leq 1$ .

To get a handle on the above problem, let us denote by *epoch*  $k$  the time elapsed in between the delivery of update  $k-1$  and the delivery of update  $k$ . During epoch  $k$ , value is being accumulated from update  $k-1$ , the most recently delivered update. The key question that we posed in the Introduction section now corresponds to how long should one accrue value from update  $k-1$ , given that it has class  $i_{k-1} = i$  with value  $\nu_i$  and value decay rate  $\alpha_i$ , before initiating the process to deliver update  $k$ .

Since processing times and class identities are i.i.d., we focus on *stationary deterministic* waiting policies, see, e.g., [9]. Specifically, when the most recent update has class  $i_{k-1} = i$  and required processing time  $Y_{k-1}$  the controlled waiting time in epoch  $k$  is given by

$$W'_k \triangleq w_i(Y_{k-1}), \quad (10a)$$

where  $w_i(\cdot)$  is a class-dependent waiting function to be optimized. That is, *the controlled waiting time in an epoch is a deterministic function of the previous epoch's processing time and sample value*. It follows from (5) that the waiting time in epoch  $k$  is

$$W_k \triangleq w_i(Y_{k-1}) + X_k. \quad (10b)$$

Observe that such waiting policies induce a stationary distribution across epochs.

Substituting the waiting policy structure in (10) back in (7) and (8), the optimization problem in (9) can be reduced to a functional optimization problem over a single epoch.

Specifically, over epoch  $k$  of duration  $T_k$ , it can be seen from Figure 1 that the accumulated age and value are

$$A_k = Y_{k-1} (W_k + Y_k) + \frac{1}{2} (W_k + Y_k)^2, \quad (11)$$

$$V_k = \int_{D_{k-1}}^{D_{k-1} + W_k + Y_k} \nu_{i_{k-1}}(t - S_{k-1}) dt. \quad (12)$$

Since the expected duration of epoch  $k$  is

$$\mathbb{E}[T_k] = \mathbb{E}[W_k + Y_k], \quad (13)$$

the average AoI and VoI become

$$\overline{\text{AoI}} = \frac{\mathbb{E}[A_k]}{\mathbb{E}[T_k]}, \quad \overline{\text{VoI}} = \frac{\mathbb{E}[V_k]}{\mathbb{E}[T_k]}. \quad (14)$$

With the shorthand definition  $\bar{\beta} \triangleq 1 - \beta$ , problem (9) now becomes

$$\min_{\{w_i(\cdot)\}} \frac{\bar{\beta} \mathbb{E}[A_k] - \beta \mathbb{E}[V_k]}{\mathbb{E}[T_k]}, \quad \text{s.t. } w_i(t) \geq 0, \forall t, i. \quad (15)$$

## IV. MAIN RESULT

In this section, we present our main result, namely the solution of problem (15). Our solution will employ the Dinkelbach method [10] which requires optimization over a free parameter  $\theta$ . Because the problem and solution are somewhat complicated, a summary of the optimal solution given in Theorem 1 is presented in Section IV-A and this is followed in Section IV-B by a discussion of its basic properties. We then present some examples of the solution in Section V. The derivation of Theorem 1 is deferred to Section VI.

### A. Solution Summary

Recalling that raw updates arrive at the processor as a rate  $\lambda$  Poisson process, the uncontrolled wait  $X_k$  is an exponential ( $\lambda$ ) random variable with moments  $\mathbb{E}[X] = 1/\lambda$ ,  $\mathbb{E}[X^2] = 2/\lambda^2$ , and MGF  $\Phi_X(s) = \lambda/(\lambda - s)$ . Since a class  $i$  sample has processing time denoted  $\hat{Y}_i$ , the (overall) processing time  $Y$  has moments

$$\mathbb{E}[Y] = \sum_i p_i \mathbb{E}[\hat{Y}_i], \quad (16a)$$

$$\mathbb{E}[Y^2] = \sum_i p_i \mathbb{E}[\hat{Y}_i^2]. \quad (16b)$$

It follows that the independent sum  $Z = X + Y$  has moments

$$\mathbb{E}[Z] = 1/\lambda + \mathbb{E}[Y], \quad (17a)$$

$$\mathbb{E}[Z^2] = 2/\lambda^2 + \frac{2}{\lambda} \mathbb{E}[Y] + \mathbb{E}[Y^2]. \quad (17b)$$

The Dinkelbach parameter  $\theta$  specifies a threshold

$$\tau(\theta) \triangleq \theta - \bar{\beta} \mathbb{E}[Z] \quad (18)$$

that is common to all classes.

The value delivered by a class  $i$  update depends on its decay rate  $\alpha_i$  through the *value decay factor*

$$\phi_i \triangleq \nu_i \Phi_X(-\alpha_i) \Phi_{\hat{Y}_i}(-\alpha_i) = \nu_i \frac{\lambda}{\lambda + \alpha_i} \Phi_{\hat{Y}_i}(-\alpha_i), \quad (19a)$$

which is used to define the class  $i$  threshold function

$$h_i(t) \triangleq \bar{\beta}t - \beta\phi_i e^{-\alpha_i t}, \quad (19b)$$

and this defines the class  $i$  minimum inter-update time

$$\bar{y}_i(\theta) \triangleq \begin{cases} h_i^{-1}(\tau(\theta)), & \tau(\theta) \geq -\beta\phi_i, \\ 0, & \text{otherwise.} \end{cases} \quad (19c)$$

Examples of these threshold functions and their inverses are shown in Figure 2.

When the *prior* processed update belongs to class  $i$  and has service time  $\hat{Y}_i$ , the controlled waiting time during which arriving raw updates are discarded at the processor is

$$\hat{W}_i = [\bar{y}_i(\theta) - \hat{Y}_i]^+. \quad (19d)$$

For fixed  $\theta$ , the expected accumulated age and value over one epoch are

$$\begin{aligned} \mathbb{E}[A_k] &= \sum_i p_i \left( (\bar{y}_i(\theta) + \mathbb{E}[Z]) \mathbb{E}[\hat{W}_i] - \mathbb{E}[\hat{W}_i^2]/2 \right) \\ &\quad + \mathbb{E}[Y] \mathbb{E}[Z] + \mathbb{E}[Z^2]/2, \end{aligned} \quad (19e)$$

$$\mathbb{E}[V_k] = \sum_i \frac{p_i}{\alpha_i} \left( \nu_i \Phi_{\hat{Y}_i}(-\alpha_i) - \phi_i \Phi_{\hat{Y}_i + \hat{W}_i}(-\alpha_i) \right), \quad (19f)$$

and the expected duration of an epoch is

$$\mathbb{E}[T_k] = \sum_i p_i \mathbb{E}[\hat{W}_i] + \mathbb{E}[Z]. \quad (19g)$$

The average AoI and VoI are then given by (14). With

$$p(\theta) \triangleq \bar{\beta} \mathbb{E}[A_k] - \beta \mathbb{E}[V_k] - \theta \mathbb{E}[T_k], \quad (19h)$$

the following theorem describes how to find the optimal  $\theta = \theta^*$  that equals the minimum weighted AoI/VoI objective in (9).

**Theorem 1** *The optimal waiting policy of problem (9) is given by the class-dependent threshold policy in (19d), with the class  $i$  threshold  $\bar{y}_i(\theta)$  in (19c). The value  $\theta = \theta^*$  given by the unique solution of  $p(\theta^*) = 0$  in (19h) is equal to the optimal solution of problem (9).*

In the derivation of the theorem in Section VI, we show that  $\theta^*$  can be found by a simple bisection search.

### B. Discussion

We observe that the solution of Theorem 1 generalizes the AoI-optimal waiting strategy found in [11] for generate-at-will systems. In that work, when the prior service time is  $y$ , the source waits for time  $w = [y^* - y]^+$  before generating the next update, for some threshold  $y^*$ . Just like  $y^*$  in [11],  $\bar{y}_i(\theta)$  can be interpreted as a minimum inter-update time.

Because the threshold functions  $h_i(t)$  are monotone increasing, the minimum inter-update time  $\bar{y}_i(\theta)$  increases with  $\theta$ . Hence, increasing  $\theta$  slows the processing rate of updates. Specifically, if the prior update was processed quickly and found to be in class  $i$ , the additional wait  $\hat{W}_i$  is inserted before permitting an arriving update to be processed. This additional wait depends on the value decay factor  $\phi_i$ ; *when the prior update is more valuable, the additional wait is larger.*

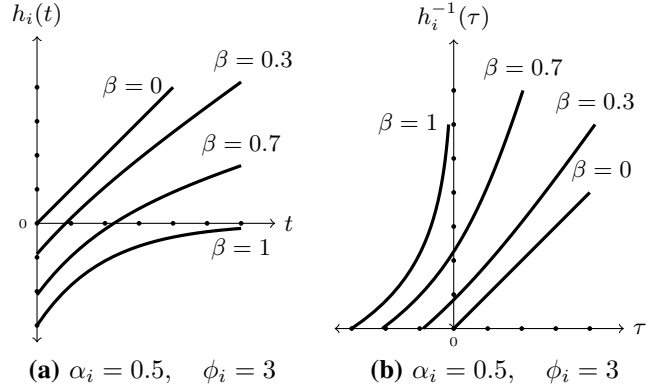


Fig. 2. Examples of (a) the threshold function  $h_i(t)$  and (b) the corresponding inverses  $h_i^{-1}(\tau)$ . The minimum  $h_i(0) = -\beta\phi_i$  occurs at  $t = 0$  and  $h_i(t) \rightarrow (1 - \beta)t$  as  $t \rightarrow \infty$ .

### V. EXAMPLES: HYPEREXPONENTIAL SERVICE

Here we consider a multi-class system such that type  $i$  samples occur with probability  $p_i$ , and have sample value  $\nu_i$  and decay rate  $\alpha_i$ . The service time is hyperexponential in that type  $i$  samples have exponential ( $\mu_i$ ) processing times  $\hat{Y}_i$ . Hence,

$$\mathbb{E}[\hat{Y}_i] = \frac{1}{\mu_i}, \quad \mathbb{E}[\hat{Y}_i^2] = \frac{2}{\mu_i^2}, \quad \Phi_{\hat{Y}_i}(-\alpha_i) = \frac{\mu_i}{\mu_i + \alpha_i}. \quad (20)$$

From (16), it follows that the service time  $Y$  has moments

$$\mathbb{E}[Y] = \sum_i \frac{p_i}{\mu_i}, \quad \mathbb{E}[Y^2] = \sum_i \frac{2p_i}{\mu_i^2}, \quad (21)$$

and  $Z$  has moments given by (17). With these values, (19a) yields the class  $i$  value decay factor

$$\phi_i = \nu_i \frac{\lambda}{\lambda + \alpha_i} \frac{\mu_i}{\mu_i + \alpha_i}, \quad (22a)$$

while  $h_i(t)$ ,  $\tau(\theta)$ ,  $\bar{y}_i(\theta)$ , and  $\hat{W}_i$  are specified by (19b)-(19d). Finding  $\mathbb{E}[A_k]$  in (19e) requires the moments  $\mathbb{E}[\hat{W}_i]$  and  $\mathbb{E}[\hat{W}_i^2]$ . With the shorthand notation  $\bar{y}_i \triangleq \bar{y}_i(\theta)$ , we use integration by parts to calculate

$$\mathbb{E}[\hat{W}_i] = \int_0^{\bar{y}_i} (\bar{y}_i - y) \mu_i e^{-\mu_i y} dy = \bar{y}_i - \frac{1}{\mu_i} + \frac{e^{-\mu_i \bar{y}_i}}{\mu_i}, \quad (23)$$

$$\mathbb{E}[\hat{W}_i^2] = \int_0^{\bar{y}_i} (\bar{y}_i - y)^2 \mu_i e^{-\mu_i y} dy = \bar{y}_i^2 - \frac{2 \mathbb{E}[\hat{W}_i]}{\mu_i}. \quad (24)$$

Applying (24) to (19e) yields

$$\begin{aligned} \mathbb{E}[A_k] &= \sum_i p_i \left( (\bar{y}_i + 1/\mu_i + \mathbb{E}[Z]) \mathbb{E}[\hat{W}_i] - \bar{y}_i^2/2 \right) \\ &\quad + \mathbb{E}[Y] \mathbb{E}[Z] + \mathbb{E}[Z^2]/2. \end{aligned} \quad (25)$$

Substituting  $\mathbb{E}[Y]$  from (21) and  $\mathbb{E}[\hat{W}_i]$  from (23) in (25) yields

$$\begin{aligned} \mathbb{E}[A_k] &= \sum_i p_i \left( \bar{y}_i^2/2 - \frac{1}{\mu_i^2} + \bar{y}_i \mathbb{E}[Z] \right. \\ &\quad \left. + (\bar{y}_i + \frac{1}{\mu_i} + \mathbb{E}[Z]) \frac{e^{-\mu_i \bar{y}_i}}{\mu_i} \right) + \frac{\mathbb{E}[Z^2]}{2}. \end{aligned} \quad (26)$$

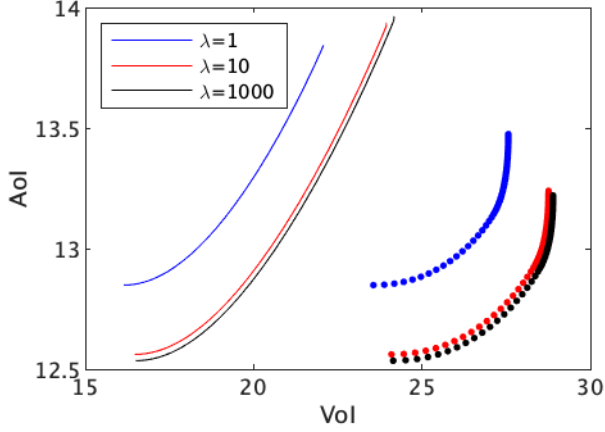


Fig. 3. AoI vs. VoI tradeoff for a two class system with hyperexponential service:  $[p_1, p_2] = [0.5, 0.5]$ ,  $[\nu_1, \nu_2] = [100, 1]$ ,  $[\alpha_1, \alpha_2] = [0.1, 1]$  and  $[\mu_1, \mu_2] = [0.1, 1]$  (solid lines) or  $[\mu_1, \mu_2] = [1, 0.1]$  (dotted lines).

From (17b) and (21), we obtain

$$\mathbb{E}[Z^2] = \frac{2}{\lambda^2} + 2 \sum_i p_i \left( \frac{1}{\lambda \mu_i} + \frac{1}{\mu_i^2} \right). \quad (27)$$

It follows that

$$\begin{aligned} \mathbb{E}[A_k] &= \sum_i p_i \left( \bar{y}_i^2 / 2 + \frac{1}{\lambda \mu_i} + \bar{y}_i \mathbb{E}[Z] \right. \\ &\quad \left. + \left( \bar{y}_i + \frac{1}{\mu_i} + \mathbb{E}[Z] \right) \frac{e^{-\mu_i \bar{y}_i}}{\mu_i} \right) + \frac{1}{\lambda^2}. \end{aligned} \quad (28)$$

To find  $\mathbb{E}[V_k]$ , we need  $\Phi_{\hat{Y}_i + \hat{W}_i}(-\alpha_i)$ . From the observation  $\hat{Y}_i + \hat{W}_i = \max(\bar{y}_i, \hat{Y}_i)$ , we can write

$$\begin{aligned} \Phi_{\hat{Y}_i + \hat{W}_i}(s) &= \mathbb{E} \left[ e^{s \max(\bar{y}_i, \hat{Y}_i)} \right] \\ &= e^{s \bar{y}_i} \mathbb{P} \left[ \hat{Y}_i \leq \bar{y}_i \right] + \int_{\bar{y}_i}^{\infty} e^{s y} \mu_i e^{-\mu_i y} dy \\ &= e^{s \bar{y}_i} (1 - e^{-\mu_i \bar{y}_i}) + \frac{\mu_i}{\mu_i - s} e^{-(\mu_i - s) \bar{y}_i}. \end{aligned} \quad (29)$$

Substituting (20) and (29) with  $s = -\alpha_i$  into (19f) yields

$$\mathbb{E}[V_k] = \sum_i \frac{p_i \nu_i \mu_i}{\alpha_i (\mu_i + \alpha_i)} \left[ 1 - \frac{\lambda e^{-\alpha_i \bar{y}_i}}{\lambda + \alpha_i} \left( 1 - \frac{\alpha_i e^{-\mu_i \bar{y}_i}}{\mu_i + \alpha_i} \right) \right]. \quad (30)$$

Furthermore, we see from (19g) and (23) that

$$\mathbb{E}[T_k] = \sum_i p_i \left[ \bar{y}_i + \frac{e^{-\mu_i \bar{y}_i}}{\mu_i} \right] + \frac{1}{\lambda}. \quad (31)$$

The selection of  $\theta$  specifies the threshold  $\tau(\theta)$  and the minimum wait  $\bar{y}_i = \bar{y}_i(\theta)$ . The AoI and VoI and the weighted combination  $p(\theta)$  in (19h) can then be calculated from  $\mathbb{E}[A_k]$  in (25),  $\mathbb{E}[V_k]$  in (30), and  $\mathbb{E}[T_k]$  in (31). Thus, for each value of  $\beta \in [0, 1)$ , we search for  $\theta^*$  satisfying  $p(\theta^*) = 0$  in order to find the minimum combined AoI and VoI in (14). That is, at  $\theta = \theta^*$ , we obtain an optimal AoI and VoI pair. By varying  $\beta$  and finding the  $\theta^*$  for each  $\beta$ , we obtain the boundary of feasible (AoI, VoI) pairs.

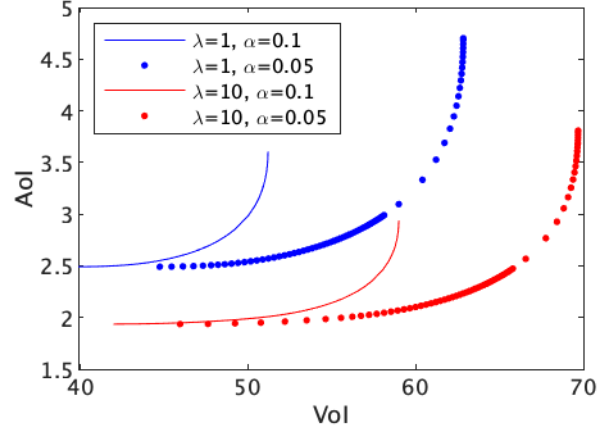


Fig. 4. AoI vs. VoI tradeoff for a two class system with hyperexponential service:  $[p_1, p_2] = [0.5, 0.5]$ ,  $[\nu_1, \nu_2] = [100, 1]$ ,  $[\mu_1, \mu_2] = [1, 1]$  and  $[\alpha_1, \alpha_2] = [0.1, 0.1]$  (solid lines) or  $[\alpha_1, \alpha_2] = [0.05, 0.05]$  (dotted lines).

In the example of Figure 3, class 1 and class 2 samples arrive equiprobably but with  $[\nu_1, \nu_2] = [100, 1]$ ; class 1 samples are  $100\times$  more valuable. Moreover, with  $[\alpha_1, \alpha_2] = [0.1, 1]$ , the class 1 value decays  $10\times$  more slowly. The figure depicts two collections (solid/dotted lines) of AoI/VoI tradeoff curves. In the solid line case,  $[\mu_1, \mu_2] = [0.1, 1]$ , corresponding to the valuable updates requiring  $10\times$  greater processing time. In the dotted line case,  $[\mu_1, \mu_2] = [1, 0.1]$ , so the valuable samples are now the updates that are processed quickly. In both cases, we see that we can choose whether to minimize age or to maximize value by varying the weight  $\beta$ . We also see that increasing the sample arrival rate  $\lambda$  improves the tradeoff by enabling more precise control of when to start processing the next sample. In both cases,  $\lambda = 10$  and  $\lambda = 1000$  are almost the same and  $\lambda = 1000$  is essentially indistinguishable from the generate-at-will system performance. We note, however, that the dotted line curves offer better AoI/VoI tradeoffs simply because processing the valuable updates fast results in less value being lost in processing.

A second example in Figure 4 examines the effect of decay parameters  $\alpha_i$ . Specifically, the two classes have the same decay rate  $\alpha_1 = \alpha_2 = \alpha$  and the figure compares  $\alpha = 0.1$  against  $\alpha = 0.05$ . As in the previous example, class 1 and 2 samples are equiprobable and class 1 samples have  $100\times$  higher value. As we would expect, reducing the decay rate  $\alpha$  increases the VoI at the monitor and improves the AoI/VoI tradeoff. Similarly, increasing the sample arrival rate  $\lambda$  improves the AoI-VoI tradeoff.

## VI. DERIVATION OF THEOREM 1

Over the interval  $[D_{k-1}, D_k]$ , value of information is accrued from update  $k-1$ , which was sampled at time  $S_{k-1}$  and delivered to the monitor at time

$$D_{k-1} = S_{k-1} + Y_{k-1}. \quad (32)$$

With exogenous arrivals and decaying exponential value functions, it follows from (12) and the substitution  $\tau = t - D_{k-1}$

that the accumulated VoI over the interval is

$$\begin{aligned} V_k &= \int_0^{W_k+Y_k} \nu_{i_{k-1}} e^{-\alpha_{i_{k-1}}(\tau+Y_{k-1})} d\tau \\ &= \frac{\nu_{i_{k-1}}}{\alpha_{i_{k-1}}} e^{-\alpha_{i_{k-1}}Y_{k-1}} \left(1 - e^{-\alpha_{i_{k-1}}(W_k+Y_k)}\right). \end{aligned} \quad (33)$$

When update  $k-1$  has class  $i_{k-1} = i$ , we denote the update processing time as  $Y_{k-1} = \hat{Y}_i$  to highlight its known class dependence. Similarly, we denote the class-dependent controlled wait by  $\hat{W}_i = w_i(\hat{Y}_i)$ . With this notation, (10b) becomes

$$W_k = \hat{W}_i + X_k. \quad (34)$$

Averaging over the class  $i_{k-1}$  of update  $k-1$ , (33) yields the expected value

$$\mathbb{E}[V_k] = \sum_i \frac{p_i \nu_i}{\alpha_i} \mathbb{E} \left[ e^{-\alpha_i \hat{Y}_i} \left(1 - e^{-\alpha_i (\hat{W}_i + X_k + Y_k)}\right) \right]. \quad (35)$$

With the value decay factor  $\phi_i$  defined in (19a), it follows from mutual independence of  $(\hat{W}_i, \hat{Y}_i)$ ,  $X_k$  and  $Y_k$  that  $\mathbb{E}[V_k]$  is given by (19f).

Over the same epoch, (11), (34) and the definition  $Z_k \triangleq X_k + Y_k$  imply that the expected accumulated age is

$$\mathbb{E}[A_k] = \sum_i p_i \mathbb{E} \left[ \hat{Y}_i (\hat{W}_i + Z_k) + \frac{1}{2} (\hat{W}_i + Z_k)^2 \right]. \quad (36)$$

Since  $Z_k$  is independent of  $\hat{Y}_i$  and  $\hat{W}_i$ , we have

$$\begin{aligned} \mathbb{E}[A_k] &= \sum_i p_i \mathbb{E} \left[ \hat{Y}_i \hat{W}_i + \hat{W}_i^2/2 + \hat{W}_i \mathbb{E}[Z] \right] \\ &\quad + \mathbb{E}[Y] \mathbb{E}[Z] + \mathbb{E}[Z^2]/2. \end{aligned} \quad (37)$$

In addition, we observe from (13), (34), and the definition of  $Z_k$  that  $\mathbb{E}[T_k]$  is given by (19g).

We now follow Dinkelbach's approach [10] to transform problem (15) into the following auxiliary problem:

$$p(\theta) \triangleq \min_{\{w_i(\cdot)\}} \bar{\beta} \mathbb{E}[A_k] - \beta \mathbb{E}[V_k] - \theta \mathbb{E}[T_k] \quad (38a)$$

$$\text{s.t. } w_i(t) \geq 0, \quad \forall i, t, \quad (38b)$$

for some  $\theta \in \mathbb{R}$ . The solution of problem (15) is now given by the unique  $\theta^*$  that solves  $p(\theta^*) = 0$  [10], which can be found by, e.g., a bisection search.

To solve problem (38), we define the Lagrangian

$$\mathcal{L} = \bar{\beta} \mathbb{E}[A_k] - \beta \mathbb{E}[V_k] - \theta \mathbb{E}[T_k] - \sum_i \int w_i(y) \eta_i(y) dy, \quad (39)$$

where  $\eta_i(y)$  is a Lagrange multiplier. It follows from (37), (19f), and (19g) that

$$\begin{aligned} \mathcal{L} &= \sum_i p_i \int \hat{f}_i(y) \left[ \bar{\beta} y w_i(y) + \frac{\bar{\beta} w_i^2(y)}{2} + w_i(y) (\bar{\beta} \mathbb{E}[Z] - \theta) \right. \\ &\quad \left. + \beta \frac{\phi_i}{\alpha_i} e^{-\alpha_i [y + w_i(y)]} \right] dy \\ &\quad - \sum_i \int w_i(y) \eta_i(y) dy + \bar{\beta} \mathbb{E}[Y] \mathbb{E}[Z] + \frac{1}{2} \bar{\beta} \mathbb{E}[Z^2] \end{aligned}$$

$$- \beta \sum_i \frac{p_i}{\alpha_i} \nu_i \Phi_{\hat{Y}_i}(-\alpha_i) - \theta \mathbb{E}[Z], \quad (40)$$

where  $\hat{f}_i(y)$  denotes the density of  $\hat{Y}_i$ . Taking the functional derivative of  $\mathcal{L}$  with respect to  $w_j(y)$  and equating to 0 yields

$$\begin{aligned} \bar{\beta} [y + w_j(y)] - \beta \phi_j e^{-\alpha_j (y + w_j(y))} \\ = \frac{\eta_j(y)}{p_j \hat{f}_j(y)} + \theta - \bar{\beta} \mathbb{E}[Z]. \end{aligned} \quad (41)$$

From the definition of the threshold function  $h_j(t)$  in (19b) and the threshold  $\tau(\theta)$  in (18), the condition (41) becomes

$$h_j(y + w_j(y)) = \frac{\eta_j(y)}{p_j \hat{f}_j(y)} + \tau(\theta). \quad (42)$$

We observe that  $h_j(t)$  is a strictly increasing function and thus has an inverse  $h_j^{-1}(\cdot)$ . Moreover, for  $w \geq 0$ ,

$$h_j(y + w) \geq h_j(y) = \bar{\beta} y - \beta \phi_j e^{-\alpha_j y}. \quad (43)$$

Hence, if  $h_j(y) > \tau(\theta)$ , or equivalently

$$y > \bar{y}_j(\theta) \triangleq h_j^{-1}(\tau(\theta)), \quad (44)$$

then (42) is satisfied with  $w_j(y) = 0$  and  $\eta_j(y) > 0$ . Otherwise, (42) is satisfied with  $\eta_j(y) = 0$  and  $w_j(y)$  such that  $y + w_j(y) = \bar{y}_j(\theta)$ . Putting these facts together, the optimal controlled-wait function is  $w_j^*(y) = [\bar{y}_j(\theta) - y]^+$ . Returning to (37), we have shown that  $\hat{W}_i = [\bar{y}_i(\theta) - \hat{Y}_i]^+$  and this implies

$$\begin{aligned} \mathbb{E}[\hat{Y}_i \hat{W}_i] &= \mathbb{E} \left[ (\bar{y}_i(\theta) - [\bar{y}_i(\theta) - \hat{Y}_i]) [\bar{y}_i(\theta) - \hat{Y}_i]^+ \right] \\ &= \mathbb{E} [\bar{y}_i(\theta) \hat{W}_i - \hat{W}_i^2]. \end{aligned} \quad (45)$$

Substituting (45) in (37) yields (19e). This concludes the derivation of Theorem 1.

## REFERENCES

- [1] Q. Zhang, Y. Chen, Y. Zhang, and Y. Xu. SIFT implementation and optimization for multi-core systems. In *Proc. IEEE IPDPS*, April 2008.
- [2] S. K. Kaul, R. D. Yates, and M. Gruteser. Real-time status: How often should one update? In *Proc. IEEE Infocom*, March 2012.
- [3] A. Kosta, N. Pappas, A. Ephremides, and V. Angelakis. Age and value of information: Non-linear age case. In *Proc. IEEE ISIT*, June 2017.
- [4] A. M. Bedewy, Y. Sun, and N. B. Shroff. The age of information in multihop networks. *IEEE/ACM Trans. Netw.*, 27(3):1248–1257, June 2019.
- [5] 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. *IEEE Trans. Inf. Theory*, 65(12):8305–8324, December 2019.
- [6] E. Najm, R. D. Yates, and E. Soljanin. Status updates through M/G/1/1 queues with HARQ. In *Proc. IEEE ISIT*, June 2017.
- [7] L. Huang and E. Modiano. Optimizing age-of-information in a multi-class queueing system. In *Proc. IEEE ISIT*, June 2015.
- [8] E. Najm and E. Telatar. Status updates in a multi-stream M/G/1/1 preemptive queue. In *Proc. IEEE Infocom*, April 2018.
- [9] A. Arafa, K. Banawan, K. G. Seddik, and H. V. Poor. Sample, quantize, and encode: Timely estimation over noisy channels. *IEEE Trans. Commun.*, 69(10):6485–6499, October 2021.
- [10] W. Dinkelbach. On nonlinear fractional programming. *Management Science*, 13(7):492–498, 1967.
- [11] Y. Sun, E. Uysal-Biyikoglu, R. D. Yates, C. E. Koksall, and N. B. Shroff. Update or wait: How to keep your data fresh. *IEEE Trans. Inf. Theory*, 63(11):7492–7508, November 2017.