Optimizing Age-of-Information in a Multi-class Queueing System

Longbo Huang[†] and Eytan Modiano[*]
[†]longbohuang@tsinghua.edu.cn, IIIS@Tsinghua University
[*]modiano@mit.edu, MIT

***Abstract*—We consider the age-of-information in a multi-class $M/G/1$ queueing system, where each class generates packets containing status information. Age of information is a relatively new metric that measures the amount of time that elapsed between status updates, thus accounting for both the queueing delay and the delay between packet generation. This gives rise to a tradeoff between frequency of status updates, and queueing delay. In this paper, we study this tradeoff in a system with heterogenous users modeled as a multi-class $M/G/1$ queue. To this end, we derive the exact peak age-of-Information (PAoI) profile of the system, which measures the "freshness" of the status information. We then seek to optimize the age of information, by formulating the problem using quasiconvex optimization, and obtain structural properties of the optimal solution.**

I. Introduction

Realtime status information is critical for optimal control in many networked systems, such as sensor networks used to monitor temperature or other physical phenomenon [1]; autonomous vehicle systems, where accurate position information is required to avoid collisions [2]; or wireless networks where realtime channel state information is needed to make scheduling decisions [3]. In all of these systems, what matters is not how fast the update information gets delivered, but rather, how accurately the received information describes the physical phenomenon being observed.

Recently, [4] proposes the notion of *age-of-information* (AoI), which measures the average time between the generation of an update message until it is received by the control unit; thus measuring the "freshness" of the available information. The early works on age-of-information consider homogeneous systems, where all entities use the same amount of resources for status update, and the length of status messages can be modeled using an i.i.d. exponential distribution.

In this paper, we consider a heterogenous systems where entities generate status messages with different length (service time) distributions. In particular, we consider a multi-class M/G/1 queueing systems; where each entity generates status update messages according to a given distribution, and derive the exact *peak age-of-information* (PAoI) value for each entity, which is the average maximum elapsed time since the latest received update packet is generated, and captures the extent to which update information is delayed. We then consider a system where, for packet management, at most one packet can be kept in the system, and compute the PAoI for the multi-class $M/G/1/1$ queue.

Next, we turn our attention to the problem of optimizing the PAoI by controlling the arrival rate of update messages (i.e., the sampling rate of the physical process being observed). We formulate the optimization problem as a minimization of a quasiconvex cost function of the system age-of-information profile. We show that in the $M/G/1/1$ case, the optimization problem is a quasi-convex program and derive properties of the optimal solution. In the general $M/G/1$ case, however, the problem is a general non-convex program for which we derive an approximate solution.

The notion of age-of-information was first introduced in [4] in the context of a single source modeled as an M/M/1 queue, and extended to multiple sources in [5]. In [6], [7] the authors consider the problem of minimizing the age of system state in vehicular networks. In [8] the AoI is analyzed for a system with random delays, and in [9] PAoI is derived for a single-class $M/M/1$ queueing system.

Our work differs from these prior works in a number of ways. First, we focus on the PAoI metric, which is closely related to the AoI, but is much more tractable, thus facilitating its optimization. Second, we consider general service time distributions, whereas previous works focus mainly on exponential service time. Finally, we minimize the *cost* of the PAoI in a system with heterogenous service requirements; where the service requirements of different entities are modeled using quasi-convex cost functions of the PAoI.

It is important to note here that AoI is different from the traditional delay metric considered in communication systems. Indeed, our results show that PAoI minimization is equivalent to minimizing the sum of update interval and update packet delay. Due to this difference, the ultimate optimization problem turns out to be non-convex.

This paper is organized as follows. In Section II we present the system model. We derive the PAoI values for $M/G/1$ and $M/G/1/1$ in Section III. We consider the system cost optimization problem in Section IV, and present numerical results in Section V. We conclude the paper in Section VI.

II. System Model

We consider a system that consists of a set of N entities, denoted by $\mathcal{N} = \{1, ..., N\}$. To disseminate entity status information, the system regulates how frequently each entity updates its status. We denote this decision by an *update rate* vector $\boldsymbol{\lambda} = (\lambda_1, ..., \lambda_N)$, where λ_n is the update rate of entity

n. We assume that the update process is Poisson with rate λ_n. After generation, update packets are relayed to a central unit for processing. To ensure that each entity eventually updates its status, we require that $0 < \lambda_{\min} \leq \lambda_n \leq \lambda_{\max}$ for all n. We denote by $\Lambda \triangleq \{\boldsymbol{\lambda} : \lambda_{\min} \leq \lambda_n \leq \lambda_{\max}, \forall n\}$ the set of feasible rate vectors.

A. Single queue model

In a practical system, different update information streams will share the limited system resources for delivery. We model this by the system shown in Fig. 1, where all update messages go through a single server queue. This single server queueing model is the same as that adopted by prior work on AoI [4], [5], [8].

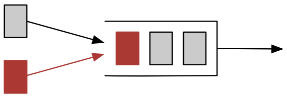

Fig. 1. The update packet delivery process in a 2-entity system.

In this queueing system, each new class-n packet arrival to the queue represents the generation of a new entity n update packet. Hence, class-n packets arrive according to a Poisson process with rate λ_n. Departures from the queue, on the other hand, represent update packet reception events at their destinations. To model the heterogeneous resource requirements, e.g., entities may have update messages with different length distributions, we assume that each update packet from entity n requires a random service time X_n, which is i.i.d. with mean x_n and second moment y_n. For convenience, we denote $x_{\max} = \max_n x_n$ and $y_{\max} = \max_n y_n$.

This system is indeed equivalent to a multi-class $M/G/1$ queue. This model captures two key features of communication in networked systems: (i) resources are shared among different update streams, and (ii) queueing can occur during traffic delivery. Adopting this simple model allows us to focus on the age aspect of the update information.

B. Age-of-Information

The *status age* of an entity at a particular time instance t is defined to be the time elapsed since its latest received update packet was generated. Fig. 2 shows the status age (denoted by $\Delta_n(t)$) of entity n. The dropping points of $\Delta_n(t)$ are the time instances when an update packet is received, which resets the age value to a lower level (i.e., the current time minus the generation time of the new update packet).

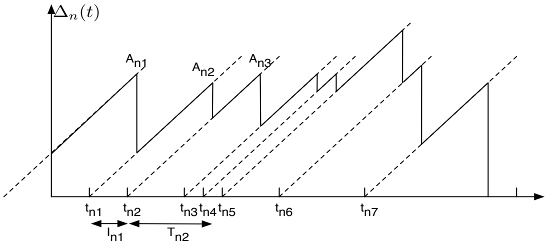

Fig. 2. The evolution of the status age of entity n in the system. Here A_{nk} denotes the k-th peak of age.

Given $\Delta_n(t)$, the average status age of entity n is defined as:

$$A_{av,n} = \lim_{T\to\infty} \frac{1}{T} \int_{t=0}^{T} \Delta_n(t)dt. \quad (1)$$

This metric is called the *age-of-information* (AoI) and was first considered in [4]. However, the AoI metric is hard to analyze. Moreover, in many systems, it is often the maximum status information delay that determines the performance loss [3]. Thus, we instead focus on the average *peak* status age. Specifically, let A_{nk} denote the k-th peak value of $\Delta_n(t)$ (See Fig. 2). The *peak age-of-information* (PAoI) metric $A_n(\boldsymbol{\lambda})$ is defined as:

$$A_{p,n}(\boldsymbol{\lambda}) \triangleq \lim_{K\to\infty} \frac{1}{K} \sum_{k=1}^{K} A_{nk}. \quad (2)$$

Here we explicitly express PAoI as a function of the update rate vector $\boldsymbol{\lambda}$. The PAoI metric was first considered in [9] for the $M/M/1/1$ queue. It represents the maximum age of information before a new update is received. PAoI is closely related to the previously considered AoI metric $A_{av,n}$, e.g., in [4] and [5], but is much more tractable, thus facilitating its optimization.[1]

C. Optimizing update rates

We model the system performance using a function of PAoI, to capture the fact that delay in status update often causes a proportional performance degradation. Specifically, we consider the following system cost function, i.e.,

$$C_{\text{sys}}(\boldsymbol{A}(\boldsymbol{\lambda})) \triangleq \max_n C_n(A_{p,n}(\boldsymbol{\lambda})). \quad (3)$$

Here $C_n(A_n)$ is the cost of entity n for having a PAoI of A_n, and it is assumed to be a quasiconvex and non-decreasing function in A_n with $C_n(0) = 0$. Our objective is to find an update rate vector $\boldsymbol{\lambda} \in \Lambda$ to minimize $C_{\text{sys}}(\boldsymbol{A}(\boldsymbol{\lambda}))$, i.e., minimize the maximum cost over all entities.

III. Computing PAoI

In this section, we compute the PAoI for two different queueing models, i.e., the $M/G/1$ model and the $M/G/1/1$ model. The two models differ in the way update packets are managed. In the $M/G/1$ model, new packets are queued if the server is busy, while they are discarded in the $M/G/1/1$ model. This packet management scheme was also considered in [9] for the $M/M/1$ model.

A. A general result for $G/G/1$ queues

We first derive a useful result for general $G/G/1$ queues. Denote by I_n the inter-arrival time of entity n packets, and let W_n be the waiting time of entity n packets and let T_n be the total sojourn time in the queue. We have the following proposition, in which the superscript "$gg1$" is used to indicate the relationship to $G/G/1$ queues.

Proposition 1: In a $G/G/1$ queue, the PAoI is given by:

$$A_{p,n}^{gg1} = \mathbb{E}\{I_n + T_n\} = \mathbb{E}\{I_n + X_n + W_n\}, \forall n. \quad \diamond \quad (4)$$

[1] Intuitively, PAoI provides an approximate upper bound for AoI, as it only considers the average sampled at the peak moments, whereas AoI in (1) computes the time average value of instantaneous age.

Proof: This relation follows from Fig. 2, where we see that the PAoI is equal to the time from the generation of an update packet until the completion of the next update packet, plus their inter-arrival time. ■

Equation (4) shows that PAoI is indeed the sum of the update interval and update packet delay. For a multi-class $G/G/1$ queue, we also know that the AoI is given by [4]:

$$A_{av,n}^{gg1} = \lambda_n \mathbb{E}\{I_n T_n + \frac{I_n^2}{2}\}. \tag{5}$$

Comparing (5) and (4), we have:

$$\begin{aligned} &A_{av,n}^{gg1} - A_{p,n}^{gg1} \qquad (6)\\ &= \lambda_n \bigg(\mathbb{E}\{I_n(T_n + \frac{I_n}{2})\} - \mathbb{E}\{I_n\}\mathbb{E}\{I_n + T_n\} \bigg) \\ &= \lambda_n \bigg(\mathbb{E}\{I_n T_n\} - \mathbb{E}\{I_n\}\mathbb{E}\{T_n\} + \frac{\mathbb{E}\{I_n^2\}}{2} - \mathbb{E}\{I_n\}^2 \bigg). \end{aligned}$$

Equation (6) provides a way for checking how close the two metrics are to each other. It can be seen that when I_n is a constant, i.e., periodic arrival, $A_{av,n}^{gg1} = A_{p,n}^{gg1} - \frac{\lambda I_n^2}{2}$. We will see in the next subsection that $A_{av,n}^{gg1} \leq A_{p,n}^{gg1}$ in the single class $M/M/1$ case. Thus, PAoI serves as an upper bound for AoI. More generally, the following lemma shows that PAoI approximates AoI for general single-class $G/G/1$ queues.

***Lemma** 1:* For a general single-class $G/G/1$ queue,

$$A_p^{gg1} - \frac{3\lambda\mathbb{E}\{I^2\}}{2} - \lambda\mathbb{E}\{I\}^2 \leq A_{av}^{gg1} \leq A_p^{gg1} + \lambda\mathbb{E}\{I^2\}/2. \diamond$$

Proof: See Appendix A. ■

B. *PAoI for multi-class $M/G/1$ queue*

Using (4), we can compute the PAoI for each entity in the $M/G/1$ queue.

Proposition 2: The PAoI for a multi-class $M/G/1$ system is given by:

$$A_{p,n}^{mg1} \quad = \quad \frac{1}{\lambda_n} + x_n + \frac{\sum_j \lambda_j y_j}{2(1 - \sum_j \lambda_j x_j)}. \quad \diamond \tag{7}$$

In (7) we have used the P-K formula to compute the waiting time in the $M/G/1$ queue [10]. It is necessary to ensure $\rho \triangleq \sum_j \lambda_j x_j < 1$, so that the queue is stable and the PAoI is finite.

We can use (7) to compute the PAoI for a single class $M/M/1$ system. In particular, with $N = 1$ and exponential service time of rate μ, (7) becomes:

$$A_p^{mm1} = \frac{1}{\mu}\bigg(1 + \frac{1}{\rho} + \frac{\rho}{1-\rho}\bigg). \tag{8}$$

In contrast, the AoI for $M/M/1$ derived in [4], is given by,

$$A_{av}^{mm1} = \frac{1}{\mu}\bigg(1 + \frac{1}{\rho} + \frac{\rho^2}{1-\rho}\bigg). \tag{9}$$

Comparing (7) to (9), we have,

$$A_p^{mm1} - A_{av}^{mm1} = \frac{1}{\mu}\rho = \frac{\lambda}{\mu^2}. \tag{10}$$

Hence, PAoI is a close upper bound of AoI for $M/M/1$ queues, yet is much more tractable.

Conservation Laws for PAoI: From (7), we also obtain the following *conservation* formula for PAoI:

$$\sum_n \lambda_n A_{p,n}^{mg1} \quad = \quad N + \rho + NW. \tag{11}$$

We also see from (7) that:

$$A_{p,n}^{mg1} - \frac{1}{\lambda_n} - x_n = A_{p,m}^{mg1} - \frac{1}{\lambda_m} - x_m, \, \forall \, n, m,$$

which implies,

$$\frac{1}{\lambda_n} - \frac{1}{\lambda_m} = (A_{p,n}^{mg1} - x_n) - (A_{p,m}^{mg1} - x_m). \tag{12}$$

Hence, the relationship between $A_{p,n}^{mg1}$ and $A_{p,m}^{mg1}$ is completely determined by λ_n and λ_m. For example, $\lambda_m > \lambda_n$ implies $A_{p,n}^{mg1} - x_n > A_{p,m}^{mg1} - x_m$.

C. *PAoI for $M/G/1/1$ queue*

Let us now consider the case when the server does not queue incoming update packets. This can be viewed as the server is performing packet management [9].

Proposition 3: The PAoI for a multi-class $M/G/1/1$ is given by:

$$A_{p,n}^{mg11} = x_n + \frac{1}{\lambda_n} + \frac{\sum_k \lambda_k x_k}{\lambda_n}, \, \forall \, n. \quad \diamond \tag{13}$$

Proof: Let Z_{jn} be the expected time to complete service for a class n packet starting from the moment a class j packet begins receiving service at the server. We have:

$$A_{p,n}^{mg11} = x_n + \frac{1}{\lambda} + \sum_j \frac{\lambda_j}{\lambda} Z_{jn}. \tag{14}$$

Equation (14) can be understood as follows. Since the status age of an entity decreases only when its next update packet is served, the (expected) peak age can be broken down into three components (see Fig. 3). The first component x_n in (14) is the processing time of the current update packet. The second component is $\frac{1}{\lambda}$, the expected time needed to get the next arrival (because packets arriving during busy periods are dropped). Then, the third component is the expected time needed until the completion of the next class n update packet (the third term), which is Z_{jn} if the next arrival turns out to be a class j packet, resulting in an average time of $\sum_j \frac{\lambda_j}{\lambda} Z_{jn}$.

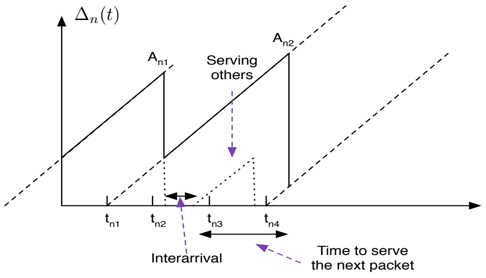

Fig. 3. Evolution of entity n's status age in the $M/G/1/1$ system.

We now solve for $A_{p,n}^{mg11}$. Note that for any $i \neq n$, we have:

$$Z_{ji} = x_j + \frac{1}{\lambda} + \sum_k \frac{\lambda_k}{\lambda} Z_{ki}. \tag{15}$$

Thus, $Z_{ji} - Z_{ki} = x_j - x_k$ for any $j, k \neq i$. Plugging this into (15) and using the fact that $Z_{ii} = x_i$, we obtain:

$$Z_{ji} = x_j + \frac{1}{\lambda} + \frac{\lambda_i}{\lambda} x_i + \sum_{k \neq i} \frac{\lambda_k}{\lambda}[Z_{ji} + x_k - x_j].$$

Therefore,

$$Z_{ji} = x_i + \frac{1}{\lambda_i} + x_j + \sum_{k \neq i} \frac{\lambda_k x_k}{\lambda_i} = x_j + \frac{1}{\lambda_i} + \sum_k \frac{\lambda_k x_k}{\lambda_i}.$$

Using this in (14), we get:

$$A_{p,n}^{mg11} = x_n + \frac{1}{\lambda} + \frac{\lambda_n}{\lambda} x_n + \sum_{j \neq n} \frac{\lambda_j}{\lambda} \left[x_j + \frac{1}{\lambda_n} + \sum_k \frac{\lambda_k x_k}{\lambda_n} \right]$$

$$= 2x_n + \frac{1}{\lambda_n} + \frac{\sum_{k \neq i} \lambda_k x_k}{\lambda_n}.$$

Rearranging the terms in the above gives (13). ■

It is interesting to note that (13) does not require the second moment of service time, which is generally required in the analysis of $M/G/1$ queues. This can be attributed to the fact that in the $M/G/1/1$ system, packets are never held in the buffer, thus the residual service time does not play a role in the computation of (13). Also, when $N = 1$, (13) recovers the result from [9] for the $M/M/1/1$ queue. It is also interesting to see in (13) that due to packet discard, the constraint $\rho < 1$ can actually be violated.

We note from (13) that, for any achievable PAoI vector $\boldsymbol{A}_p^{mg11} = (A_{p1}^{mg11}, ..., A_{pN}^{mg11})$,

$$\lambda_n A_{p,n}^{mg11} - \lambda_n x_n \quad = \quad \sum_k \lambda_k x_k + 1. \tag{16}$$

Since the right-hand-side (RHS) does not depend on n, this implies that:

$$\lambda_n A_{p,n}^{mg11} - \lambda_n x_n \quad = \quad \lambda_m A_{p,m}^{mg11} - \lambda_m x_m. \tag{17}$$

Similar to (12), the relationship of $A_{p,n}^{mg11}$ and $A_{p,m}^{mg11}$ is uniquely determined by λ_n and λ_m. We also have the following conservation formula:

$$\sum_n \lambda_n A_{p,n}^{mg11} \quad = \quad N + (N+1)\rho. \tag{18}$$

Comparing (7) and (13), we have:

$$A_{p,n}^{mg11} - A_{p,n}^{mg1} \quad = \quad \frac{\sum_k \lambda_k x_k}{\lambda_n} - \frac{\sum_j \lambda_j y_j}{2(1 - \sum_j \lambda_j x_j)}. \tag{19}$$

This shows that $A_{p,n}^{mg11}$ can be much smaller than $A_{p,n}^{mg1}$ when the update rates are large, i.e., when ρ is close to 1. Thus, even though packets can be dropped in the $M/G/1/1$ system, such dropping may actually result in PAoI reduction as queueing delay is reduced.

IV. PAoI OPTIMIZATION

Having computed the PAoI for the two cases, we now consider the problem of optimizing the update rates, i.e., minimize $C_{\text{sys}}(\boldsymbol{\lambda})$. This formulation enables us to provide differentiated service to different applications. In the following, we start with the $M/G/1/1$ queue and then consider the $M/G/1$ queue.

A. $M/G/1/1$ *optimization*

In this case, the utility optimization problem takes the following form:

$$\min_{\boldsymbol{\lambda}} : \quad C_{\text{sys}}(\boldsymbol{\lambda}) = \max_n C_n(x_n + \frac{1 + \sum_k \lambda_k x_k}{\lambda_n}) \tag{20}$$
$$\text{s.t.} \quad \boldsymbol{\lambda} \in \Lambda.$$

The following lemma shows that although (20) is not convex, it can still be efficiently solved.

***Lemma** 2:* Problem (20) is a quasiconvex program. ◇

Proof: First, we see that $A_n = x_n + \frac{1+\sum_k \lambda_k x_k}{\lambda_n}$ is a linear-fractional function in $\boldsymbol{\lambda}$. Since each $C_n(A_n)$ function is quasiconvex and nondecreasing in A_n, $C_n(A_n)$ is quasiconvex in $\boldsymbol{\lambda}$. As the $\max$ operator preserves quasiconvexity, we conclude that $C_{\text{sys}}(\boldsymbol{\lambda})$ is also quasiconvex in $\boldsymbol{\lambda}$ and (20) is a quasiconvex program over the convex set Λ [11]. ■

Therefore, the optimization problem (20) can be solved by the bisection procedure described below [11]. Define

$$\phi_t \triangleq \begin{cases} 0 & C_{\text{sys}}(\boldsymbol{\lambda}) \leq t \\ \infty & \text{else.} \end{cases} \tag{21}$$

We see that $C_{\text{sys}}(\boldsymbol{\lambda}) \leq t$ is equivalent to $\phi_t \leq 0$, i.e., if $\boldsymbol{\lambda}$ ensures $\phi_t \leq 0$, it also ensures $C_{\text{sys}}(\boldsymbol{\lambda}) \leq t$. Hence, we then use the following bi-section algorithm to solve (20).

`Bisection`: Set $l = 0$ and $u = \max_n C_n(x_{\max} + \frac{1+N\lambda_{\max} x_{\max}}{\lambda_{\min}})$. Fix $\epsilon > 0$. Then, repeat the following until $u - l \leq \epsilon$:

1) Set $t = (l+u)/2$
2) Solve the following problem:
$$\min: \ 1, \quad \text{s.t. } \phi_t \leq 0, \ \boldsymbol{\lambda} \in \Lambda. \tag{22}$$
3) If (22) is feasible, set $u = t$; otherwise set $l = t$. ◇

Using (17), we also obtain the following properties of the optimal solution $\boldsymbol{\lambda}^*$ to (20), where $\succeq$ denotes entrywise larger.

***Lemma** 3:* Let $\boldsymbol{\lambda}^*$ be an optimal solution of (20). Then, (i) $\exists \ \hat{\boldsymbol{\lambda}} \succeq \boldsymbol{\lambda}^*$ such that $\max_n \hat{\lambda}_n = \lambda_{\max}$ and $C_{\text{sys}}(\hat{\boldsymbol{\lambda}}) = C_{\text{sys}}(\boldsymbol{\lambda}^*)$, and (ii) if all entities are identical, i.e., $x_1 = x_2$ and $C_n(A) = C_m(A)$, then $\lambda_n = \lambda_{\max}$, $\forall n$ is an optimal solution. ◇

Proof: First, note from (17) that for a given $\boldsymbol{\lambda}$, if we let $n_0 = \arg\min_n (A_n - x_n)$, then we can express each λ_m as:

$$\lambda_m = \lambda_{n_0} \frac{A_{n_0} - x_{n_0}}{A_m - x_m}, \ \forall m. \tag{23}$$

Consider an optimal solution $\boldsymbol{\lambda}^*$ and let $\boldsymbol{A}^*$ be the corresponding PAoI vector. We construct a $\hat{\boldsymbol{\lambda}}$ as follows. Denote $n_0^* = \arg\min_n (A_n^* - x_n)$. Then, we keep the ratio between any pair of rates fixed and proportionally increase all λ_n until $\lambda_{n_0^*} = \lambda_{\max}$. From (17) and (13), we see that $\hat{\boldsymbol{A}} \preceq \boldsymbol{A}^*$. Since each $C_n(A)$ is nondecreasing, we have $C_n(\hat{A}_n) \leq C_n(A_n^*)$, which implies $C_{\text{sys}}(\hat{\boldsymbol{\lambda}}) = C_{\text{sys}}(\boldsymbol{\lambda}^*)$. This proves (i).

When all entities are identical, the rates must be the same for all entities. Hence, $\lambda_n = \lambda_{\max}$ for all n. ■

B. $M/G/1$ *optimization*

In $M/G/1$, the optimization problem becomes:

$$\min: \quad C_{\text{sys}}(\boldsymbol{\lambda}) = \max_n C_n(A_{p,n}^{mg1}) \tag{24}$$
$$\text{s.t.} \quad \frac{1}{\lambda_n} + x_n + \frac{\sum_j \lambda_j y_j}{2(1 - \sum_j \lambda_j x_j)} = A_{p,n}^{mg1}, \ \forall n \tag{25}$$
$$\sum_n \lambda_n x_n \leq 1, \ \lambda_n > 0.$$

Different from problem (20), here the LHS in constraint (25) is a sum of two linear-fractional functions, which may not be quasiconvex any more. Thus, to proceed, we approximate $A_n(\boldsymbol{\lambda})$ with another function $B_n(\boldsymbol{\lambda})$ defined as:

$$B_n(\boldsymbol{\lambda}) \triangleq 2 \max\left(\frac{1}{\lambda_n} + x_n, \frac{\sum_j \lambda_j y_j}{2(1 - \sum_j \lambda_j x_j)} \right). \tag{26}$$

That is, we solve problem (24) with $A_{p,n}^{mg1}$ replaced by B_n. The main advantage of introducing $B_n(\boldsymbol{\lambda})$ is that it is quasiconvex in $\boldsymbol{\lambda}$. Hence, the function $C_{\text{sys}}(\boldsymbol{B}(\boldsymbol{\lambda})) \triangleq \max_n C_n(B_n)$ can be efficiently minimized by the `Bisection` algorithm.

We now look at the performance of the approximation. Define β_n the maximum increasing slope of $C_n(A)$, i.e., $\beta_n \triangleq \inf\{\beta : |C_n(A_1) - C_n(A_2)| \leq \beta|A_1 - A_2|, \forall\, A_1, A_2\}$. We then have the following lemma, where $\boldsymbol{\lambda}_B^*$ is the optimal solution of the approximation program.

***Lemma** 4:* Let $\boldsymbol{\lambda}^*$ be an optimal solution of the original problem (24) and denote $\boldsymbol{A}^*$ the resulting PAoI. Then,

$$C_{\text{sys}}(\boldsymbol{\lambda}^*) \leq C_{\text{sys}}(\boldsymbol{\lambda}_B^*) \leq C_{\text{sys}}(\boldsymbol{\lambda}^*) + \max_n \beta_n A_n^*. \quad \diamond \qquad (27)$$

Proof: See Appendix B. ■

When each $C_n(A)$ is linear in A, i.e., $C_n(A) = w_n A$, we have $\beta_n = w_n$. In this case, we can conclude from (33) that $C_{\text{sys}}(\boldsymbol{\lambda}^*) \leq C_{\text{sys}}(\boldsymbol{\lambda}_B^*) \leq 2C_{\text{sys}}(\boldsymbol{\lambda}^*)$.

V. Numerical Results

We present a simple numerical example with $N = 2$ entities, and constant service times $x_1 = 1$ and $x_2 = 3$. $\lambda_{\min} = 0.01$ and $\lambda_{\max} = 10$. The cost functions are given by $C_1(A_1) = 4A_1^2$ and $C_2(A_2) = A_2^2$, and $C_{\text{sys}}(\boldsymbol{\lambda}) = \max(C_1, C_2)$.

In Fig. 4 we plot the cost values for the $M/G/1/1$ queue. The minimum value of $C_{\text{sys}}(\boldsymbol{\lambda})$ is achieved at $\lambda_1 = 10$ and $\lambda_2 = 6$, with a resulting $C_1 = 60.84$ and $C_{\text{sys}}(\boldsymbol{\lambda}) = C_2 = 61.36$. In this case, the PAoI vector is $\boldsymbol{A} = (3.9, 7.83)$ and the results match Lemma 3.

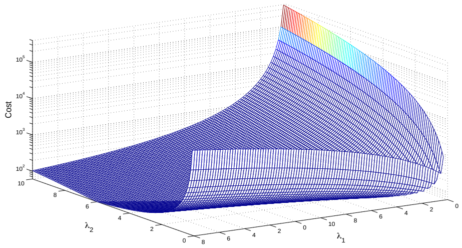

Fig. 4. Cost value for the $M/G/1/1$ queue.

We next look at the $M/G/1$ case in Fig. 5. Since $\rho < 1$ must be satisfied to ensure a finite PAoI, if a $\boldsymbol{\lambda}$ violates $\rho < 1$, we set its PAoI value to a constant (the flat region). In this case, the minimum is achieved at $\boldsymbol{\lambda} = (0.29, 0.125)$ with $\boldsymbol{A} = (6.56, 13.11)$. Thus, $C_2 = 171.92$ and $C_1 = C_{\text{sys}}(\boldsymbol{\lambda}) = 172.15$. It can be verified that (12) holds.

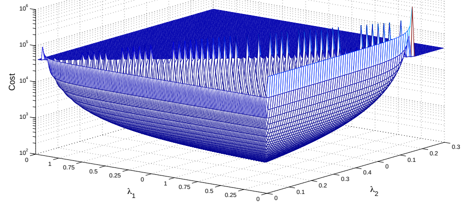

Fig. 5. Cost value for the $M/G/1$ queue.

We also compute the optimal solution of the approximation approach to be $\boldsymbol{\lambda}_B^* = (0.285, 0.17)$. The resulting PAoI vector is $\boldsymbol{A} = (8.94, 13.31)$ and the cost vector is $(C_1, C_2) = (319.69, 177.16)$, implying $C_{\text{sys}}(\boldsymbol{\lambda}_B^*) = 319.69$. It is not hard to verify that $C_{\text{sys}}(\boldsymbol{\lambda}_B^*) \leq 2C_{\text{sys}}(\boldsymbol{\lambda}^*)$.

VI. Conclusion

We study the age-of-information in a general multi-class $M/G/1$ queueing system. The age-of-information is a new metric for system performance that represents not just the queueing delay, but also the delay in generating new information updates. Our main contribution is to generalize the available results to systems with heterogeneous service time distributions, accounting for the fact that different entities may have different service requirements for their status updates. We derive exact peak-age-of-information expressions for both a $M/G/1$ system and $M/G/1/1$ system. Using the PAoI measure allows us to optimize system cost, as a function of PAoI, by choice of the update interval.

VII. Acknowledgement

This work was supported in part by the National Basic Research Program of China Grant 2011CBA00300, 2011CBA00301, the National Natural Science Foundation of China Grant 61033001, 61361136003, 61303195, Tsinghua Initiative Research Grant, and the China Youth 1000-talent Grant. The work of E. Modiano was supported by NSF Grant CNS-1217048 and ONR Grant N00014-12-1-0064.

References

[1] RFID Journal. Microsoft uses wireless sensors to track data center temperatures. *http://www.rfidjournal.com/articles/view?4587*, Feb 2009.

[2] J. Kim, H. Kim, K. Lakshmanan, and R. Rajkumar. Parallel scheduling for cyber-physical systems: analysis and case study on a self-driving car. *Proceedings of the ACM/IEEE 4th International Conference on Cyber-Physical Systems*, 2013.

[3] A. Reddy, S. Banerjee, A. Gopalan, S. Shakkottai, and L. Ying. On distributed scheduling with heterogeneously delayed network-state information. *Queueing Systems, Vol. 72, No. 3-4, pp 193-218*, Dec 2012.

[4] S. Kaul, R. Yates, and M. Gruteser. Real-time status: How often should one update? *Proceedings of INFOCOM mini-conference*, 2012.

[5] R. D. Yates and S. Kaul. Real-time status updating: Multiple sources. *Proceedings of ISIT*, 2012.

[6] S. Kaul, R. Yates, and M. Gruteser. On piggybacking in vehicular networks. *Proceedings of Globecom*, 2011.

[7] S. Kaul, M. Gruteser, V. Rai, and J. Kenney. Minimizing age of information in vehicular networks. *Proceedings of SECON*, 2011.

[8] C. Kam, S. Kompella, and A. Ephremides. Age of information under random updates. *Proceedings of ISIT*, 2012.

[9] M. Costa, M. Codreanu, and A. Ephremides. Age of information with packet management. *Proceedings of ISIT*, 2014.

[10] D. P. Bertsekas and R. G. Gallager. *Data Networks*. Prentice Hall, 1992.

[11] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

Appendix A – Proof of Lemma 1

Here we prove Lemma 1. We drop all subscripts as $N = 1$.

Proof: Since $T = W + X$ and X is independent of I, we have $\mathbb{E}\{I_n T_n\} - \mathbb{E}\{I_n\}\mathbb{E}\{T_n\} = \mathbb{E}\{IW\} - \mathbb{E}\{I\}\mathbb{E}\{W\}$. Denote T_Q the time it takes to clear the packets in the queue when a new packet arrives (not including the new arrival). Then, $W = (T_Q - I)^+$. Here I is the inter-arrival time until the next packet arrives. Since I is independent of T_Q, we have:

$$\begin{aligned}&\mathbb{E}\{IW\}\\ =\ &\int_I \int_{t=I}^{\infty} I(t-I)f(t)dtf(I)dI\end{aligned}$$

$$\begin{aligned} &= \int_I \left(I \int_{t=I}^{\infty} tf(t)dt - I^2 \text{Pr}\{T_Q \geq I\} \right) f(I)dI \quad (28) \\ &\leq \int_I I \int_{t=I}^{\infty} tf(t)dtf(I)dI. \quad (29) \end{aligned}$$

Using $(T_Q - I)^+ + I \geq T_Q$ and $I \geq 0$, we have:

$$\int_{t=I}^{\infty} tf(t)dt \leq \mathbb{E}\{T_Q\} \leq \mathbb{E}\{(T_Q - I)^+\} + \mathbb{E}\{I\}. \quad (30)$$

Plugging this into (29), we get:

$$\begin{aligned} \mathbb{E}\{IW\} &\leq \int_I I\big(\mathbb{E}\{(T_Q - I)^+\} + \mathbb{E}\{I\}\big) f(I)dI \\ &= \mathbb{E}\{I\}\mathbb{E}\{(T_Q - I)^+\} + \mathbb{E}\{I\}^2. \quad (31) \end{aligned}$$

Using this in (6), we obtain:

$$A_{av}^{gg1} \leq A_p^{gg1} + \frac{\lambda \mathbb{E}\{I^2\}}{2}.$$

To derive the lower bound, we have from (28) that:

$$\begin{aligned} \mathbb{E}\{IW\} &= \int_I \left(I \int_{t=I}^{\infty} tf(t)dt - I^2 \text{Pr}\{T_Q \geq I\} \right) f(I)dI \\ &\geq \int_I I \left(\mathbb{E}\{T_Q\} - \int_{t=0}^{I} tf(t)dt \right) f(I)dI - \mathbb{E}\{I^2\} \\ &\geq \int_I I \left(\mathbb{E}\{T_Q\} - I \right) f(I)dI - \mathbb{E}\{I^2\} \\ &\geq \mathbb{E}\{I\}\mathbb{E}\{W\} - 2\mathbb{E}\{I^2\}. \quad (32) \end{aligned}$$

In the last inequality, we have used $T_Q \geq W$. Plugging (32) into (6) proves the lower bound and completes the proof of the lemma. ■

APPENDIX B – PROOF OF LEMMA 4

We prove Lemma 4 here.

Proof: From the definition of B_n, we see that given any $\boldsymbol{\lambda}$, $A_{p,n}^{mg1}(\boldsymbol{\lambda}) \leq B_n(\boldsymbol{\lambda}) \leq 2A_{p,n}^{mg1}(\boldsymbol{\lambda})$. Thus, we have for $\boldsymbol{\lambda}_B^*$ that:

$$\begin{aligned} &C_{\text{sys}}(\boldsymbol{A}(\boldsymbol{\lambda}_B^*)) \leq C_{\text{sys}}(\boldsymbol{B}(\boldsymbol{\lambda}_B^*)) \quad (33) \\ &\qquad \leq C_{\text{sys}}(\boldsymbol{B}(\boldsymbol{\lambda}^*)) \leq C_{\text{sys}}(2\boldsymbol{A}(\boldsymbol{\lambda}^*)). \end{aligned}$$

Using the definition of β_n, we have for each n that:

$$C_n(2A_n(\boldsymbol{\lambda}^*)) \leq C_n(A_n(\boldsymbol{\lambda}^*) + \beta_n A_n^*. \quad (34)$$

Taking the $\max$ over the above and combining it with (33), we see that (27) follows. ■