

Age of Information Optimization and State Error Analysis for Correlated Multi-Process Multi-Sensor Systems

Egemen Erbayat

erbayat@gwu.edu

The George Washington University
Washington, DC, USA

Peng Zou

pzou94@gwu.edu

The George Washington University
Washington, DC, USA

Ali Maatouk

ali.maatouk@yale.edu

Yale University
New Haven, Connecticut, USA

Suresh Subramaniam

suresh@gwu.edu

The George Washington University
Washington, DC, USA

ABSTRACT

In this paper, we examine a multi-sensor system where each sensor may monitor more than one time-varying information process and send status updates to a remote monitor over a single channel. We consider that each sensor's status update may contain information about more than one information process in the system subject to the system's constraints. To investigate the impact of this correlation on the overall system's performance, we conduct an analysis of both the average Age of Information and source state estimation error at the monitor. Building upon this analysis, we subsequently explore the impact of the packet arrivals, correlation probabilities, and rate of processes' state change on the system's performance. Next, we consider the case where sensors have limited sensing abilities and distribute a portion of their sensing abilities for different processes. We optimize this distribution to minimize the total AoI of the system. Interestingly, we show that monitoring multiple processes from a single source may not always be beneficial. Additionally, our results highlight that the optimal sensing distribution for diverse arrival rates may exhibit a fast regime change instead of undergoing smooth changes.

1 INTRODUCTION

In the rapidly developing landscape of networked systems, timeliness plays an essential role in multiple aspects of communication, decision-making, and information processing, contributing significantly to the efficiency and effectiveness of systems. The Age of Information (AoI) metric proposed in [19] stands as an important measure, reflecting the timeliness of information delivery in communication networks. In recent years, AoI has been well-studied in a single server setting with single or multiple sources [8, 14, 15, 25, 26, 31, 32, 36, 37], scheduling problems for different settings [13, 17, 22–24, 27], resource-constrained systems [1–3, 9, 28].

In sensor networks, collaborative sensing between the different components of the network has been shown to aid in improving the overall performance of the network [12]. Particularly, in such scenarios, numerous small sensor devices are strategically scattered around an area, monitoring different processes and sending updates to one or multiple central controllers [29]. Home security systems with multiple motion sensors are a good example of how devices can work together to improve efficiency. Each sensor can focus on a specific area and send status updates for that area. However, if there is an overlap between the fields of view of different sensors,

they can share information about those areas. This collaboration between devices is known as correlation. In scenarios where network resources are constrained, such collaboration can strengthen the system's efficacy and efficiency. It resembles orchestrating a network of compact, intelligent devices working in unison to gather and exchange data, thereby enabling thorough and punctual monitoring.

The correlation among sources of this nature is crucial for optimizing the AoI. When information sources that perceive the same aspects, such as motion detection, are densely deployed, there is a correlation among status updates in both spatial and temporal dimensions. In [11], a sensor network that has overlapping fields is considered, and the authors present a joint optimization approach for fog node assignment and transmission scheduling for sensors to minimize the age of multi-view image data. Similarly, the authors consider cameras that monitor overlapping areas as a correlation and propose scheduling algorithms for multi-channel systems in [34]. In [18, 35], the authors propose probability-based correlation models and present sensor scheduling policies aimed at minimizing AoI. Authors in [30] model correlation as a discrete-time Wiener process and offer a scheduling policy that considers AoI and monitoring error. All of the papers studied correlation with given parameters. Still, there is a need for further research on how changes in correlation affect system performance and what the optimal correlation parameters are under constraints, as previous works only focus on scheduling problems with given correlation parameters.

In this paper, we introduce a system model that consists of processes, sensors observing those processes, and a single server to monitor the states of physical processes. First, we evaluate the estimation error performance for each process separately, as well as AoI, in relation to correlation parameters.

In addition, we evaluate correlation probabilities to optimize AoI under certain constraints for different scenarios. There are three possible scenarios regarding the impact of the number of processes tracked by a sensor on its performance. These are: 1) The performance of a sensor is not affected by the number of processes it tracks. 2) When a sensor tracks more processes, its performance improves. 3) Tracking more processes weakens the performance of a sensor. Finally, we investigate the impact of correlations on those metrics with different arrival rates and state change rates by providing numerical results. The main contributions of this paper are summarized as follows:

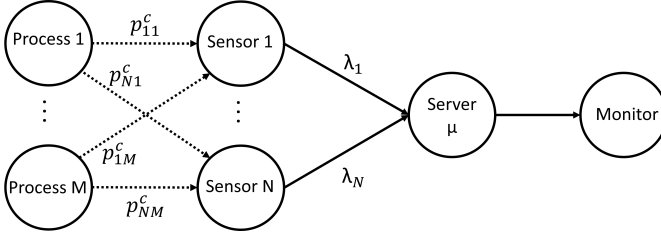


Figure 1: Illustration of our system model.

- As a first step, we introduce the system model, taking into account the correlation at hand. Then, through analysis of this system, we present an equivalent system from any process's perspective.
- Following that, we analyze the AoI and state estimation error metrics, formulating closed-form expressions for their averages in the considered M/M/1/1 system. Taking into account all possible events, our calculations are based on stochastic analysis of the average AoI and error ratio.
- We evaluate the distribution of correlations to optimize AoI for three different scenarios. We define sensor constraint functions to implement those scenarios, and we derive conditions for an optimal solution when the sensor constraints preserve convexity.
- Last, we present numerical implementations to validate the closed-form expressions we derived. We then compare different parameter configurations, focusing on their average AoI and error ratio. Our results highlight the importance of having a correlation in minimizing both AoI and estimation error.
- After verifying our analysis and the positive effect of having correlation, we present optimal solutions for three different scenarios in the correlation probability distribution problem. Equal distribution of all sensor sensing abilities among processes is one of the optimal solutions in the first two scenarios, which is found by KKT conditions. For the last scenario, the optimal distribution policy has a dramatic regime switch after a threshold.

The rest of the paper is organized as follows. We present the system model in Section 2. Afterward, we formulate the equivalent and simplified system in Section 3. The analysis of AoI and error ratio is then conducted in Section 4 and Section 5, respectively. In Section 6, we put our optimization problem into perspective and propose solutions to find the optimal sensing distribution. Finally, we present numerical results in Section 7, and we present our conclusions in the last section, Section 8.

2 SYSTEM MODEL

Let us consider a sensor network where N sensors monitor M information processes. To keep the monitor updated, each sensor generates status updates and sends them through a common server/channel, as shown in Figure 1. We consider that the service time of each packet is exponentially distributed with a service rate μ . We also assume that sensor i generates packets according to a Poisson process of rate λ_i . We suppose that the server has no buffer,

given the literature that has shown its optimality for AoI minimization [4]. Accordingly, any arriving packet that finds the server busy is dropped [5]. With all the above in mind, we define λ as a vector representing the arrival rates from the different sensors, where λ_i is the arrival rate from sensor i for $i = 1, \dots, N$. Specifically, we have:

$$\lambda^T = [\lambda_1 \quad \lambda_2 \quad \dots \quad \lambda_N]. \quad (1)$$

As for the information process, we consider that each physical process evolves as a time-varying discrete stochastic process. Particularly, the physical process j is modeled as a Markov chain with K different states. To represent these state changes, we use Ω_{ab}^j to denote the transition probability from state a to state b of process j . In matrix form, the transition matrix $\Omega_j \in [0, 1]^{K \times K}$ can be defined as follows:

$$\Omega_j = \begin{bmatrix} \Omega_{11}^j & \Omega_{12}^j & \dots & \Omega_{1K}^j \\ \Omega_{21}^j & \Omega_{22}^j & \dots & \Omega_{2K}^j \\ \vdots & \vdots & \ddots & \vdots \\ \Omega_{K1}^j & \Omega_{K2}^j & \dots & \Omega_{KK}^j \end{bmatrix}. \quad (2)$$

The corresponding Markov Chain for state changes is shown in Figure 2.

We assume that transitions for process j occur at exponentially distributed epochs with a rate of ζ_j . Accordingly, the generation of the status updates by the sensors and the information process changes are decoupled. Decoupling the generation of status updates from the information processing changes enables each process's status to evolve independently, irrespective of whether it's being actively tracked or not, ensuring a more resilient and adaptable system architecture capable of accommodating diverse operational scenarios and requirements.

In this paper, we consider that the Markov chain in Figure 2 is irreducible and aperiodic. To that end, we can conclude the existence and uniqueness of the chain's stationary distribution. We denote the stationary distribution of the Markov chain formed with Ω_j by:

$$\psi_j = [\psi_1^j \quad \psi_2^j \quad \dots \quad \psi_K^j]. \quad (3)$$

This stationary distribution can be obtained by solving the equation

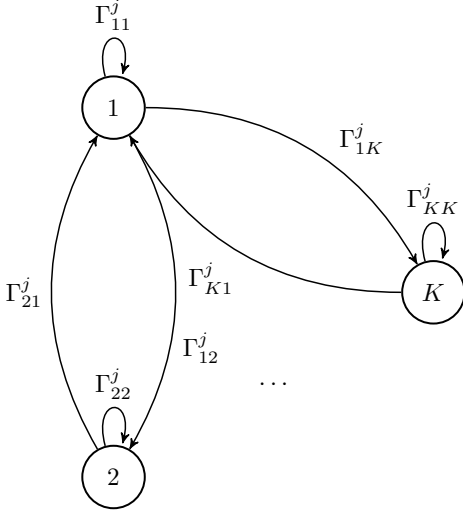
$$\Omega_j \cdot \psi_j = \psi_j, \quad (4)$$

and normalizing the resulting vector to ensure that the sum of its components is equal to 1 [7].

To model the correlation among the different sensor observations, we assume that each packet generated by sensor i contains information about the process j with a correlation probability p_{ij}^c . The information the packet has is the state of the processes at the generation time of the packet. To that end, we define the correlation matrix $\mathbf{P}_C \in [0, 1]^{N \times M}$ as follows:

$$\mathbf{P}_C = \begin{bmatrix} p_{11}^c & p_{12}^c & \dots & p_{1M}^c \\ p_{21}^c & p_{22}^c & \dots & p_{2M}^c \\ \vdots & \vdots & \ddots & \vdots \\ p_{N1}^c & p_{N2}^c & \dots & p_{NM}^c \end{bmatrix} \quad (5)$$

After having outlined the system model, we now proceed to formulate the equivalent and simplified system in Section 3.

Figure 2: Markov chain model of a physical process j .

3 SYSTEM SIMPLIFICATION THROUGH EQUIVALENCE

In the considered system, the originator of the packet containing information about any arbitrary process j is irrelevant from the monitor's perspective. In fact, concerning process j , what matters to the monitor is whether the served packet contains information about process j or not rather than which sensor provided the update. To that end, we define the informative arrival rate vector λ^* as follows:

$$\lambda^{*T} = [\lambda_1^* \quad \lambda_2^* \quad \dots \quad \lambda_M^*] = \lambda^T \mathbf{P}_C, \quad (6)$$

where λ_j^* represents the informative arrival rate for process j . As a last step, we let λ_C be the arrival rate of the server, which is the sum of all arrival rates, as follows:

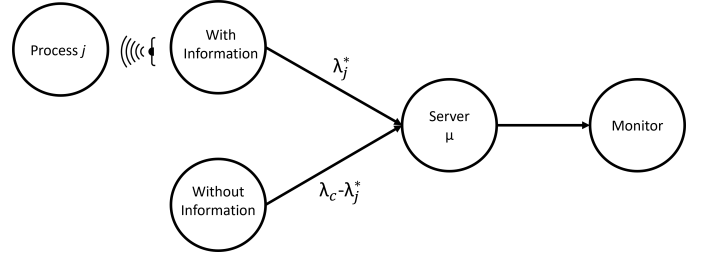
$$\lambda_C = \sum_{i=1}^N \lambda_i. \quad (7)$$

With the above entities in mind, we provide the following system equivalence lemma.

LEMMA 1. *Consider process j among M processes. From the monitor's perspective, the system is equivalent to Figure 3. In this equivalent system, there are two packet sources: packets with information and packets without information. The arrival rate of packets with information is λ_j^* , while the arrival rate of packets without information is $\lambda_C - \lambda_j^*$.*

PROOF. The details can be found in Appendix A. \square

Using the above equivalence, we can analyze the environment by reducing the original system to M independent systems, each with two sources as depicted in Fig. 3. We label a status update as informative for process j if it contains information on the process j . Otherwise, we label it as uninformative. This categorization helps in a way that if the packet is uninformative for process j , then it does not affect anything about process j on the monitor side. We need to consider whether the status update has information on

Figure 3: Equivalent system model from process j 's perspective.

process j or not and do our calculations based on informative status updates. In the upcoming section on AoI analysis, we calculate the AoI for each process, taking into account both informative and uninformative status updates, to comprehensively evaluate their impact on the system's dynamics.

4 AGE OF INFORMATION ANALYSIS

In this section, we consider the age function introduced in [19] as a performance metric. Mathematically, the AoI of process j at time t , denoted by $\Delta_j(t)$, can be defined as:

$$\Delta_j(t) = t - T_j, \quad (8)$$

where T_j represents the time at which the most recent informative packet for process j was generated. Particularly, the age at the monitor for each process j increases linearly over time until an informative status update is received, upon which a drop in the age takes place. As mentioned in Section 2 and Section 3, the packet in the server may or may not have information about each process. If the served packet has information about process j , the AoI for process j decreases just after the end of the service time. However, if the served packet has no information on process j , the AoI for process j continues to increase linearly. Let t_k denote the time instant when the k -th packet is generated, and t'_k represents the time instant when this packet completes service. When the server is busy with an informative or uninformative packet upon the arrival of a new packet, the new packet is dropped. To that end, we denote by t_n^d the time instant when the n -th dropped packet was generated. We define T_k as the service time of k -th packet, given by

$$T_k := t'_k - t_k, \quad (9)$$

and Y_k as the inter-departure time between two consecutive packets, given by

$$Y_k := t'_k - t'_{k-1}. \quad (10)$$

We also define \tilde{Y}_j^l as the inter-departure time between the l -th informative packet and the $(l-1)$ -th informative packet from process j 's perspective. Since Y_k shares the same distribution for all k , we define the random variable Y to represent them collectively. Similarly, considering that \tilde{Y}_j^l shares the same distribution for all l , we define the random variable \tilde{Y}_j to represent them as a group. To understand the AoI process better, we illustrate the evolution of the AoI in Figure 4. The age of information for process 1 at the destination node follows a linear increase over time. When a new informative status update is received, the age is reset to the time

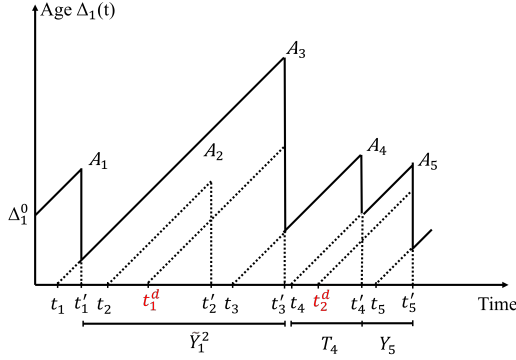


Figure 4: Evolution of AoI for process 1.

difference between the current time instant and the timestamp of the received update (A_1). However, if the status update is uninformative for process 1, the age continues to increase linearly (A_2). The packets arriving at times t_1^m and t_2^m are dropped. The server is occupied with an uninformative packet at time t_1^m and an informative packet at time t_2^m for process 1.

Then, we define the effective arrival rate as the rate of packets that arrive when the server is idle. Let λ_j^e be the effective arrival rate for packets that are informative for process j . Consequently, we have

$$\lambda_j^e = \frac{\mu \lambda_j^*}{\mu + \lambda_C}. \quad (11)$$

Afterwards, AoI for process j is shown in Lemma 2 in terms of \tilde{Y}_j , μ and λ_j^e .

LEMMA 2. The average AoI Δ_j for process j is:

$$\Delta_j = \lambda_j^e \left(\frac{1}{2} \mathbb{E}[\tilde{Y}_j^2] + \frac{\mathbb{E}[\tilde{Y}_j]}{\mu} \right). \quad (12)$$

PROOF. We adapt the methodology outlined in [8] to our scenario by leveraging their approach of utilizing inter-arrival times for calculating the age of information. Despite the absence of informative categorization in [8], their method of deriving the age of information based on inter-arrival times remains applicable. The main modification involves substituting variables to align with our informative arrivals. \square

To have the relationship between \tilde{Y}_j and Y_j , we define \tilde{p}_j as the probability of serving an informative packet for process j , as shown below:

$$\tilde{p}_j = \frac{\lambda_j^*}{\lambda_C}. \quad (13)$$

After defining \tilde{p}_j , we utilize it to establish the relationship between \tilde{Y}_j and Y_j . Using the relationship, we subsequently present the AoI for process j in Lemma 1, formulated in terms of μ , Y_j , and λ_j^e .

THEOREM 1. In the M/M/1/1 system, the average AoI for process j is:

$$\Delta_j = \frac{\lambda_C}{\lambda_C + \mu} \left(\frac{\mu \mathbb{E}[Y^2]}{2} + \frac{\mu \mathbb{E}[Y](1 - \tilde{p}_j)}{\tilde{p}_j} + \mathbb{E}[Y] \right). \quad (14)$$

PROOF. The details can be found in Appendix B. \square

To provide an interpretation of the above formula, we can see that if the value of \tilde{p}_j is 1, it means that every packet is informative for process j . In such a scenario, the system acts like a single sensor system, continuously sending status updates of process j from its perspective. Here, Δ_j represents the average AoI in an M/M/1/1 system, as described in [8]. On the other hand, as \tilde{p}_j approaches 0, the packets become uninformative. As a result, the AoI tends to infinitely, since informative status updates are received more and more infrequently.

5 ERROR RATIO ANALYSIS

We define a binary function $\epsilon_j(t)$ such that if the state information the monitor has for process j is the same as the state of process j at time t , then $\epsilon_j(t)$ is equal to 1. Otherwise, it is equal to 0. Then, we define the error of process j as the ratio of the total duration when $\epsilon_j(t)$ is equal 0 over the entire time horizon, denoted by ϵ_j . Particularly, we have

$$\epsilon_j = 1 - \lim_{T \rightarrow \infty} \frac{1}{T} \int_0^T \epsilon_j(t) dt. \quad (15)$$

Due to the nature of our system, ϵ_i and ϵ_j are independent for any $i, j \in \{1, \dots, N\}$ with $i \neq j$. The reason behind that is that state changes of two processes are independent of each other, and the system functions as N different independent systems as demonstrated in Section 3. Therefore, we derive the generic error ϵ for any process to simplify our analysis. Particularly, we drop the index j of the considered entities, and we use ϵ , $\epsilon(t)$, ζ , λ^* and Ω to denote the system parameters in the remainder of the paper.

To find ϵ analytically, we investigate a Markov Chain that considers the current process state, the state the monitor has, and the state of the served packet (defined below). Our proposed Markov chain is 3-dimensional with dimensions (x, y, z) . The states x and y represent the current state and the monitor state, respectively, and each can take values between 1 and K . The state z can have three possible values as depicted below:

- $z = 0$: This state indicates that the server is currently idle.
- $z = 1$: This state signifies that the server is actively serving a packet containing information from the process of interest.
- $z = 2$: In this state, the server is occupied by a packet, but this packet does not contain information about the process in question. In other words, it carries information about the other processes.

Note that when x and y are identical, the current state and the state the monitor has are identical. To this end, we redefine $\epsilon(t)$ as

$$\epsilon(t) = \begin{cases} 1 & \text{if } x = y, \\ 0 & \text{otherwise.} \end{cases} \quad (16)$$

With all the above in mind, we note that the system has three types of events: packet arrivals, packet departures, and state changes. Each event causes a transition in this three-dimensional Markov

Chain. Let \mathbf{P}_M and π be the transition probability matrix corresponding to those transitions and the stationary distribution of the three-dimensional Markov chain. Note that we consider that \mathbf{P}_M is irreducible and aperiodic in order to ensure the existence and uniqueness of the stationary distribution. In fact, given that the one-dimensional Markov chain shown in Section 2 characterized by both irreducibility and aperiodicity is used to form the considered three-dimensional Markov Chain, the three-dimensional Markov chain is irreducible and aperiodic. Next, we let $\pi(x, y, z)$ denote the stationary probability of being at state (x, y, z) . Then, by definition, the following equation is verified:

$$\sum_{x=1}^K \sum_{y=1}^K \sum_{z=0}^2 \pi(x, y, z) = 1. \quad (17)$$

In order to understand the stationary distribution of a Markov Chain, we first need to calculate the probability of state change until the packet is served, given that the server is occupied with an informative packet. To do this, we need to find the probability of the process transitioning from state i to state j , denoted as p_{ij}^n , while an informative packet is being served. We derive p_{ij}^n for all $i = 1, \dots, N$ and $j = 1, \dots, M$ in the following lemma, Lemma 3. Then, we form \mathbf{P}_M in Lemma 4.

LEMMA 3. Let $\mathbf{P}_N \in [0, 1]^{K \times K}$ represent the matrix of elements p_{ij}^n 's. The matrix can be obtained as follows:

$$\mathbf{P}_N = \begin{bmatrix} p_{11}^n & p_{12}^n & \cdots & p_{1K}^n \\ p_{21}^n & p_{22}^n & \cdots & p_{2K}^n \\ \vdots & \vdots & \ddots & \vdots \\ p_{K1}^n & p_{K2}^n & \cdots & p_{KK}^n \end{bmatrix} = \frac{\mu}{\mu + \zeta} \left(\mathbf{I} - \frac{\zeta \Omega}{\mu + \zeta} \right)^{-1}. \quad (18)$$

PROOF. The details can be found in Appendix C. \square

LEMMA 4. Let $\mathbf{P}_M(x_1, y_1, z_1) \rightarrow (x_2, y_2, z_2) \in [0, 1]$ be the transition probability from (x_1, y_1, z_1) to (x_2, y_2, z_2) . \mathbf{P}_M can be obtained as follows:

For every $x_1, x_2, y_1 = 1, \dots, K$:

$$\mathbf{P}_M(x_1, y_1, 0) \rightarrow (x_2, y_1, 0) = \Omega_{x_1 x_2} \frac{\zeta}{\zeta + \lambda_C} \quad (19)$$

$$\mathbf{P}_M(x_1, y_1, 1) \rightarrow (x_2, y_1, 1) = \Omega_{x_1 x_2} \frac{\zeta}{\zeta + \mu} \quad (20)$$

$$\mathbf{P}_M(x_1, y_1, 2) \rightarrow (x_2, y_1, 2) = \Omega_{x_1 x_2} \frac{\zeta}{\zeta + \mu} \quad (21)$$

For every $x_1, y_1 = 1, \dots, K$:

$$\mathbf{P}_M(x_1, y_1, 0) \rightarrow (x_1, y_1, 1) = \frac{\lambda^*}{\zeta + \lambda_C} \quad (22)$$

$$\mathbf{P}_M(x_1, y_1, 0) \rightarrow (x_1, y_1, 2) = \frac{\lambda_C - \lambda^*}{\zeta + \lambda_C} \quad (23)$$

$$\mathbf{P}_M(x_1, y_1, 2) \rightarrow (x_1, y_1, 0) = \frac{\mu}{\zeta + \mu} \quad (24)$$

For every $x_1, y_1, y_2 = 1, \dots, K$:

$$\mathbf{P}_M(x_1, y_1, 1) \rightarrow (x_1, y_2, 0) = \frac{\mu}{\zeta + \mu} p_{y_2 x_1}^n \frac{\psi_{y_2}}{\psi_{x_1}} \quad (25)$$

Otherwise:

$$\mathbf{P}_M(x_1, y_1, z_1) \rightarrow (x_2, y_2, z_2) = 0. \quad (26)$$

PROOF. The details can be found in Appendix D. \square

Then, we obtain $\pi(x, y, z)$ using \mathbf{P}_M . Note that we cannot directly use $\pi(x, y, z)$ to calculate the error over the entire time span because it is an embedded Markov chain, not a continuous-time Markov chain [33]. That is, the state transitions in \mathbf{P}_M do not occur at every small time step; instead, they occur only with events of packet arrival, packet departure, and process state change, which makes the time spent in each (x, y, z) state at a single visit different from each other. Therefore, we need to incorporate the length of time spent in each (x, y, z) state. To that end, we let $w(x, y, z)$ denote the weight of state (x, y, z) that represents the weighted holding time at state (x, y, z) over the entire time span. In other words, $Ww(x, y, z)$ represents the expected time waited at state (x, y, z) during W state transitions. Note that, all these events, i.e., packet arrival, packet departure, and state change of the process, are memoryless, independent, and identically distributed, which preserves Markov property in embedded chains. Due to the memoryless property, only the occurrence time of the next event has an impact on the time spent in each state. With all the above in mind, we define a weighted holding time function $w(x, y, z) = \pi(x, y, z) \mathbb{E}[T_{(x, y, z)}]$ where $\mathbb{E}[T_{(x, y, z)}]$ is the expected holding time that represents the expected time waited until the next jump in state (x, y, z) at every transition and the holding time in state (x, y, z) is independent of $\pi(x, y, z)$. We present holding times in Lemma 5. $\mathbb{E}[T_{(x, y, z)}]$ values vary with different z values but changes in x or y do not affect $\mathbb{E}[T_{(x, y, z)}]$ because the possible events that can happen are the same for the same z . For example, the possible events are packet arrivals and state changes of the process if $z = 0$ for all $x, y = 1, \dots, K$.

LEMMA 5. Let $\mathbb{E}[T_{(x, y, z)}]$ be the expected holding time in state (x, y, z) . Then, $\mathbb{E}[T_{(x, y, z)}]$ can be calculated for every $x, y = 1, \dots, K$ as follows:

$$\mathbb{E}[T_{(x, y, 0)}] = \frac{1}{\zeta + \lambda_C}, \quad (27)$$

$$\mathbb{E}[T_{(x, y, 1)}] = \mathbb{E}[T_{(x, y, 2)}] = \frac{1}{\zeta + \mu}. \quad (28)$$

PROOF. The details can be found in Appendix E. \square

Then, the error ratio ϵ_j becomes the ratio of the sum of weighted holding times when x and y are not equal to the sum of all weighted holding times as follows:

$$\epsilon = \frac{\sum_{y \neq x} \sum_{z=0}^2 w(x, y, z)}{\sum_{x=1}^K \sum_{y=1}^K \sum_{z=0}^2 w(x, y, z)}. \quad (29)$$

6 AVERAGE AGE OPTIMIZATION

Correlation is an important factor in minimizing AoI, as demonstrated in equation (14) and decisions such as the placement of sensors can affect the correlation among them. It is obvious that a higher correlation leads to a smaller AoI. However, for example, increasing the correlation value of one sensor may decrease that of another due to the sensor's sensing ability. Therefore, it is crucial to distribute correlation probabilities to minimize AoI. In this section, we explore how to assign \mathbf{P}_C under certain constraints representing different scenarios to minimize the sum average AoI. Let Δ_{sum}

denote the sum AoI.

$$\Delta_{sum} = \sum_{j=1}^M \frac{\lambda_C}{\lambda_C + \mu} \left(\frac{\mu \mathbb{E}[Y^2]}{2} - \mu \mathbb{E}[Y] + \mathbb{E}[Y] + \frac{\mu \mathbb{E}[Y]}{\tilde{p}_j} \right). \quad (30)$$

To that end, the objective is to solve the following problem:

$$\begin{aligned} \min_{\mathbf{P}_C \in [0,1]^{N \times M}} \Delta_{sum} \\ \text{s.t. } h_i(\mathbf{P}_C) \leq 0, \forall i \in [N], \end{aligned}$$

where $h_i(\mathbf{P}_C)$ represents the i^{th} sensor's sensing ability constraint. Each sensor has its own constraint, which takes into account all correlation probabilities. The optimal solution has to meet all N sensor constraints. We keep $h_i(\mathbf{P}_C)$ general for now. However, later in this section, we define three different $h_i(\mathbf{P}_C)$ functions for representing different scenarios.

Given that the only parameters affected by variable \mathbf{P}_C in Δ_{sum} are \tilde{p}_j 's, we can remove the other parts to simplify the problem. Specifically, the optimal \mathbf{P}_C values are the same resulting from solving the problem below:

$$\begin{aligned} \min_{\mathbf{P}_C \in \mathbb{R}^{N \times M}} f(\mathbf{P}_C) = \sum_{j=1}^M \frac{1}{\tilde{p}_j} \\ \text{s.t. } h_i(\mathbf{P}_C) \leq 0, \forall i \in [N] \end{aligned} \quad (31)$$

$$0 \leq p_{ij}^c \leq 1, \quad i = 1, \dots, N; \quad j = 1, \dots, M,$$

where $[\tilde{p}_1 \quad \tilde{p}_2 \quad \dots \quad \tilde{p}_M] = \frac{\lambda^T \mathbf{P}_C}{\lambda_C}$ and constraint i represents a constraint for the sensor i . The convexity of objective function f is shown in Lemma 6.

LEMMA 6. *The objective function f is a convex function.*

PROOF. The details can be found in Appendix F. \square

We have a summation of ratios problem that is typically difficult to solve [16] when the problem is non-convex. If our objective function is convex, it doesn't necessarily mean that our problem is convex as well. This depends on whether the sensor constraints preserve convexity. If they do not, finding a solution may not be straightforward. Hence, we need to take into account the convexity of the problem while approaching it. When dealing with convex problems, the Karush-Kuhn-Tucker (KKT) conditions are sufficient for optimality [6]. To derive KKT conditions, we formulate Lagrange Function of the optimization problem as follows:

$$\begin{aligned} \mathcal{L}(\mathbf{P}_C, \boldsymbol{\tau}, \mathbf{v}, \boldsymbol{\xi}) = \sum_{j=1}^M \frac{\lambda_C}{\sum_{i=1}^N p_{ij}^c \lambda_i} + \sum_{i=1}^N \sum_{j=1}^M (p_{ij}^c - 1) \tau_{ij} \\ - \sum_{i=1}^N \sum_{j=1}^M p_{ij}^c v_{ij} + \sum_{i=1}^N h_i(\mathbf{P}_C) \xi_i. \end{aligned} \quad (32)$$

The KKT conditions for the optimization problem described in eq. (31) are as follows.

$$\tau_{ij}^* - v_{ij}^* - \frac{\lambda_C \lambda_i}{(\sum_{k=1}^N p_{kj}^c \lambda_k)^2} + \sum_{k=1}^N \xi_k^* \frac{d}{dp_{ij}^c} h_k(\mathbf{P}_C^*) = 0, \quad i = 1, \dots, N; \quad j = 1, \dots, M; \quad (33)$$

$$(p_{ij}^{c*} - 1) \tau_{ij}^* = 0, \quad i = 1, \dots, N; \quad j = 1, \dots, M; \quad (34)$$

$$p_{ij}^{c*} v_{ij}^* = 0, \quad i = 1, \dots, N; \quad j = 1, \dots, M; \quad (35)$$

$$h_i(\mathbf{P}_C^*) \xi_i^* = 0, \quad i = 1, \dots, N; \quad (36)$$

$$\boldsymbol{\tau}^*, \mathbf{v}^*, \boldsymbol{\xi}^* \geq 0, \quad (37)$$

$$1 \geq \mathbf{P}_C^* \geq 0, \quad (38)$$

$$h_i(\mathbf{P}_C^*) \geq 0 \quad i = 1, \dots, N. \quad (39)$$

In the sequel, we proceed under the assumption that the sensing ability constraint of each sensor operates independently of others and their respective assignments. This assumption is made to establish scenarios where each sensor's sensing ability is solely dependent on itself, without influence from other devices. Notably, $\text{row}_i(\mathbf{P}_C)$ denotes the probability assignments of sensor i across all processes. Consequently, $h_i(\mathbf{P}_C)$ exclusively comprises variables from $\text{row}_i(\mathbf{P}_C)$ when considering independent sensors. Hence, we can reformulate equation (33) for independent sensors as follows:

$$\tau_{ij}^* - v_{ij}^* - \frac{\lambda_C \lambda_j}{(\sum_{k=1}^N p_{kj}^c \lambda_k)^2} + \xi_i^* \frac{d}{dp_{ij}^c} h_i(\mathbf{P}_C^*) = 0 \quad i = 1, \dots, N \quad j = 1, \dots, M \quad (40)$$

If $p_{ij}^c \neq 1$, then we get that $\tau_{ij}^* = 0$; conversely, if $p_{ij}^c = 0$, then we get that $v_{ij}^* = 0$ in eq. (34) and (35). Therefore, at least one of τ_{ij}^* and v_{ij}^* is zero because p_{ij}^c can not be 0 and 1 at the same time. In addition, $\xi_i^* = 0$ means that $h_i(\mathbf{P}_C^*) \neq 0$, noting the eq. (36). In eq. (40), all τ_{ij}^* , v_{ij}^* and ξ_i^* can not be zero because $\frac{\lambda_C \lambda_j}{(\sum_{k=1}^N p_{kj}^c \lambda_k)^2}$ is non-zero in eq. (40). Therefore, if the sensing ability constraint for sensor i is not tight in a feasible set, ξ_i^* must be 0, which means p_{ij}^{c*} is either 0 or 1, otherwise all τ_{ij}^* , v_{ij}^* and ξ_i^* are 0. In other words, p_{ij}^{c*} is either 0 or 1 for all $j \in [M]$ if sensor i is unable to operate at its maximum sensing ability. We can also say that the objective function gets smaller with the increase in any p_{ij}^{c*} so that $p_{ij}^{c*} = 1$ when the sensing ability constraint is not tight. We can conclude that every sensor i uses its maximum sensing ability or $\text{row}_i(\mathbf{P}_C^*) = 1$ while the constraints are feasible. The optimal solution highly depends on the constraints, so we define three different constraints representing three different possible scenarios to analyze the minimum sum AoI. A sensor's performance can be impacted by the number of processes it tracks, and those scenarios represent the effects of the number of processes tracked by a sensor on the sensor's performance. The three effects we consider are: (1) no impact, (2) improved performance with more processes tracked, and (3) weakened performance with more processes tracked.

(1) **Linear Constraint Example:**

$$\mathbf{P_C} \mathbf{1} - \mathbf{b} \leq 0$$

This constraint indicates that the total probability remains constant even when probabilities are altered among different processes. If the sensor i is unable to consistently generate packets with updates, then $b_i < 1$. In addition, if there are some intersections between processes, b_i can be larger than 1. This sensor can be thought of as a camera that can track different areas and change its own position without any loss. Let us consider the case where b is smaller than 1. This reflects a scenario where a camera may suffer from malfunctioning, thus hindering it from gathering information about the process it monitors. On the other hand, if b is larger than 1, this translates to the cases where the areas have some intersections, which represent a correlation between processes, or the camera has the ability to sense more than one process at the same time. Example feasible sets are shown for a sensor with 2 processes in Figure 5.

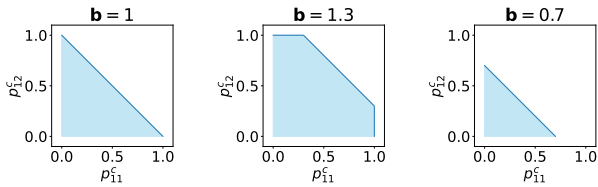


Figure 5: Example feasible set for a linear constraint for sensor 1 with 2 processes.

To minimize the sum of AoI, it is crucial to allocate probabilities optimally. This optimization process begins with obtaining $\sum_{i=0}^M \tilde{p}_i = \lambda^T \mathbf{P_C} \mathbf{1} / \lambda_C \leq \lambda^T \mathbf{b}$. Let us consider the case that we have a sensor that can track all processes at the same time, then the row i of the matrix $\mathbf{P_C}$ is equal to 1. The reason behind it can be having a powerful camera or high intersection rates between processes. Therefore, the sensor can track more processes than the system has, which concludes that M is less than b_i . This is because the maximum value of the i th row of $\mathbf{P_C} \mathbf{1}$ can be M . On the other hand, when the number of processes M is greater than or equal to the threshold b_i , the sensors use their maximum sensing ability to reach the minimum AoI. In other words, we can say that $\text{row}_i(\mathbf{P_C} \mathbf{1}) = \min(M, b_i)$. Hence, we conclude that $\sum_{i=0}^M \tilde{p}_j^* = \lambda^T \mathbf{P_C} \mathbf{1} / \lambda_C = \lambda^T (\min(\mathbf{M}, \mathbf{b}))$ to reach optimal sensing ability distribution. The objective presented in eq. (31) is equivalent to maximizing the harmonic mean. It is also known that the harmonic mean is always smaller than or equal to the arithmetic mean, and they are only equal when all values are identical. The arithmetic mean of the \tilde{p}_j^* is expressed as $\frac{1}{M} \sum_{j=0}^M \tilde{p}_j^*$, which is equivalent to $\frac{1}{M} \lambda^T (\min(\mathbf{M}, \mathbf{b}))$ and is a constant value. Therefore, we can conclude that the arithmetic mean is constant, and its value equals the maximum possible harmonic mean when all \tilde{p}_j^* are equal. Finally, the minimum AoI can be obtained by making $\tilde{p}_j^* = \lambda^T (\min(\mathbf{M}, \mathbf{b})) / M$ for all j , which can also be obtained from KKT conditions by letting $\frac{d}{dp_{ij}^*} h_k(\mathbf{P_C}^*) = 1$. This result implies that any $\mathbf{P_C}$ satisfies conditions in sensing ability constraints while all \tilde{p}_j are equal to the optimal solution, and there might be more than one optimal solution when $N > 1, M > 1$. One

possible solution is distributing sensing abilities equally such that $p_{ij}^{c*} = \min(M, b_i) / M, \forall i \in [N], j \in [M]$ and KKT conditions hold for this solution.

(2) **Quadratic Convex Constraint Example:**

$$\text{row}_i(\mathbf{P_C}) \text{row}_i(\mathbf{P_C})^T - b_i \leq 0 \quad i = 1, \dots, N$$

We set b_i values for intersections and sensor sensing abilities such that they can be less than or greater than 1 to represent the intersections and sensor sensing abilities, similar to the previous case. Then, the convex constraint states that the total probability may increase when probabilities are altered among different processes. To have a constraint that ensures the total probabilities increase when the sensor tracks more processes, let us consider a camera as a sensor that can track processes for different time periods. The camera can track a single process for a time period, and it can increase the probability of having information, but increasing the length of the time period may have a diminishing return. On the other hand, it is possible to split time and track two or more processes. While the probabilities of tracked processes decrease with an increase in the number of processes tracked, the sum of probabilities increases compared to tracking a single process. Example feasible sets for a sensor with 2 processes are illustrated in Figure 6.

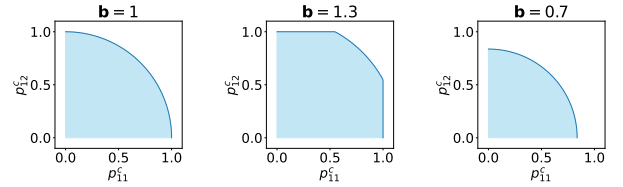


Figure 6: Example feasible set for a quadratic convex constraint for sensor 1 with 2 processes.

Similarly to the previous case, the sensor may have the ability to track more processes than M and it can track all processes but if unable, it uses all of its sensing ability to reach optimum, which can be written as $\text{row}_i(\mathbf{P_C}) \text{row}_i(\mathbf{P_C})^T = \min(M, b_i)$. If the sensor has the ability to track all processes ($b_i \geq M$) then the all probability values for the sensor is 1, ($p_{ij}^{c*} = 1, \forall j \in [M]$). For the sensors that can't track all processes at the same time ($b_i < M$), we use KKT conditions. Let us consider sensor i and assume $p_{ij}^{c*} \neq 1$ or $p_{ij}^{c*} \neq 0$ for all $j \in [M]$, we find that $v_{ij}^* = \tau_{ij}^* = 0$ for all $j \in [M]$. Additionally, the derivative of $h_k(\mathbf{P_C}^*)$ with respect to p_{ij}^* is $2p_{ij}^{c*}$. Substituting these values into equation (40), we obtain $\frac{\lambda_i}{\lambda_C \xi_i^*} = 2p_{ij}^{c*} \tilde{p}_j^{*2}$. When we keep i constant, we have the same left-hand side, so the optimal solution must satisfy these conditions. In this case, a potential solution could be to distribute the sensing ability of the sensor equally $p_{ij}^{c*} = \sqrt{\min(M, b_i) / M}, \forall i \in [N], j \in [M]$ and KKT conditions hold for this solution.

(3) **Quadratic Concave Constraint Example:**

$$b_i - \text{row}_i(\mathbf{1} - \mathbf{P_C}) \text{row}_i(\mathbf{1} - \mathbf{P_C})^T \leq 0 \quad i = 1, \dots, N$$

We assign b_i values to intersections and sensor sensing ability, which can be less than or greater than 1 to represent them, similar to previous cases. This constraint indicates that the overall probability may decrease when probabilities are changed across different

processes, rather than tracking a single process. Let us consider an example where a camera can track more than one process, but changing the camera's pose takes some time, during which it cannot generate updates with information. In this scenario, tracking more processes increases the total probability of losses, and the total probability is higher when the camera focuses only on a single process. However, if there is no update from the other processes, its AoI goes to infinity, which is also an undesired distribution. This trade-off decides the optimal distribution. Figure 7 illustrates the feasible sets for a sensor that has 2 processes.

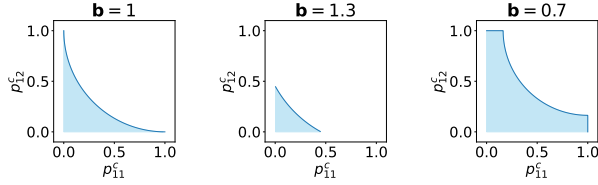


Figure 7: Example feasible set for a quadratic concave constraint for sensor 1 with 2 processes.

In this case, we end up with a non-convex set, unlike the previous two cases. As a result, our optimization problem is no longer convex. Although equal distribution satisfies KKT conditions, it may not be the global optimum. Therefore, non-convex optimization approaches, such as ADAM optimizer [20], can be applied in such cases. In the next section, first, we verify our theoretical analysis by presenting numerical results, and then we present optimal \mathbf{P}_C distributions for different scenarios.

7 NUMERICAL RESULTS

We categorize our analysis into two groups. First, we assume each sensor is assigned to a process, which means every sensor has a target process and every packet from this sensor contains information about the target process. In this case, the packet may also contain information about other sources. We vary different system parameters to verify our theoretical results in Section 4 and Section 5. After verifying the theoretical analysis, we investigate the trade-off model described in Section 6 such that each sensor has some sensing abilities and constraints. Considering constraints, the sensors can distribute their sensing abilities over processes and we investigate the best \mathbf{P}_C distribution to get the minimum sum AoI.

7.1 Sensor-Process Assignment Analysis

In this section, we provide numerical results for the average AoI and error ratio for a process in a system with two processes with two states and two sensors to see the effects of correlation probability. Our simulations are unit-time-based and were run for 1 million units of time. The lowest arrival rate is 0.5 arrivals per unit time. This means we have at least 5×10^5 arrivals for each process to guarantee convergence. Simulation results are shown as circles, and theoretical expressions from our analysis are shown as solid lines in the figures. The strong match between them verifies the validity of our analysis. We have three figures that depict different relationships. The first two show the effect of correlation on AoI and error with varying arrival rates. The third one shows how the

error changes with the state change rate for different correlations. We set Ω_1 , Ω_2 and \mathbf{P}_C as follows for the three cases.

$$\Omega_1 = \Omega_2 = \begin{bmatrix} 0.4 & 0.6 \\ 0.3 & 0.7 \end{bmatrix}, \quad \mathbf{P}_C = \begin{bmatrix} 1 & 1-p \\ 1-p & 1 \end{bmatrix},$$

where p is variable to vary p_{21}^c and p_{12}^c . As a result of our analysis, only $col_j(\mathbf{P}_C)$ has an effect on metrics for process j . Therefore, we only focus on $col_1(\mathbf{P}_C)$ in this part.

We start with investigating the effects of correlation on AoI in Figure 8. We vary the correlation probability p_{21}^c and λ_1 with a service rate μ fixed to 4, state change rate for process 1 and process 2, ζ_1 and ζ_2 fixed to 4, the arrival rate for sensor 2, λ_2 is fixed to 8. As p_{21}^c increases, AoI decreases as expected. However, the impact of correlation gets smaller with the increase in sensor 1's arrival rate λ_1 . The reason is that an increase in λ_1 causes a diminishing AoI drop, and correlation has a small impact when the λ_1 is high enough. As λ_1 increases, the status updates become more frequent, but there is another limitation on AoI, which is service time. Therefore, the AoI converges while λ_i is increasing. However, if λ_1 is small, and there is another sensor with a high arrival rate λ_2 , correlation plays a huge role in AoI.

Next, we observe the error ratio for the same system in Figure 9. We vary the correlation probability p_{21}^c and λ_1 with a service rate μ fixed to 4, state change rate for process 1 and process 2, ζ_1 and ζ_2 fixed to 4, arrival rate for sensor 2, λ_2 fixed to 8. We see a very similar pattern to Figure 8 in Figure 9. As p_{21}^c increases, the error drops as expected because of an increase in the ratio of informative packets in the server. Correlation has a powerful effect on error ratio, unlike AoI for $\lambda_i = 8$. As p_{21}^c , the difference between errors of different λ_i values decreases because all packets become informative. However, an increase in the arrival rate causes convergence and it cannot cause an error-less system because both μ and ζ_1 are 4. State changes during service time cause errors and the error causing them can not be reducible by neither increasing arrival rate nor increasing correlation.

Lastly, we investigate the system with a service rate μ fixed to 4, arrival rate for sensor 1, λ_1 fixed to 2, arrival rate for sensor 2, λ_2 fixed to 8. We vary the correlation probability p_{21}^c and ζ_1 . We only show the error ratio for this system in Figure 10 because changes in ζ_1 do not have any effect on AoI. With the increase in ζ_1 , the error is increasing as expected because the state changes more frequently. For small ζ_1 , the error is almost zero for all correlation values. The error ratio converges to $\langle \psi, \psi \rangle$, inner product of stationary distribution ψ defined in (3), as ζ_1 goes to ∞ because λ_1 and λ_2 become meaningless when ζ_1 is too large. Although the start and end points of all plots for error are pretty similar, when the correlation is high, the increase becomes slower, which makes the system more robust.

7.2 Sensing Ability-Constrained Sensor Optimization Model

To evaluate optimal assignment, we consider a system such that it has two sensors and two processes. We set $\lambda_1 = 1$, $\mu = 4$, $\mathbf{b} = 1$ and λ_2 as variable. Then, we use the SLSQP algorithm [21] for convex problems to converge to the global optimum. For the non-convex part, we use grid search to find the distribution that provides the global minimum AoI to avoid converging a local minimum. We use a step size of 10^{-3} for each probability variable. Our numerical

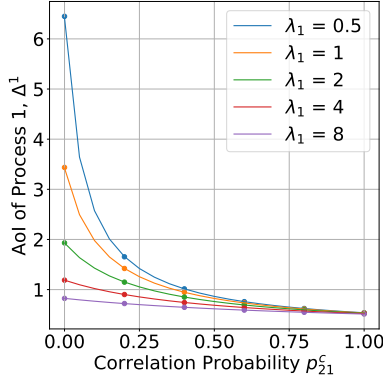


Figure 8: AoI versus p_{21}^c for different λ_i values with $\mu = 4, \zeta_1 = 4, \zeta_2 = 4, \lambda_2 = 8$.

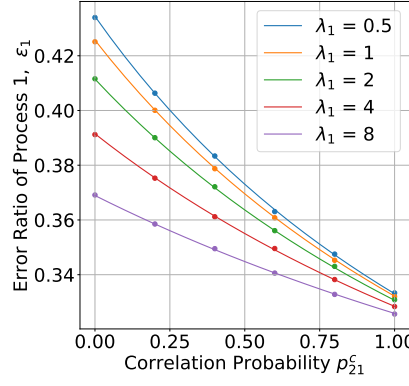


Figure 9: Error ϵ_1 versus p_{21}^c for different λ_i values with $\mu = 4, \zeta_1 = 4, \zeta_2 = 4, \lambda_2 = 8$.

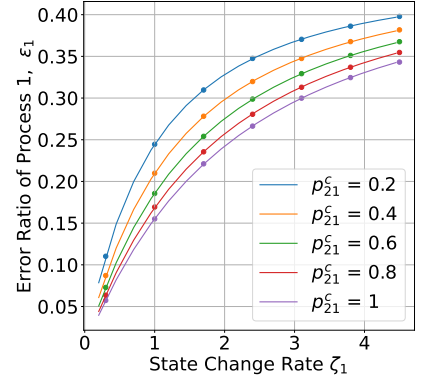
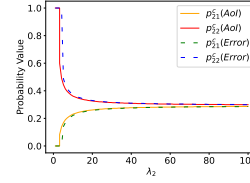


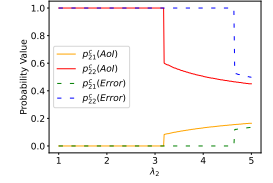
Figure 10: Error ϵ_1 versus ζ_1 for different p_{21}^c values with $\mu = 4, \zeta_2 = 4, \lambda_1 = 2, \lambda_2 = 8$.

findings are presented for two sensors and two process systems. We use three constraints defined in Section 6, and we obtain the constraints are always tight when $\mathbf{b} \leq M$ and sensors use their maximum sensing abilities, which verifies our argument derived from KKT conditions. The other result we see is equal distribution is optimal for the first two constraints. After verifying the first two constraints, we evaluate the third constraint that makes the problem non-convex. The results are shown in Figure 11. We show only p_{21}^c and p_{22}^c because we got $p_{11}^c = 1$ and $p_{12}^c = 0$. Note that switching p_{11}^c and p_{21}^c to p_{12}^c and p_{22}^c , respectively, results in the same AoI value due to the symmetry of the case. As a result, we observe equal sensing probabilities are not always the best distribution, in contrast to the previous cases. When sensor λ 's are close to each other, assigning a sensor to a process provides the minimum AoI. However, if one sensor dominates the other in terms of arrival rate, the equal distribution provides the optimal AoI for the dominating sensor. Interestingly, we observe that the value of p_{21}^c does not change smoothly from 1. We obtain $p_{21}^c = 0$ if $\lambda_2 \lesssim 3.18$ but we see a dramatic regime switch to 0.08 after around 3.18. The reason behind this is tracking more than one process cause a loss of total sensing ability. Therefore, if λ 's are close to each other, there is a tendency to track a single process from each sensor. However, if one λ is significantly larger than the other, the system will converge to the configuration with a single sensor. In order to avoid infinite AoI values, both processes must be tracked by the dominating sensor. In addition to AoI minimization, we also consider error minimization in our. We see a similar trend in error minimization, too. Until a threshold that is different from AoI threshold, optimal assignment is assigning a process to the sensor. After the threshold, it converges to equal distribution while λ_2 is increasing. Note that each process's error value has an upper bound, unlike AoI. Therefore, we have a larger regime change threshold for error than for AoI.

It is easier to find the optimal assignment when the constraints are convex, considering all possible cases. To find the optimal assignments, we can use KKT conditions or apply Newton's algorithm. However, finding a global optimum requires the use of an optimizer or grid search algorithm when the problem is non-convex.



(a) λ_2 versus optimal p_{21}^c and p_{22}^c value with $\lambda_1 = 1, \mu = 4, \mathbf{b} = 1$.



(b) A closer look at the regime switch in Figure a.

Figure 11: Optimal distribution for quadratic non-convex distribution example.

8 CONCLUSION

Our paper investigated a multi-user system with a single server that has correlated information. Each process's state changes randomly, and a monitor tracks their status updates through sensors such that status update packets generated by a sensor may contain details about other processes. This introduces the concept of correlation. The study focused on understanding the effects of correlation on the system, particularly in terms of the age of information and the error ratio of the monitor's estimation. We have presented analytical expressions and conducted comparisons to evaluate the influence of correlation across different system parameters. Our research delves into how sensors with restricted sensing abilities allocate resources across processes. We have discovered intricate links between correlation, process dynamics, and optimization tactics for managing the age of information. Interestingly, we have found cases where monitoring multiple processes from one source may not always be advantageous, and optimal resource distribution for different arrival rates might fluctuate unexpectedly. Through this exploration, the study sheds light on the complicated dynamics of correlated multi-state multi-user systems and provides insight into how correlation affects the age of information and estimation accuracy.

ACKNOWLEDGMENT

This work was supported in part by NSF grant CNS-2219180.

REFERENCES

- [1] Ahmed Arafa, Jing Yang, Sennur Ulukus, and H. Vincent Poor. 2020. Age-Minimal Transmission for Energy Harvesting Sensors With Finite Batteries: Online Policies. *IEEE Transactions on Information Theory* 66, 1 (2020), 534–556. <https://doi.org/10.1109/TIT.2019.2938969>
- [2] Baran Tan Bacinoglu, Elif Tugce Ceran, and Elif Uysal-Biyikoglu. 2015. Age of information under energy replenishment constraints. In *2015 Information Theory and Applications Workshop (ITA)*. 25–31. <https://doi.org/10.1109/ITA.2015.7308962>
- [3] Abdulrahman Baknina, Omur Ozel, Jing Yang, Sennur Ulukus, and Aylin Yener. 2018. Sending Information Through Status Updates. In *2018 IEEE International Symposium on Information Theory (ISIT)*. 2271–2275. <https://doi.org/10.1109/ISIT.2018.8437496>
- [4] Ahmed M. Bedewy, Yin Sun, and Ness B. Shroff. 2016. Optimizing data freshness, throughput, and delay in multi-server information-update systems. In *2016 IEEE International Symposium on Information Theory (ISIT)*. 2569–2573. <https://doi.org/10.1109/ISIT.2016.7541763>
- [5] Dimitri Bertsekas and Robert Gallager. 1996. *Data Networks* (second ed.). Prentice Hall.
- [6] Stephen Boyd and Lieven Vandenbergh. 2004. *Convex optimization*. Cambridge university press.
- [7] George P.H. Styan C.C. Paige and Peter G. Wachter †. 1975. Computation of the stationary distribution of a markov chain. *Journal of Statistical Computation and Simulation* 4, 3 (1975), 173–186. <https://doi.org/10.1080/00949657508810122> arXiv:https://doi.org/10.1080/00949657508810122
- [8] Maice Costa, Marian Codreanu, and Anthony Ephremides. 2016. On the Age of Information in Status Update Systems With Packet Management. *IEEE Transactions on Information Theory* 62, 4 (2016), 1897–1910. <https://doi.org/10.1109/TIT.2016.2533395>
- [9] Shahab Farazi, Andrew G. Klein, and D. Richard Brown. 2018. Average age of information for status update systems with an energy harvesting server. In *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. 112–117. <https://doi.org/10.1109/INFOCOMW.2018.8406846>
- [10] Robert G. Gallager. 1996. *Poisson Processes*. Springer US, Boston, MA, 31–55. https://doi.org/10.1007/978-1-4615-2329-1_2
- [11] Qing He, Gyorgy Dan, and Viktoria Fodor. 2018. Minimizing age of correlated information for wireless camera networks. In *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. 547–552. <https://doi.org/10.1109/INFOCOMW.2018.8406914>
- [12] Shibo He, Kun Shi, Chen Liu, Bicheng Guo, Jiming Chen, and Zhiguo Shi. 2022. Collaborative Sensing in Internet of Things: A Comprehensive Survey. *IEEE Communications Surveys & Tutorials* 24, 3 (2022), 1435–1474. <https://doi.org/10.1109/COMST.2022.3187138>
- [13] Yu-Pin Hsu, Eytan Modiano, and Lingjie Duan. 2017. Age of information: Design and analysis of optimal scheduling algorithms. In *2017 IEEE International Symposium on Information Theory (ISIT)*. 561–565. <https://doi.org/10.1109/ISIT.2017.8006590>
- [14] Longbo Huang and Eytan Modiano. 2015. Optimizing age-of-information in a multi-class queueing system. In *2015 IEEE International Symposium on Information Theory (ISIT)*. 1681–1685. <https://doi.org/10.1109/ISIT.2015.7282742>
- [15] Yoshiaki Inoue, Hiroyuki Masuyama, Tetsuya Takine, and Toshiyuki Tanaka. 2019. A General Formula for the Stationary Distribution of the Age of Information and Its Application to Single-Server Queues. *IEEE Trans. Inf. Theor.* 65, 12 (dec 2019), 8305–8324. <https://doi.org/10.1109/TIT.2019.2938171>
- [16] Yunchol Jong. 2012. Practical Global Optimization Algorithm for the Sum-of-Ratios Problem. arXiv:1207.1153 [math.OC]
- [17] Igor Kadota, Abhishek Sinha, Elif Uysal-Biyikoglu, Rahul Singh, and Eytan Modiano. 2018. Scheduling Policies for Minimizing Age of Information in Broadcast Wireless Networks. *IEEE/ACM Transactions on Networking* 26, 6 (2018), 2637–2650. <https://doi.org/10.1109/TNET.2018.2873606>
- [18] Anders E. Kaler and Petar Popovski. 2019. Minimizing the Age of Information From Sensors With Common Observations. *IEEE Wireless Communications Letters* 8, 5 (2019), 1390–1393. <https://doi.org/10.1109/LWC.2019.2919528>
- [19] Sanjit Kaul, Roy Yates, and Marco Gruteser. 2012. Real-time status: How often should one update?. In *2012 Proceedings IEEE INFOCOM*. 2731–2735. <https://doi.org/10.1109/INFOCOM.2012.6195689>
- [20] Diederik P. Kingma and Jimmy Ba. 2017. Adam: A Method for Stochastic Optimization. arXiv:1412.6980 [cs.LG]
- [21] D. Kraft. 1988. *A Software Package for Sequential Quadratic Programming*. Wiss. Berichtsweesen d. DFVLR. <https://books.google.com/books?id=4rKaGwAACAAJ>
- [22] Chengzhang Li, Qingyu Liu, Shaoran Li, Yongce Chen, Y. Thomas Hou, Wenjing Lou, and Sastry Kompella. 2022. Scheduling With Age of Information Guarantee. *IEEE/ACM Transactions on Networking* 30, 5 (2022), 2046–2059. <https://doi.org/10.1109/TNET.2022.3156866>
- [23] Ali Maatouk, Mohamad Assaad, and Anthony Ephremides. 2019. Minimizing The Age of Information: NOMA or OMA?. In *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. 102–108. <https://doi.org/10.1109/INFOCOMW.2019.8845254>
- [24] Ali Maatouk, Mohamad Assaad, and Anthony Ephremides. 2020. On the Age of Information in a CSMA Environment. *IEEE/ACM Transactions on Networking* 28, 2 (2020), 818–831. <https://doi.org/10.1109/TNET.2020.2971350>
- [25] Ali Maatouk, Saad Kriouile, Mohamad Assaad, and Anthony Ephremides. 2020. The Age of Incorrect Information: A New Performance Metric for Status Updates. *IEEE/ACM Transactions on Networking* 28, 5 (2020), 2215–2228. <https://doi.org/10.1109/TNET.2020.3005549>
- [26] Elie Najm, Rajai Nasser, and Emre Telatar. 2018. Content Based Status Updates. In *2018 IEEE International Symposium on Information Theory (ISIT)*. 2266–2270. <https://doi.org/10.1109/ISIT.2018.8437577>
- [27] Jiayu Pan, Ahmed M. Bedewy, Yin Sun, and Ness B. Shroff. 2021. Minimizing Age of Information via Scheduling over Heterogeneous Channels. In *Proceedings of the Twenty-Second International Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing (Shanghai, China) (MobiHoc '21)*. Association for Computing Machinery, New York, NY, USA, 111–120. <https://doi.org/10.1145/3466772.3467040>
- [28] Parisa Rafiee, Zhuoxuan Ju, and Miloš Doroslovački. 2024. Adaptive ON/OFF Scheduling to Minimize Age of Information in an Energy-Harvesting Receiver. *IEEE Sensors Journal* 24, 3 (2024), 3898–3911. <https://doi.org/10.1109/JSEN.2023.3339598>
- [29] C. S. Raghavendra, Krishna M. Sivalingam, and Taieb Znati (Eds.). 2004. *Wireless sensor networks*. Kluwer Academic Publishers, USA.
- [30] R Vallabh Ramakanth, Vishrant Tripathi, and Eytan Modiano. 2023. Monitoring Correlated Sources: AoI-based Scheduling is Nearly Optimal. arXiv:2312.16813 [cs.NI]
- [31] Alkan Soysal and Sennur Ulukus. 2021. Age of Information in G/G/1 Systems: Age Expressions, Bounds, Special Cases, and Optimization. *IEEE Trans. Inf. Theor.* 67, 11 (nov 2021), 7477–7489. <https://doi.org/10.1109/TIT.2021.3095823>
- [32] Yin Sun, Elif Uysal-Biyikoglu, Roy Yates, C. Emre Koksal, and Ness B. Shroff. 2016. Update or wait: How to keep your data fresh. In *IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications*. 1–9. <https://doi.org/10.1109/INFOCOM.2016.7524524>
- [33] Howard M. Taylor and Samuel Karlin. 2011. *An Introduction To Stochastic Modeling* (fourth edition ed.). Academic Press.
- [34] Jingwen Tong, Liqun Fu, and Zhu Han. 2022. Age-of-Information Oriented Scheduling for Multichannel IoT Systems With Correlated Sources. *IEEE Transactions on Wireless Communications* 21, 11 (2022), 9775–9790. <https://doi.org/10.1109/TWC.2022.3179305>
- [35] Vishrant Tripathi and Eytan Modiano. 2022. Optimizing age of information with correlated sources. In *Proceedings of the Twenty-Third International Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing (Seoul, Republic of Korea) (MobiHoc '22)*. Association for Computing Machinery, New York, NY, USA, 41–50. <https://doi.org/10.1145/3492866.3549719>
- [36] Roy D. Yates and Sanjit K. Kaul. 2019. The Age of Information: Real-Time Status Updating by Multiple Sources. *IEEE Transactions on Information Theory* 65, 3 (2019), 1807–1827. <https://doi.org/10.1109/TIT.2018.2871079>
- [37] Peng Zou, Ali Maatouk, Jin Zhang, and Suresh Subramaniam. 2023. How Costly Was That (In)Decision?. In *2023 21st International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt)*. 278–285. <https://doi.org/10.23919/WiOpt58741.2023.10349833>

A PROOF OF LEMMA 1

To prove our lemma, we apply the splitting property of the Poisson process. Let $N(t)$ be a Poisson process with rate parameter λ . If events are split into two groups with probabilities p and $1 - p$, then the resulting processes $N_1(t)$ and $N_2(t)$ are independent Poisson processes with rate parameters $p\lambda$ and $(1 - p)\lambda$ respectively[10].

From process j perspective, we can split arrivals from sensor i into two groups as informative and uninformative arrivals with probabilities p_{ij}^c and $1 - p_{ij}^c$. Rate of arrivals from sensor i is λ_i so rate of informative arrivals for process j from sensor i is $p_{ij}^c \lambda_i$. Then, we can merge all informative arrivals as a single process because all of them are Poisson. Total arrival rate of packets with information of process j 's state λ_j^* equals to

$$\lambda_j^* = \sum_{i=1}^N p_{ij}^c \lambda_i \quad (41)$$

As a vector,

$$\lambda^{*T} = [\lambda_1^* \quad \lambda_2^* \quad \dots \quad \lambda_M^*] = \lambda^T P_C \quad (42)$$

The importance of the packet is whether it has information of process j so we can say that The system with N sensors and arrival rates λ shown in Figure 1 equivalents to the system with two sources and arrival rates of λ_j^* and $\lambda_C - \lambda_j^*$ shown in Figure 3 from process j 's perspective

B PROOF OF THEOREM 1

We need to find $\mathbb{E}[\tilde{Y}_j]$ and $\mathbb{E}[\tilde{Y}_j^2]$. First, let A be a set defined as a set that consists of indices corresponding to arrivals between the $(l-1)$ -th and l -th informative arrivals. Note that l -th informative arrival may not correspond to l -th arrival if there is any uninformative arrival before it. So, Let the l -th informative arrival correspond to the k -th arrival, and let the $(l-1)$ -th informative arrival correspond to the $(k-r)$ -th arrival. A can be rewritten as,

$$A = \{a \mid k-r < a \leq k \text{ for } r \geq 1 \text{ and } k \geq r\} \quad (43)$$

where $P(r) = (1 - \tilde{p}_j)^{r-1} \tilde{p}_j$ for $r \geq 1$. From [8], we also know that

$$\mathbb{E}[Y] = \frac{1}{\lambda_C} + \frac{1}{\mu} \quad (44)$$

$$\mathbb{E}[Y^2] = \frac{2(\lambda_C^2 + \lambda_C \mu + \mu^2)}{\lambda_C^2 \mu^2} \quad (45)$$

When we express \tilde{Y}_j^l in terms of Y_a , we get

$$\tilde{Y}_j^l = \sum_{a \in A} Y_a \quad (46)$$

$$\begin{aligned} \mathbb{E}[\tilde{Y}_j] &= \sum_{r=1}^{\infty} P(R=r) r \mathbb{E}[Y] \\ &= \mathbb{E}[Y] \sum_{r=1}^{\infty} (1 - \tilde{p}_j)^{r-1} \tilde{p}_j r \\ &= \mathbb{E}[Y] \mathbb{E}[R] \\ &= \mathbb{E}[Y] / \tilde{p}_j \end{aligned} \quad (47)$$

Similarly,

$$\mathbb{E}[\tilde{Y}_j^2 | r] = \mathbb{E}[(\sum_{a \in A} Y_a)^2] = r \mathbb{E}[Y^2] + 2 \binom{r}{2} \mathbb{E}[Y]^2 \quad (48)$$

$$\begin{aligned} \mathbb{E}[\tilde{Y}_j^2] &= \sum_{r=1}^{\infty} \tilde{p}_j (1 - \tilde{p}_j)^{r-1} \mathbb{E}[\tilde{Y}_j^2 | R=r] \\ &= \frac{\mathbb{E}[Y^2]}{\tilde{p}_j} + \frac{2\mathbb{E}[Y]^2 (1 - \tilde{p}_j)}{\tilde{p}_j^2} \end{aligned} \quad (49)$$

After that, we replace $\mathbb{E}[\tilde{Y}_j]$ and $\mathbb{E}[\tilde{Y}_j^2]$ to findings in equation (12) to get

$$\Delta_i = \frac{\lambda_C}{\lambda_C + \mu} \left(\frac{\mu \mathbb{E}[Y^2]}{2} + \frac{\mu \mathbb{E}[Y] (1 - \tilde{p}_j)}{\tilde{p}_j} + \mathbb{E}[Y] \right) \quad (50)$$

C PROOF OF LEMMA 3

Both state change and service time events are exponential random variables, so they are memoryless. Using the memoryless property, the probability distribution function of the remaining service time t is

$$f_T(t|\mu) = \mu e^{-t\mu} \quad (51)$$

The number of state changes k over time t is a Poisson process is

$$P(k|t) = \frac{e^{-(\zeta t)} (\zeta t)^k}{k!} \quad (52)$$

Next, we determine the probabilities of transitioning from the initial state i to the final state j after k transitions by evaluating $(P^s)_{ij}^k$. After that, we get the matrix consisting of transitioning probabilities from state i to state j when service time is t and the number of state changes k as

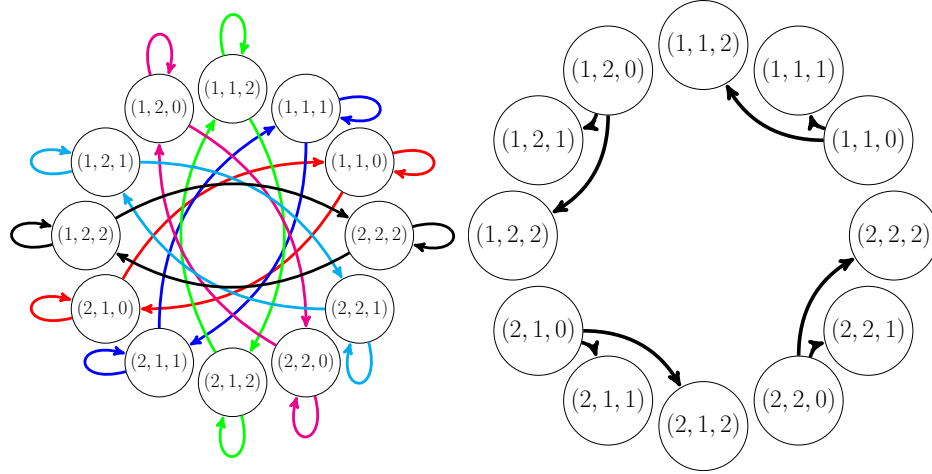
$$P_N(k, t) = \mu e^{-\mu t} \frac{e^{-(t\zeta)} (t\zeta)^k}{k!} \Omega^k dt \quad (53)$$

Then, we obtain

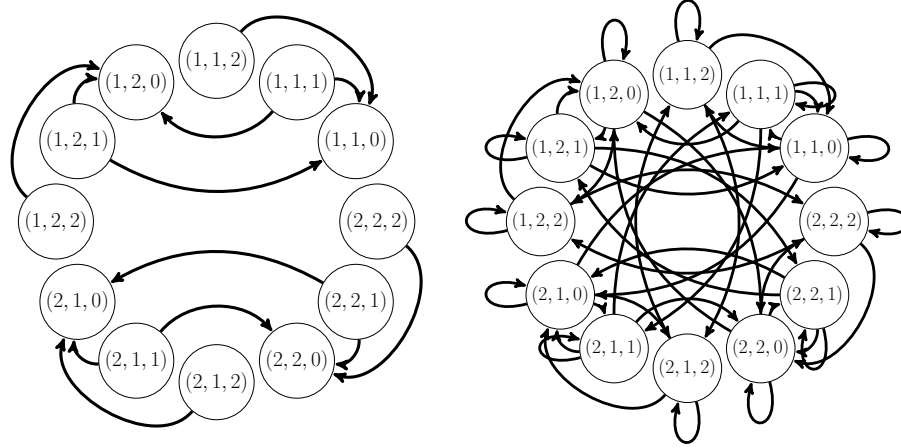
$$\begin{aligned} P_N &= \int_0^{\infty} \mu e^{-\mu t} \sum_{k=0}^{\infty} \frac{e^{-(t\zeta)} (t\zeta)^k}{k!} \Omega^k dt \\ &= \sum_{k=0}^{\infty} \int_0^{\infty} \mu e^{-\mu t} \frac{e^{-(t\zeta)} (t\zeta)^k}{k!} \Omega^k dt \\ &= \sum_{k=0}^{\infty} \frac{\Omega^k}{k!} \int_0^{\infty} \mu e^{-\mu t} e^{-(t\zeta)} (t\zeta)^k dt \\ &= \sum_{k=0}^{\infty} \frac{\Omega^k}{k!} \Gamma(k+1, 0) \zeta^k \mu \cdot (\mu + \zeta)^{-k-1} \\ &\quad \text{where } \Gamma(k+1, 0) = k! \\ &= \sum_{k=0}^{\infty} \Omega^k \zeta^k \mu \cdot (\mu + \zeta)^{-k-1} \\ &= \frac{\mu}{\mu + \zeta} \sum_{k=0}^{\infty} \left(\frac{\zeta \Omega}{\mu + \zeta} \right)^k \\ &= \frac{\mu}{\mu + \zeta} \sum_{k=0}^{\infty} \left(\frac{\zeta \Omega}{\mu + \zeta} \right)^k \\ &= \frac{\mu}{\mu + \zeta} \left(I - \frac{\zeta \Omega}{\mu + \zeta} \right)^{-1} \end{aligned} \quad (54)$$

D PROOF OF LEMMA 4

To understand transitions and their probabilities, first, we look at events that may occur and cause a state transition. There are three events in the system: packet arrival, packet departure, and state change for the given process. If the server is idle and an event occurs, the event may be packet arrival or state change for the given process with probabilities $\frac{\lambda_C}{\zeta + \lambda_C}$ or $\frac{\zeta}{\zeta + \lambda_C}$, respectively but it can not be packet departure. The arrivals may be informative and change the information the monitor has or not with probabilities



(a) Possible transitions when a state change occurs. (b) Possible transitions when a packet arrives.



(c) Possible transitions when a packet departs. (d) All transitions together.

Figure 12: Transitions of P_M .

$\tilde{p}_i = \frac{\lambda_i^*}{\lambda_C}$ or $(1 - \tilde{p}_i) = \frac{\lambda_C - \lambda_i^*}{\lambda_C}$, respectively. Conversely, if the server is busy with an informative or uninformative packet and an event occurs, the event may be packet departure or state change for the given process with probabilities $\frac{\mu}{\xi + \mu}$ or $\frac{\xi}{\xi + \mu}$, respectively but it can not be packet arrival. Then, we obtain 7 different possibilities that cause transitions. They are:

- (1) **Server is idle, and state change for the given process happens.**
- (2) **Server is busy with an informative packet, and state change for the given process happens.**
- (3) **Server is uninformative with an informative packet, and state change for the given process happens.**
- (4) **Server is idle, and an informative packet arrives.**
- (5) **Server is idle, and an uninformative packet arrives.**
- (6) **Server is busy with an informative packet, and the packet departs.**

- (7) **Server is uninformative with an informative packet, and the packet departs.**

To understand transitions better, we illustrate a Markov chain for states (x, y, z) when $K = 2$ in Figure 12. The transitions caused by the process' state change are shown in Figure (12a). Change in the process' state may affect only the current state x value and the state information the monitor has y , server state z stays constant when the process' state changes. In Figure (12b), the transitions caused by packet arrivals are shown. Packet arrival has an effect on only server state z , and z goes from 0 to 1 if the packet is informative. Otherwise, z goes from 0 to 2. The transitions caused by packet departures are shown in Figure (12c). If the packet is uninformative, z goes to 0 from 2 because the server becomes idle and x, y stays constant. However, if the packet is informative, the packet has the information when it arrives and it updates y while departing. So, z goes to 0 from 1, and y goes to the value the packet has, which can be all possible values with some probability. Lastly, we show all transitions in Figure (12d).

Then we calculate each transition probability to form \mathbf{P}_M matrix. First, state change does not affect the information the monitor has and the server's state, which are y and z , but it causes a transition for x from initial x_1 to x_2 with probability $\Omega_{x_1 x_2}$ for every $x_1, x_2 = 1, \dots, K$. After having those probabilities, we get equations (19), (20), and (21) as multiplication of $\Omega_{x_1 x_2}$ and probability of having state change before packet arrival or departure.

Packet arrival does not affect the current state of the process and the information the monitor has, which are x and y , but it causes a transition for z from initial 0 to 1 or 2 with probabilities $\frac{\lambda_i^*}{\lambda_C}$ and $\frac{\lambda_C - \lambda_i^*}{\lambda_C}$, respectively. Then, we get the equations (22) and (23), as multiplication of the probability of being informative or uninformative and probability of having packet arrival before state change.

Uninformative packet departure does not affect the current state of the process x and the information the monitor has y , but it causes a transition for z from initial 2 to 0. Therefore, we get equation (24) as the probability of having packet departure before state change.

Informative packet departure does not affect the current state of the process x , but it causes a transition for (y, z) from initial $(y_1, 1)$ to $(y_2, 0)$ with some probability. To find this, we define two events, X_1 and X_2 , as the state of the process when a packet arrives and the state of the process when the packet departs. Then, the probability of transition from $(x_1, y_1, 1)$ to $(x_1, y_2, 0)$ if packet departure happens before state change of process is $P[X_1 = y_2 | X_2 = x_1]$ because the packet has the information of state when it arrives. We calculated $P[X_2 = x_2 | X_1 = y_2] = p_{y_2 x_1}^n$ in Lemma 3 which is the probability of the process' state is x_2 after the packet is served when the process' state is y_2 initially. Using Bayes' rule, we obtain:

$$P[X_1 = y_2 | X_2 = x_1] = \frac{P[X_2 = x_1 | X_1 = y_2] \cdot P[X_1 = y_2]}{P[X_2 = x_1]} \quad (55)$$

where $P[X_2 = x_1] = \psi_{x_1}$ and $P[X_1 = y_2] = \psi_{y_2}$ in equation (3). After getting $P[X_1 = y_2 | X_2 = x_1]$ as $p_{y_2 x_1}^n \frac{\psi_{y_2}}{\psi_{x_1}}$, we multiply it by the probability of packet departure before state change of process to get equation (25).

E PROOF OF LEMMA 5

Time is spent in a state at every visit until an event happens. All three events have exponential distribution between two occurrences, and they are Poisson processes. Let the event of the process' state change, the event of arrival, and the event of departure be $N_S(t)$ with rate parameter ζ , $N_A(t)$ with rate parameter λ_C , $N_D(t)$ with rate parameter μ , respectively. Using the merging property of the Poisson process, we obtain the event occurrence process for each state. The event of the process' state change can happen in all states. The event of arrival happens only when the server is idle, which is $z = 0$. The event of departure happens only when the server is busy, which is $z = 1$ and $z = 2$.

For any x and y when $z = 0$, the possible events are the event of arrival or the event of the process' state change, so the event occurrence process becomes $N_S(t) + N_A(t)$, which is a Poisson process with rate $\zeta + \lambda_C$ and the expected interarrival time is equal to $\mathbb{E}[T_{(x_1, y_1, 0)}]$. Thus, For every $x_1, y_1 = 1, \dots, K$:

$$\mathbb{E}[T_{(x_1, y_1, 0)}] = \frac{1}{\zeta + \lambda_C} \quad (56)$$

For any x and y when $z = 1$ or $z = 2$, the possible events are the event of departure or the event of the process' state change, so the event occurrence process becomes $N_D(t) + N_A(t)$, which is a Poisson process with rate $\mu + \lambda_C$ and the expected interarrival time is equal to $\mathbb{E}[T_{(x_1, y_1, 1)}]$ or $\mathbb{E}[T_{(x_1, y_1, 2)}]$. Thus, For every $x_1, y_1 = 1, \dots, K$:

$$\mathbb{E}[T_{(x_1, y_1, 0)}] = \frac{1}{\zeta + \lambda_C} \quad (57)$$

$$\mathbb{E}[T_{(x_1, y_1, 1)}] = \mathbb{E}[T_{(x_1, y_1, 2)}] = \frac{1}{\zeta + \mu} \quad (58)$$

F PROOF OF LEMMA 6

As a first step, we aim to prove that the function $g(\tilde{\mathbf{p}})$ defined as $g(\tilde{\mathbf{p}}) = \frac{1}{\tilde{p}_1} + \frac{1}{\tilde{p}_2} + \dots + \frac{1}{\tilde{p}_n}$ is convex using its Hessian matrix. The Hessian matrix of g is given by:

$$H_g = \begin{bmatrix} \frac{2}{\tilde{p}_1^3} & 0 & \dots & 0 \\ 0 & \frac{2}{\tilde{p}_2^3} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \frac{2}{\tilde{p}_n^3} \end{bmatrix}$$

All the diagonal elements of the Hessian matrix are positive when \tilde{p}_i is positive for all i . We know that \tilde{p}_i is formed using probabilities and positive arrival rates so all the diagonal elements of the matrix are positive. In addition, all the off-diagonal elements are zero. This makes the Hessian matrix positive-definite and the function g convex. After that, we want to show that $f(\mathbf{P}_C) = g(\frac{\mathbf{P}_C \boldsymbol{\lambda}}{\lambda_C})$ is convex. $g(\frac{\mathbf{P}_C \boldsymbol{\lambda}}{\lambda_C})$ is an affine composition of g so the operation preserves the functions convexity.