TIMELY STATUS UPDATING WITH TIME STAMP ERRORS

by

Md Nurul Absar Siddiky

A thesis submitted to the faculty of
The University of North Carolina at Charlotte
in partial fulfillment of the requirements
for the degree of Master of Science in
Electrical Engineering

Charlotte

2025

Approved by:

Dr. Ahmed Arafa

Dr. Jiang (Linda) Xie

Dr. Ran Zhang

ABSTRACT

MD NURUL ABSAR SIDDIKY. Timely Status Updating with Time Stamp Errors. (Under the direction of DR. AHMED ARAFA)

This thesis examines a status updating system where multiple processes are sampled and transmitted through a shared communication channel. Each process has its dedicated server that processes its samples before time stamping them for transmission. Time stamps, however, are prone to errors, and hence the status updates received may not be credible. Our setting models the time stamp error rate as a function of the servers' busy times. Hence, to reduce errors and enhance credibility, servers need to process samples on a relatively prolonged schedule. This, however, deteriorates timeliness, which is captured through the age of information (AoI) metric.

An optimization problem is formulated whose goal to characterize the optimal processes' schedule and sampling instances to achieve the optimal trade-off between timeliness and credibility. The problem is first solved for a single process setting, where it is shown that a *threshold-based sleep-wake schedule* is optimal, in which the server wakes up and is allowed to process newly incoming samples only if the AoI surpasses a certain threshold that depends on the required timeliness-credibility trade-off. Such insights are then extended to the multi-process setting, where two main scheduling and sleep-wake policies, namely round-robin scheduling with threshold-waiting and asymmetric scheduling with zero-waiting, are introduced and analyzed.

Following this, another perspective of the problem is considered in which a single process is served by multiple servers, each having its dedicated channel to the destination. That is, a time-stamped sample from the same process gets transmitted multiple times by different servers. Each server introduces its own time stamp error, and each channel takes its own time to deliver its time stamps. At the destination, the average of all time stamps is computed. This is shown to reduce the time stamp error,

at the expense of waiting an extra time until all time stamps are received. Hence, a trade-off arises between adding more servers, which reduces the error, and having to wait more time for all of them to finish, which hurts the AoI. We characterize the *optimal number of servers* needed in this variant problem setting that balances AoI and time stamp error.

The results of this thesis provide a framework for designing efficient status updating systems, offering practical insights into balancing timeliness and credibility in real-time monitoring, networked control, and wireless communication systems.

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to my supervisor, Professor Dr. Ahmed Arafa, for his invaluable guidance, encouragement, and support throughout the course of my research. His expertise and insight have been instrumental in shaping this work, and I am truly grateful for the opportunities to learn under his mentorship.

I would also like to thank my thesis committee members, Professor Dr. Jiang (Linda) Xie and Professor Dr. Ran Zhang, for their thoughtful feedback, constructive suggestions, and valuable contributions to this research. Their expertise and advice have significantly enriched the quality of this work.

Finally, I extend my heartfelt thanks to my family, friends, and colleagues for their unwavering support and encouragement throughout this journey.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

AoI	An acronym for Age of Information.
AS	An acronym for Asymmetric Scheduling.
i.i.d.	An acronym for Independent and Identically Distributed.
RR	An acronym for Round Robin.

CHAPTER 1: INTRODUCTION

Several current and emerging applications in communications, networking and control require timely information processing and transfer in order to accurately achieve their goals. This has led to the emergence of the age of information (AoI) metric, which assesses data freshness data at the destinations [1], and is defined as the difference between the current time and the time stamp of the latest received data [2]. In time-sensitive applications, it is crucial to measure the AoI accurately in order to take timely decisions. However, when time stamps are erroneous, the AoI value becomes unreliable, and the credibility of the decision-making process becomes questionable. In this work, we introduce the notion of *timeliness-credibility* trade-off through modeling analyzing the effects of time stamp errors on AoI optimization in a system where multiple processes are monitored through a shared communication channel, see Fig. 1.1.

Optimizing AoI, or maximizing timeliness and freshness of data, has been considered in a plethora of works in the literature. The pioneering work in queuing networks [2] and what follows in that line of research have shown that AoI-optimal policies are neither throughput-optimal (high server utilization) nor delay-optimal (low server utilization). Rather, AoI aims at balancing server utilization to deliver fresh data. Other lines of research to which these ideas are extended include, e.g., energy harvesting communications [3], federated learning [4], gossip networks [5], data trading [6], coding [7], internet-of-things (IoT) networks [8], random access networks [9], edge computing [10], and privacy-preserving systems [11].

A notable challenge to achieve accurate AoI is the issue of *timestomping* [12], where time stamps on data packets are intentionally falsified, typically as part of

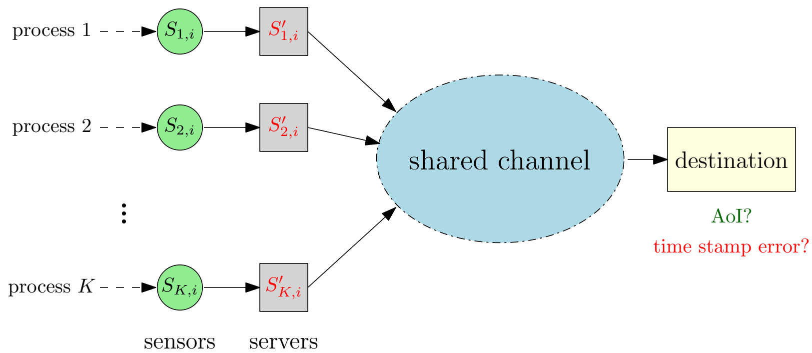

Figure 1.1: System model: process k's ith sample arrives at time $S_{k,i}$, yet is time-stamped as $S'_{k,i}$ by its server.

an adversarial attack. This manipulation can occur through several mechanisms, including falsifying time stamps to make stale data appear fresh, introducing network delays, or due to natural factors such as sensor malfunctions or synchronization issues. These time stamp errors lead to inaccurate AoI calculations, which in turn result in misleading assessment of data freshness. This performance degradation is particularly concerning in energy-constrained IoT systems, remote monitoring, and decentralized networks, where accurate updates are critical. In [13], the effects of adversarial time stamp manipulation in gossip networks have been studied, demonstrating that even a compromised node in fully connected networks can significantly increase AoI and worsen how it scales with the size of the network.

Although time stamp manipulation has been explored in gossip networks, there is a lack of research on how time stamp inaccuracies affect AoI in conventional update systems. To address this gap, our work investigates the *optimization of both AoI and time stamp accuracy,* with the goal of enhancing system reliability and timeliness. Minimizing AoI alone is insufficient when time stamp errors are present, as these errors can lead to poor decision making in critical applications such as remote sensing and energy-constrained systems. Therefore, we propose integrating time stamp error management into AoI optimization to provide more accurate and reliable decision-making and system performance.

Specifically, we consider a system in which time stamps from multiple processes are sent through a shared channel towards a destination. Each process has a dedicated server to process its samples and assign them time stamps before sending them on the channel. A server introduces time stamp errors with a rate that depends on its busy time. That is, to reduce the errors, a server needs to *sleep* for a while. This, in turn increases the AoI and reduces timeliness. Hence, a trade-off arises between minimizing AoI and minimizing time stamp errors. We introduce an optimization problem to characterize the optimal trade-off by optimizing the sleep-wake schedules of the servers. We first solve the problem for the single process setting. Towards that end, we show that the optimal sleep-wake schedule has a *threshold structure:* the server wakes up only if the AoI surpasses a certain threshold that depends on the target credibility of time stamps. We then build on these insights and present two main scheduling policies for the multiple-source setting: round-robin scheduling with threshold-waiting, and asymmetric scheduling with zero-waiting. We analyze and compare the performances of both policies and show that the optimal choice between them highly depends on the system parameters and the target time stamp credibility.

Following this, another perspective of the problem is considered. Instead of serving multiple processes over the same channel, we serve one process over multiple channels. A time stamp from a single process is served by multiple servers and sent through multiple channels to the destination, see Fig. 1.2. Each server introduces its own time stamp error, and each channel introduces its own delay. At the destination, an average of the received time stamps is computed. We show that this averaging reduces the time stamp error, yet increases the AoI since the destination needs to wait for all servers/channels to finish before computing the average. Hence, we characterize another trade-off between AoI and time stamp credibility, captured through the number of available servers/channels. The optimal number of servers that balances AoI and error is then analyzed under a zero-wait policy.

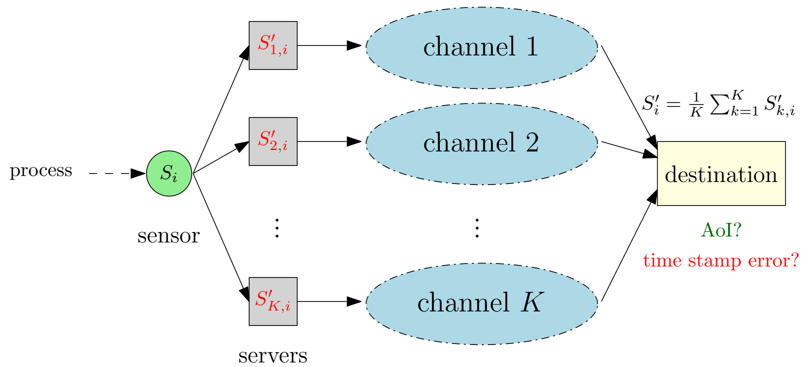

Figure 1.2: Variant system model: a single process is sampled at time S_i and transmitted K times through orthogonal channels. Destination computes the average of all received time stamps $\{S'_{k,i}\}$ to get an estimate S'_i.

The rest of the thesis is organized as follows: Chapter 2 discusses the main setting of monitoring multiple processes through a shared channel; Chapter 3 then discusses the variant problem of monitoring a single process through multiple channels; and Chapter 4 drawas some conclusions and discusses potential future work.

CHAPTER 2: SINGLE CHANNEL STATUS UPDATING SYSTEM WITH TIME STAMP ERROR

2.1 System Model and Problem Formulation

We consider a status update system composed of K sensors and K servers, as shown in Fig. 1.1. Sensor k receives samples from a λ_k-Poisson process, and passes them to server k for time-stamping and transmission. Transmissions go through a shared communication channel that adds a random delay and can only be utilized by one server at a time.

Let π denote the transmission schedule. This schedule also defines a sleep-wake policy for servers (and sensors); to avoid samples becoming stale, sensor k does not acquire new samples and goes into sleep mode unless it is scheduled to transmit according to π. More precisely, when sensor k's turn comes up for the ith time, it may choose to continue to sleep (or wait) for $W_{k,i}$ extra time units, after which it wakes up and becomes ready to receive new samples. Then, it receives its ith sample after $X_{k,i}$ time units. Note that $X_{k,i}$'s are independent and identically distributed (i.i.d.). $\sim \exp(\lambda_k)$ across process k samples. We denote by $\{W_{k,i}\}$ the servers' waiting policy.

Now, let $S_{k,i}$ denote the arrival time of the ith sample of the kth process. Such sample gets served immediately upon arrival, and reaches the destination at time

$$D_{k,i} = S_{k,i} + Y_{k,i}, \tag{2.1}$$

where $Y_{k,i}$'s are i.i.d. across samples and processes, denoting channel busy times. We assume that the destination is *unaware* of the values of the channel busy times $\{Y_{k,i}\}$; only the sensor-server side is aware.

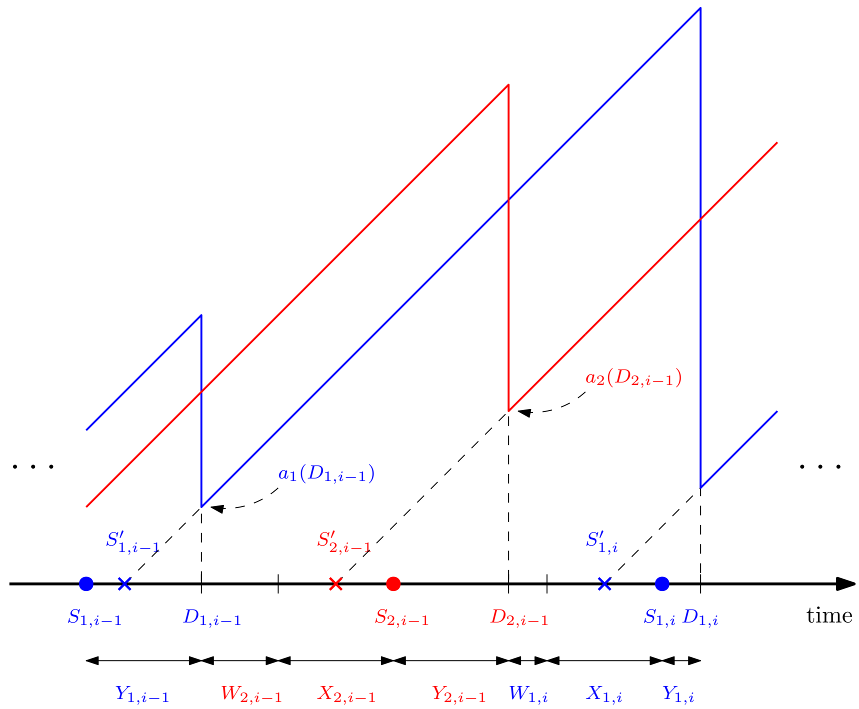

Figure 2.1: Example AoI evolution at the destination for $K = 2$ processes (1 in blue; 2 in red). Filled circles denote true time stamps and crosses denote received (erroneous) time stamps.

Servers may introduce time stamp errors, in which the *received* time stamp of the process k's ith sample is given by $S'_{k,i}$ as opposed to the true time stamp $S_{k,i}$. Errors occur at a rate that depends on the sleep-wake schedule of the sensors as we explain below. Statistically, we assume that $S'_{k,i}$ and $S_{k,i}$ are related as follows:

$$\mathrm{E}\left[S'_{k,i}|S_{k,i}, D_{k,i}\right] = S_{k,i}, \tag{2.2}$$

$$\mathrm{Var}\left(S'_{k,i}|S_{k,i}, D_{k,i}\right) = h_k\left(S_{k,i} - X_{k,i} - S_{k,i-1}\right), \tag{2.3}$$

where $\mathrm{E}[\cdot]$ and $\mathrm{Var}(\cdot)$ denote expectation and variance, respectively, and $h_k(\cdot)$ is some monotonically *decreasing convex* function. Hence, the server's introduced (and received) time stamp $S'_{k,i}$ is unbiased from the true time stamp $S_{k,i}$, yet its variance is inversely proportional with the inter-sampling duration. Therefore, to reduce errors of a certain server, one needs to reduce its sampling rate. The rationale is that *errors occur more often when servers do not get enough sleep time.* Such approach has been considered previously in, e.g., use-dependent channels [14]. While our results are presented for general functions $h_k(\cdot)$, our experiments will be focusing on exponentially-decaying functions given by

$$h_k(x) = e^{-\alpha_k x}, \tag{2.4}$$

for some parameter $\alpha_k \geq 0$ denoting server k's *recovery rate*. That is, higher values of α_k represent faster recovery, in which server k introduces relatively less errors and can tolerate being awake for relatively longer periods of time, and vice versa.

We assess timeliness at the destination using AoI. For process k, the AoI at time t is defined as

$$a_k(t) = t - S'_{k,i}, \quad D_{k,i} \leq t < D_{k,i+1}. \tag{2.5}$$

An example of the AoI evolution at the destination for $K = 2$ processes is shown in Fig. 2.1. Note that due to timestamping errors, the AoI value seen at the destination may not represent the true value of the AoI. We define by an *epoch* the time elapsed in between two consecutive deliveries of samples from a specific process. Let us denote by

$$L_{k,i} \triangleq D_{k,i} - D_{k,i-1} \tag{2.6}$$

the length of the ith epoch for process k. We are interested in the long-term average AoI for process k defined by the area under its AoI curve. From Fig. 2.1, such quantity is given by

$$\overline{\mathtt{AoI}}_k = \limsup_{n\to\infty} \frac{\sum_{i=1}^{n} \mathrm{E}\left[a_k\left(D_{k,i-1}\right) L_{k,i}\right] + \frac{1}{2}\mathrm{E}\left[L_{k,i}^2\right]}{\sum_{i=1}^{n} \mathrm{E}\left[L_{k,i}\right]}. \tag{2.7}$$

As we can see from the above, the timeliness measured at the destination is *not always credible* due to time stamping errors. Therefore, one also needs to measure the long-term average error for a specific process when evaluating its timeliness. Such quantity is given by

$$\overline{\mathtt{e}}_k = \limsup_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} \mathrm{E}\left[\left(S_{k,i} - S'_{k,i}\right)^2\right]. \tag{2.8}$$

Using (2.2) and (2.3), observe that one can reduce the value of $\overline{\mathtt{e}}_k$ by increasing the inter-sampling duration. However, this may negatively impact timeliness. Hence, *a trade-off arises between timeliness and credibility.* Note that a schedule π combined with a waiting policy $\{W_{k,i}\}$ completely characterize the values of $\overline{\mathtt{AoI}}_k$ and $\overline{\mathtt{e}}_k$ for all processes. Our main goal is to optimize a weighted average of timeliness and credibility. That is, to solve the following optimization problem:

$$\min_{\pi,\ \{W_{k,i}\geq 0\}} \quad \sum_{k=1}^{K} \beta_k \overline{\mathtt{AoI}}_k + (1-\beta_k)\,\overline{\mathtt{e}}_k, \tag{2.9}$$

for some $\beta_k \in [0,1],\ \forall k$.

We first solve the single-process version of the above problem in the next section. Then, we present solutions for the multi-process version in the following one.

2.2 The Single Process Setting

In the case of $K=1$ process, we drop the index k from all the variables, and drop the schedule π from the optimization problem in (2.9). That is, the only variable of the optimization problem in (2.9) is now the waiting policy $\{W_i\}$.

Towards characterizing the long-term average AoI, observe that the starting AoI of epoch i is given by

$$a\left(D_{i-1}\right)=\begin{cases}Y_{i-1}-\left(S'_{i-1}-S_{i-1}\right), & S'_{i-1}\geq S_{i-1}\\ Y_{i-1}+S_{i-1}-S'_{i-1}, & S'_{i-1}<S_{i-1}\end{cases}$$
$$=Y_{i-1}+S_{i-1}-S'_{i-1}. \tag{2.10}$$

Next, we focus on *stationary deterministic* waiting policies in which the waiting time in epoch i is given by a deterministic function of the channel busy time in epoch $i-1$. That is,

$$W_i \triangleq \omega\left(Y_{i-1}\right), \tag{2.11}$$

for some function $\omega(\cdot)$ to be optimized. Such waiting policy has been shown optimal in similar settings of AoI optimization [11], in which the channel busy times are i.i.d.[1] Such choice of waiting policies induces a stationary distribution across epochs. Specifically, the ith epoch length is now given by

$$L_i=\omega\left(Y_{i-1}\right)+X_i+Y_i, \tag{2.12}$$

and the long-term average AoI now reduces to the following:

$$\overline{\texttt{AoI}}=\frac{\mathrm{E}\left[a\left(D_{i-1}\right)L_i\right]+\frac{1}{2}\mathrm{E}\left[L_i^2\right]}{\mathrm{E}\left[L_i\right]}. \tag{2.13}$$

Now let us further analyze the term $\mathrm{E}\left[a\left(D_{i-1}\right)L_i\right]$ in the numerator above. Using (2.2) and (2.3), one can see that the time stamp error $S_{i-1}-S'_{i-1}$ only depends on channel busy time Y_{i-2}, and is therefore independent from L_i. Further, one can show

[1]Observe that the waiting policy is determined completely at the sensor-server side where full knowledge of the channel busy times Y_i's is provided. Hence, the policy is implementable in our setting.

that its average value is equal to 0 as follows:

$$\begin{aligned}\mathrm{E}\left[S_{i-1}-S'_{i-1}\right] &= \mathrm{E}\left[\mathrm{E}\left[S_{i-1}-S'_{i-1}|S_{i-1},D_{i-1}\right]\right]\\ &= \mathrm{E}\left[\mathrm{E}\left[S'_{i-1}|S_{i-1},D_{i-1}\right]-\mathrm{E}\left[S'_{i-1}|S_{i-1},D_{i-1}\right]\right]=0.\end{aligned} \tag{2.14}$$

Therefore, using (2.10) we get that

$$\mathrm{E}\left[a\left(D_{i-1}\right)L_i\right]=\mathrm{E}\left[Y_{i-1}L_i\right]. \tag{2.15}$$

In other words, the time stamp error, on average, does not affect the long-term average AoI viewed from the destination.[2]

We now turn to the long-term average error. Since we have stationary distributions across epochs, we get

$$\begin{aligned}\bar{\mathtt{e}} =& \mathrm{E}\left[\left(S_i-S'_i\right)^2\right]\\ =& \mathrm{E}\left[\mathrm{E}\left[\left(S_i-S'_i\right)^2|S_i,D_i\right]\right]\\ =& \mathrm{E}\left[\mathrm{E}\left[\left(\mathrm{E}\left[S'_i|S_i,D_i\right]-S'_i\right)^2|S_i,D_i\right]\right]\\ =& \mathrm{E}\left[\mathrm{Var}\left(S'_i|S_i,D_i\right)\right]\\ =& \mathrm{E}\left[h\left(S_i-X_i-S_{i-1}\right)\right]\\ =& \mathrm{E}\left[h\left(Y_{i-1}+\omega\left(Y_{i-1}\right)\right)\right].\end{aligned} \tag{2.16}$$

The optimization problem is now given by

$$\min_{\omega(\cdot)\geq 0} \quad \overline{\mathtt{AoI}}+\frac{1-\beta}{\beta}\bar{\mathtt{e}}, \tag{2.17}$$

[2]This, however, necessitates the addition of a credibility measure as in (2.8).

which can be equivalently represented as

$$\min_{\omega(\cdot)\geq 0} \quad \overline{\texttt{AoI}}, \quad \text{s.t.} \quad \overline{\texttt{e}} \leq \tau, \tag{2.18}$$

for some $\tau \geq 0$ [15]. We call the constraint in (2.18) the *credibility constraint.* We focus on analyzing the second formulation in (2.18) in the remainder of this section. Specifically, we follow Dinkelbach's approach [16] to transform the problem into the following auxilliary one:

$$\begin{aligned} p(\theta) \triangleq \min_{\omega(\cdot)\geq 0} \quad & \mathrm{E}\left[\left(Y_{i-1}-\theta\right)\left(\omega\left(Y_{i-1}\right)+X_i+Y_i\right)\right]+\frac{1}{2}\mathrm{E}\left[\left(\omega\left(Y_{i-1}\right)+X_i+Y_i\right)^2\right] \\ \text{s.t.} \quad & \mathrm{E}\left[h\left(Y_{i-1}+\omega\left(Y_{i-1}\right)\right)\right] \leq \tau, \end{aligned} \tag{2.19}$$

for some $\theta \in \mathbb{R}$. The optimal solution of problem (2.18) is now given by the unique θ^* that solves $p(\theta^*) = 0$, which can be found by, e.g., a bisection search.

The objective function of problem (2.19) can be further simplified as follows:

$$\begin{aligned} & \mathrm{E}\left[\left(Y_{i-1}-\theta\right)\omega\left(Y_{i-1}\right)\right]+\left(\mu_Y-\theta\right)\left(\frac{1}{\lambda}+\mu_Y\right) \\ & +\frac{1}{2}\mathrm{E}\left[\omega\left(Y_{i-1}\right)^2\right]+\mathrm{E}\left[\omega\left(Y_{i-1}\right)\right]\left(\frac{1}{\lambda}+\mu_Y\right) \\ & +\frac{1}{\lambda^2}+\frac{1}{\lambda}\mu_Y+\frac{1}{2}\mu_{Y^2}, \end{aligned} \tag{2.20}$$

where μ_Y and μ_{Y^2} denote the first and second moments of Y_i, respectively. We now introduce the following Lagrangian for problem (2.19):

$$\begin{aligned} \mathcal{L} = & \int \left(\left(y-\theta+\frac{1}{\lambda}+\mu_Y\right)\omega(y)+\frac{1}{2}\omega(y)^2\right) f_Y(y)dy \\ & +\left(\mu_Y-\theta\right)\left(\frac{1}{\lambda}+\mu_Y\right)+\frac{1}{\lambda^2}+\frac{1}{\lambda}\mu_Y+\frac{1}{2}\mu_{Y^2} \\ & +\gamma\left(\int h\left(y+\omega(y)\right) f_Y(y)dy-\tau\right)-\int \eta(y)\omega(y)dy, \end{aligned} \tag{2.21}$$

where $f_Y(y)$ denotes the distribution of Y_i, whereas γ and $\eta(y)$ are Lagrange multipliers. Taking the functional derivative of the above with respect to $\omega(y)$, equating to 0, and rearranging, we get

$$y + \omega(y) + \gamma h'\left(y + \omega(y)\right) = \theta - \frac{1}{\lambda} - \mu_Y + \frac{\eta(y)}{f_Y(y)}, \tag{2.22}$$

where $h'(\cdot)$ denotes the derivative of $h(\cdot)$. Now observe that since $h(\cdot)$ is convex, and $\gamma \geq 0$, the function

$$H_\gamma(x) \triangleq x + \gamma h'(x) \tag{2.23}$$

is monotonically increasing. Hence, by (2.22) we have

$$\omega(y) = H_\gamma^{-1}\left(\theta - \frac{1}{\lambda} - \mu_Y + \frac{\eta(y)}{f_Y(y)}\right) - y. \tag{2.24}$$

By complementary slackness [15], we get $\eta(y) = 0$ if $\omega(y) > 0$, in which case $H_\gamma^{-1}\left(\theta - \frac{1}{\lambda} - \mu_Y\right) > y$. On the other hand, if $H_\gamma^{-1}\left(\theta - \frac{1}{\lambda} - \mu_Y\right) < y$, then we must have $\eta(y) > 0$ so as to increase the argument inside H_γ^{-1} and make $\omega(y)$ non-negative. This means, again by complementary slackness, that $\omega(y) = 0$. Combining the arguments, we finally have the optimal waiting policy that solves problem (2.19) given by

$$\omega^*(y) = \left[H_{\gamma^*}^{-1}\left(\theta - \frac{1}{\lambda} - \mu_Y\right) - y\right]^+ \tag{2.25}$$

where $[\cdot]^+ \triangleq \max(\cdot, 0)$ and γ^* denotes the optimal Lagrange multiplier associated with the credibility constraint.

The result above shows that the optimal waiting policy has a threshold structure; as long as the starting AoI of an epoch is below a certain threshold, given by

$H_{\gamma^*}^{-1}\left(\theta - \frac{1}{\lambda} - \mu_Y\right)$, the sensor should continue in sleeping mode until the AoI surpasses that threshold, and then wake up. The threshold, however, remains partially unknown unless we can evaluate γ^*. We do so indirectly as follows. First, let us assume that $\gamma^* = 0$. In this case, the threshold is simply given by

$$H_0^{-1}\left(\theta - \frac{1}{\lambda} - \mu_Y\right) = \theta - \frac{1}{\lambda} - \mu_Y. \tag{2.26}$$

We now use the above to check if the credibility constraint is satisfied. If it is not, then it must be that $\gamma^* > 0$, which means by complementary slackness that the credibility constraint is satisfied with equality. In this case, all we need to do is to find some threshold value ξ that solves

$$\mathrm{E}\left[h\left(Y_{i-1} + [\xi - Y_{i-1}]^+\right)\right] = \tau, \tag{2.27}$$

which can be evaluated by, e.g., a bisection search since $h(\cdot)$ is monotonically decreasing.

The approach above provides the optimal solution of problem (2.19), i.e., it evaluates $p(\theta)$. The final step to link all this back to problem (2.18) is to find the optimal value θ^* that solves $p(\theta^*) = 0$. We have now proven the following theorem that summarizes the theoretical results in this section:

The optimal solution of problem (2.18) is given by a threshold-waiting policy $\omega^*(\cdot) = [\xi^* - \cdot]^+$. The threshold ξ^* is given by $\theta^* - \frac{1}{\lambda} - \mu_Y$, provided that the credibility constraint is satisfied. Otherwise, it is given by the solution of (2.27). The value of θ^* is such that $p(\theta^*) = 0$ in (2.19).

2.3 The Multi-Process Setting

We now use the results developed for the single process setting to present solutions for problem (2.9) in the case of $K \geq 2$ processes. We first note that finding the jointly optimal scheduling and waiting policies is highly nontrivial. One reason behind this is that the functions governing the time-stamp errors, $h_k(\cdot)$'s, can vary from one server/process to another. For example, let us consider the exponentially decaying model in (2.4) for $K = 2$. If $\alpha_1 > \alpha_2$, then server 1 *recovers faster* than server 2. Hence, it could be optimal to schedule process 1 *more often* than process 2 so as to allow a sufficient time for server 2 to recover and reduce errors. Thus, the often studied round-robin schedule (or maximum-age-first) in the AoI literature [17] may not be optimal in our setting. This, in addition to the fact that we need to evaluate optimal waiting times renders the problem challenging.

To alleviate this hurdle, in this preliminary work on this problem we aim at developing policies that optimize the scheduling policy or the waiting policy *individually*, as opposed to jointly, and compare their performances against each other. We focus on two specific kinds of policies that we discuss next.

2.3.1 Round Robin Scheduling with Threshold Waiting

The first policy that we discuss considers a round-robin (RR) schedule, in which process 1 is scheduled for sampling, followed 2, all the way to K, and then the schedule repeats. We denote this schedule by π_{RR}.

As for the waiting policy, and given the results of the single process setting, we combine RR scheduling with a threshold-waiting policy, which is illustrated as follows. Instead of waiting prior to each process sampling, we only wait once before scheduling process 1. This is then followed by the scheduled RR transmissions. We note that such approach has been shown optimal in the relatively similar setting of [17]. Hence, let us focus on process 1's epoch. The waiting time at the beginning of epoch i is now given by the following function of epoch $i-1$'s sum service times:

$$\omega\left(\sum_{k=1}^{K} Y_{k,i-1}\right) = \left[\xi - \sum_{k=1}^{K} Y_{k,i-1}\right]^+, \tag{2.28}$$

for some threshold ξ to be optimized. For simplicity of presentation, we refer to the above expression by ω_i^{RR}.

Now observe that for the kth process, we get from (3.4) that

$$L_{k,i} = \sum_{s=k+1}^{K} X_{s,i-1} + Y_{s,i-1} + \omega_i^{RR} + \sum_{s=1}^{k} X_{s,i} + Y_{s,i}. \tag{2.29}$$

The above expression, together with the definition of ω_i^{RR} in (2.28), shows that the location of the waiting time could possibly lead to different distributions of epochs across different processes. However, we note that the *sums* of their corresponding AoI's, denoted by $\overline{\texttt{AoI}}_k(\pi_{RR})$, and time stamp errors, denoted by $\overline{\texttt{e}}_k(\pi_{RR})$, can be shown to be still be the same. Based on this, we focus on the case in which all β_k's are equal in this work. This implies that the relationship between AoI and the time-stamp error remains consistent across all processes, leading to a uniform stationary behavior in the system's performance.[3]

Using the above expression, we simplify equations (2.7) and (2.8) and specialize them to the case of RR scheduling with threshold waiting. The resulting expressions are used for specific service time distribution to get closed-form expression for $\overline{\texttt{AoI}}_k(\pi_{RR})$ and $\overline{\texttt{e}}_k(\pi_{RR})$ in terms of the threshold ξ, which can then be found by, e.g., line search algorithms.

2.3.2 Asymmetric Scheduling with Zero Waiting

The second policy under consideration is an asymmetric scheduling (AS) policy, indicated by π_{AS}. In here, process 1 is scheduled for m_1 sampling and transmission trials, followed by process 2 for m_2 trials, and so on until process K completes its m_k trials, after which the schedule repeats. In this asymmetric schedule, different from π_{RR}, we do not consider waiting prior to transmission.

Using π_{AS}, an epoch for any process k includes the same number of (possibly

[3]The general case in which β_k's are not equal can be solved by optimizing the *location* of the waiting time, and is to be analyzed in future work.

different) transmissions from every other process. Since the waiting time is 0 and all random variables are i.i.d., it follows that the system is stationary and all epochs' distributions are the same. We denote the corresponding AoI and time stamp error for process k by $\overline{\texttt{AoI}}_k(\pi_{AS})$ and $\overline{\texttt{e}}_k(\pi_{AS})$, respectively.

Let us denote by $X_{k,i}^{(j)}$ and $Y_{k,i}^{(j)}$ the jth inter-arrival and service times of the kth process in the ith epoch, respectively, where the epoch index is counted with respect to process 1 without loss of generality. Therefore, we have

$$L_{k,i} = \sum_{s=k+1}^{K} \sum_{j=1}^{m_s} X_{s,i-1}^{(j)} + Y_{s,i-1}^{(j)} + \sum_{s=1}^{k} \sum_{j=1}^{m_s} X_{s,i}^{(j)} + Y_{s,i}^{(j)}. \tag{2.30}$$

Based on the above expression, we specialize the equations in (2.7) and (2.8) and derive the AoI and time stamp error expressions for AS scheduling with zero waiting. Given a service time distribution, one can then find the optimal selection of the number of trials m_k for process k.

2.4 Numerical Results

In this section, we present some numerical results to further illustrate the theoretical analysis of this chapter. We focus on showing the AoI vs. time stamp error trade-off under different system settings. From the optimization problem in (2.9), such a trade-off can be characterized by varying the values of β_k's. In our simulations, and in agreement with our theoretical results, we focus on the case in which $\beta_k = \beta,\ \forall k$. The service time distribution is $\sim \exp(1/\mu_Y)$.

We first present results for the single process setting. In Fig. 2.2, we vary the value of $\beta \in [0, 1]$ to show how the AoI behaves with respect to time stamp error. We set $\lambda = 9$ and $\mu_Y = 1$. Clearly, the higher the value of β the better the AoI and the worse the error, and vice versa. Moreover, as the recovery rate α increases, the trade-off behaves better: for a relatively higher value of α, one can achieve lower errors for the same AoI values. As a baseline, we show the single AoI-error pair achieved by

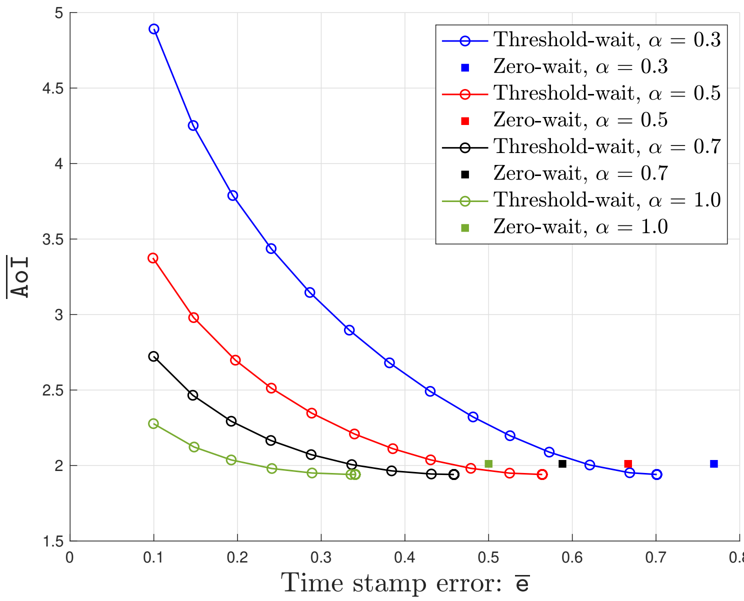

Figure 2.2: Single process AoI vs. time-stamp error for different recovery rate α values.

the zero-wait policy. It is clear from the figure that the optimal threshold-wait policy outperforms zero-wait in terms of both AoI and error for relatively higher values of β.

Next, we compare the behavior of π_{RR} and π_{AS} for $K = 2$ processes. We set $\lambda_1 = \lambda_2 = 6$, $\alpha_2 = 50$, and $\mu = 1.5$, and plot the sum AoI vs. sum time stamp error (by varying β) in Fig. 2.3. The results generally show that the behavior of π_{RR} relative to π_{AS} depends on the value of α_1 and β. For instance, for smaller α_1 and smaller β, π_{AS} is favored upon π_{RR}. While for larger α_1 and larger β the situation is reversed. For intermediate values no specific policy dominates the other. This shows that the choice of the scheduling and waiting policy for this problem is highly dependent on the system dynamics, especially the servers' recovery rates.

A key observation is that when $\alpha_1 = 0.1$, the Asymmetric Scheduling policy outperforms the Round Robin policy for most of the plotted range, particularly when the total error is low (i.e., $\overline{e}_1 + \overline{e}_2 \lesssim 0.7$). This outcome is due to the fact that a low α_1 implies that server 1 has a slow recovery rate, and thus requires longer idle peri-

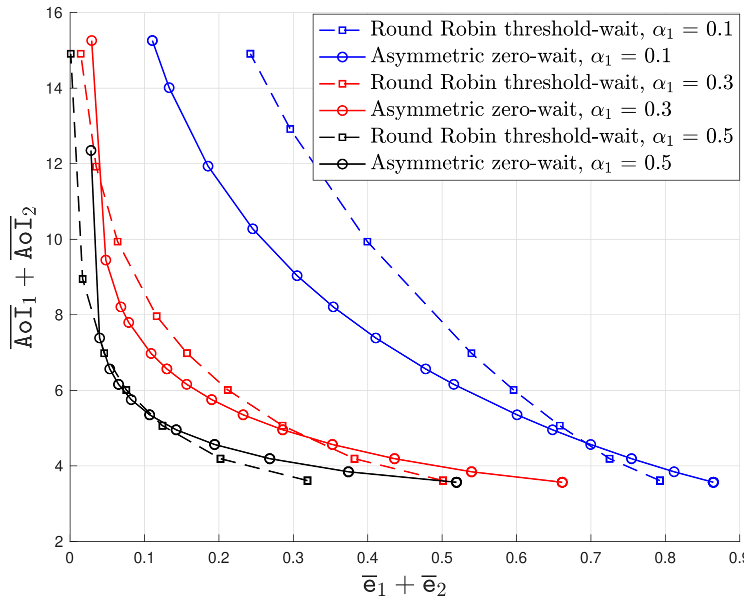

Figure 2.3: Two processes sum AoI vs. sum time stamp error for different process 1 recovery rate α_1 values.

ods to reduce its time-stamp error. The Asymmetric Scheduling policy, by assigning multiple consecutive transmission slots (i.e., m_2 transmissions) to process 2 before switching to process 1, implicitly gives server 1 a longer idle window to recover. This idle time helps reduce server 1's error without the need for explicit waiting. Additionally, because Asymmetric Scheduling avoids deliberate waiting and reuses the channel more frequently for the same process, it achieves better timeliness (lower AoI) while still managing error effectively under low α_1. In contrast, the Round Robin policy uses a threshold-based wait time only once per cycle (before process 1's turn), and then strictly alternates between processes. When α_1 is low, this limited waiting is insufficient for server 1 to fully recover between transmissions, resulting in higher error and suboptimal performance in the low-error regime.

However, the scenario changes when α_1 increases to 0.5. With this higher recovery rate, server 1 becomes more robust to continuous transmissions and does not need extended idle periods to maintain low error. In this case, the advantage of Asymmetric

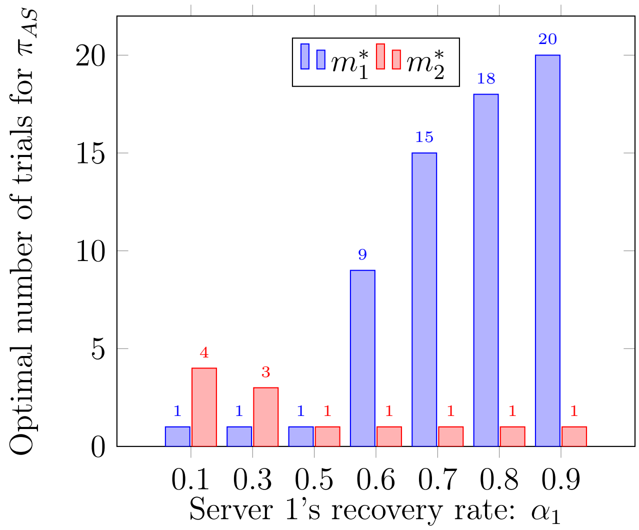

Figure 2.4: Optimal AS policy behavior: (m_1^*, m_2^*) vs. α_1.

Scheduling in giving server 1 more idle time becomes less significant. Meanwhile, the Round Robin policy can now take full advantage of threshold-waiting: by optimally tuning the threshold wait time before each cycle, it balances both processes' recovery needs without sacrificing too much freshness. Since both servers are now capable of handling more frequent transmissions with minimal error, the structured alternation and controlled waiting in Round Robin yield a better overall trade-off, particularly in the higher-error regime. This reversal in performance highlights how the optimality of a scheduling policy critically depends on the servers' recovery rates and the desired timeliness-credibility trade-off.

Finally, the column chart in Fig. 2.4 illustrates the optimal number of trials for π_{AS}, m_1^* and m_2^*, vs. different values of server 1's recovery rate α_1. Here, we set $\alpha_2 = 0.5$, $\lambda_1 = \lambda_2 = 90$, $\mu = 50$, and $\beta = 0.5$. As α_1 increases, server 1 recovers relatively faster than server 2, and therefore m_1^* increases while m_2^* decreases as seen in the figure. This trend underscores the importance of tailoring transmission strategies to specific server recovery rates to optimize timeliness and credibility.

CHAPTER 3: MULTI-CHANNEL STATUS UPDATING SYSTEM WITH TIME STAMP ERROR

3.1 System Model

In this chapter, we consider a system model consisting of a single process, a single sensor, and K servers. Each server transmits sampled data to a destination through its dedicated channel, see Fig. 1.2, where the sensor receives samples from the process at a Poisson rate λ and forwards each sample to all K servers. Each server then transmits the update to the destination through its orthogonal channel, ensuring no interference between channels.

The sensor samples the process at time S_i after an interval $X_i \sim \exp(\lambda)$. Upon sampling, the sensor immediately transmits the sample to all K servers. Each server processes the sample, records its receiving timestamp, and forwards it to the destination through its dedicated channel. The channel busy time for the ith sample transmitted through the k-th channel is denoted by $Y_{k,i}$, where $Y_{k,i}$'s are independent and identically distributed (i.i.d.) across samples and servers. The destination needs to wait for all samples from all channels before updating its AoI. Hence, the total *channel time* for the ith sample is given by

$$\tilde{Y}_i \triangleq \max(Y_{1,i}, Y_{2,i}, \ldots, Y_{K,i}). \tag{3.1}$$

Consequently, the ith sample arrives at the destination at time

$$D_i = S_i + \tilde{Y}_i. \tag{3.2}$$

Each server may introduce time stamp errors. For instance, the received time stamp of the ith sample processed by server k is denoted by $S'_{k,i}$, which may differ from other time stamps delivered through other servers, let alone the true time stamp S_i. The destination estimates the time stamp of the ith sample at the destination by computing the average of the erroneous time stamps from all K servers as follows:

$$\tilde{S}_i \triangleq \frac{1}{K} \sum_{k=1}^{K} S'_{k,i}. \tag{3.3}$$

We define the ith epoch, L_i, as done previously. In terms of the notation of this chapter, this is given by

$$\begin{aligned} L_i =& D_i - D_{i-1} \\ =& X_i + \tilde{Y}_i. \end{aligned} \tag{3.4}$$

In this chapter, we focus on zero-wait policies. Next, we present our main results.

3.2 Main Results and Problem Formulation

In this section we present our main results of this chapter, which is characterizing the long-term average AoI and time stamp error for this variant multi-channel setting considered. We then formulate an optimization problem to choose the optimal number of servers/channels to characterize the optimal trade-off between AoI and error.

We focus on the case of $Y_{k,i}$'s being i.i.d.$\sim \exp(\mu)$.

Lemma 1. *The long-term average AoI for the multi-channel system with K servers is given by*

$$\overline{\texttt{AoI}}(K) = \frac{1}{\mu} \sum_{k=1}^{K} \frac{1}{k} + \frac{\frac{1}{2}\left[\frac{1}{\lambda^2} + \frac{1}{\mu^2}\sum_{k=1}^{K}\frac{1}{k^2} + \left(\frac{1}{\lambda} + \frac{1}{\mu}\sum_{k=1}^{K}\frac{1}{k}\right)^2\right]}{\frac{1}{\lambda} + \frac{1}{\mu}\sum_{k=1}^{K}\frac{1}{k}}. \tag{3.5}$$

Proof. Observe that epoch lengths are i.i.d. Hence, one can use the result in (2.13) to characterize the AoI. Next, we evaluate each term in (2.13) when tailored to our current setting.

We first compute $\mathrm{E}(L_i)$ as

$$\begin{aligned}\mathrm{E}(L_i) =& \mathrm{E}[X_i] + \mathrm{E}[\tilde{Y}_i] \\ =& \frac{1}{\lambda} + \frac{1}{\mu}\sum_{k=1}^{K}\frac{1}{k},\end{aligned} \tag{3.6}$$

where the second inequality follows by noticing that $\tilde{Y}_i$ is the first order statistic of K i.i.d. $\sim \exp(\mu)$ random variables, see, e.g., [18].

Next, we compute $\mathrm{E}\left[a(D_{i-1})L_i\right]$. Observe that as done in Chapter 2, we have

$$a(D_{i-1}) = \tilde{Y}_{i-1} + S_{i-1} - \tilde{S'}_{i-1}, \tag{3.7}$$

which is independent from L_i. Hence, we have

$$\begin{aligned}\mathrm{E}\left[a(D_{i-1})L_i\right] =& \mathrm{E}[\tilde{Y}_{i-1}]\mathrm{E}[L_i] \\ =& \left(\frac{1}{\mu}\sum_{k=1}^{K}\frac{1}{k}\right)\left(\frac{1}{\lambda} + \frac{1}{\mu}\sum_{k=1}^{K}\frac{1}{k}\right).\end{aligned} \tag{3.8}$$

Finally, we compute $\mathrm{E}(L_i^2)$ as

$$\begin{aligned}\mathrm{E}(L_i^2) =& \mathrm{E}\left(X_i + \tilde{Y}_i\right)^2 \\ =& \mathrm{Var}(X_i + \tilde{Y}_i) + \left(\mathrm{E}(X_i + \tilde{Y}_i)\right)^2 \\ =& \frac{1}{\lambda^2} + \frac{1}{\mu^2}\sum_{k=1}^{K}\frac{1}{k^2} + \left(\frac{1}{\lambda} + \frac{1}{\mu}\sum_{k=1}^{K}\frac{1}{k}\right)^2,\end{aligned} \tag{3.9}$$

where the last equality follows again from the first order statistic properties of exponential random variables [18].

Substituting the above three quantities into the AoI formula in (2.13) gives the result in the lemma. $\square$

Lemma 2. *The long-term average time stamp error for the multi-channel system with K servers is given by*

$$\bar{\mathsf{e}}(K) = \frac{1}{K}\sum_{k=1}^{K}\sum_{l=0}^{K-1}(-1)^l\binom{K-1}{l}\frac{\mu}{\alpha_k+(l+1)\mu}. \tag{3.10}$$

Proof. Since the epoch lengths are i.i.d., we have

$$\begin{aligned}
\bar{\mathsf{e}}(K) &= \mathrm{E}\left[(S_i-\tilde{S}_i)^2\right] \\
&= \mathrm{E}\left[\mathrm{E}\left[(S_i-\tilde{S}_i)^2 \mid S_i, D_i\right]\right] \\
&= \mathrm{E}\left[\mathrm{E}\left[(\mathrm{E}[\tilde{S}_i \mid S_i, D_i]-\tilde{S}_i)^2 \mid S_i, D_i\right]\right] \\
&= \mathrm{E}\left[\mathrm{Var}(\tilde{S}_i \mid S_i, D_i)\right] \\
&= \mathrm{E}\left[\mathrm{Var}\left(\frac{1}{K}\sum_{k=1}^{K}S'_{k,i} \mid S_i, D_i\right)\right] \\
&= \mathrm{E}\left[\frac{1}{K^2}\sum_{k=1}^{K}\mathrm{Var}(S'_{k,i} \mid S_i, D_i)\right] \\
&= \frac{1}{K^2}\mathrm{E}\left[\sum_{k=1}^{K}h_k(S_i-X_i-S_{i-1})\right] \\
&= \frac{1}{K^2}\sum_{k=1}^{K}\mathrm{E}\left[e^{-\alpha_k\tilde{Y}_{i-1}}\right] \\
&= \frac{1}{K^2}\sum_{k=1}^{K}\int_0^\infty e^{-\alpha_k y}yK\mu e^{-\mu y}(1-e^{-\mu y})^{K-1}dy.
\end{aligned} \tag{3.11}$$

Expanding $(1-e^{-\mu y})^{K-1}$ using the binomial theorem, we get

$$\begin{aligned}
(1-e^{-\mu y})^{K-1} &= \sum_{\ell=0}^{K-1}\binom{K-1}{\ell}(-e^{-\mu y})^\ell \\
&= \sum_{\ell=0}^{K-1}(-1)^\ell\binom{K-1}{\ell}e^{-\ell\mu y}.
\end{aligned} \tag{3.12}$$

Substituting this expansion into the integral we have

$$\overline{\mathtt{e}}(K) = \frac{1}{K}\sum_{k=1}^{K}\sum_{\ell=0}^{K-1}(-1)^{\ell}\binom{K-1}{\ell}\frac{\mu}{\alpha_k+(\ell+1)\mu}\int_0^{\infty}(\alpha_k+(\ell+1)\mu)e^{-(\alpha_k+(\ell+1)\mu)y}dy, \tag{3.13}$$

which gives the result in the lemma upon evaluating the integral and simplifying. □

The expressions derived above, $\overline{\mathtt{AoI}}(K)$ and $\overline{\mathtt{e}}(K)$, can be shown to be behaving oppositely with K. Specifically, and as we show in the next section, adding more servers/channels increases $\overline{\mathtt{AoI}}(K)$ and decreases $\overline{\mathtt{e}}(K)$. Therefore, there exists an optimal value of K that minimizes the weighted sum of both expressions. This can be formulated as the following optimization problem:

$$\min_{K} \quad \beta\overline{\mathtt{AoI}}(K) + (1-\beta)\overline{\mathtt{e}}(K), \tag{3.14}$$

for some $\beta \in [0, 1]$.

3.3 Numerical Results

In this section, we present some experimental results to validate the theoretical analysis derived in this chapter. First, Fig. 3.1 illustrates the impact of varying the number of redundant channels or servers K on the system performance in terms of the average Age of Information (AoI), $\overline{\text{AoI}}(K)$, and the average timestamp error, $\overline{e}(K)$. The results are generated using an arrival rate $\lambda = 0.5$, fixed service rate $\mu = 0.5$ and fixed server recovery rate $\alpha = 0.1$ for all servers. As shown in the figure, increasing K leads to a monotonically increasing AoI due to higher cumulative service delays introduced by more parallel servers (higher channel time). Conversely, the time stamp error decreases significantly with K, attributed to the error-averaging effect at the destination. The plot highlights a key trade-off: while increasing redundancy improves time stamp precision, it also degrades information freshness. This underscores the

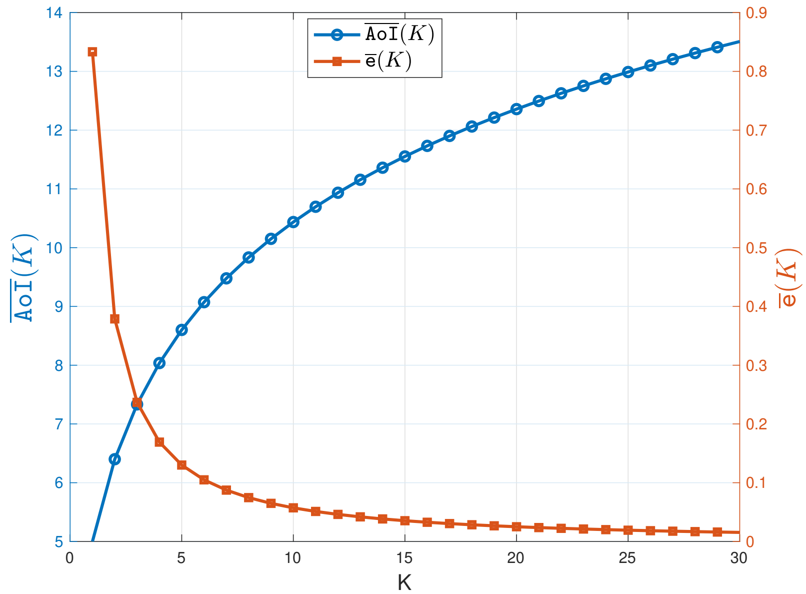

Figure 3.1: Impact of K on AoI and timestamp error.

importance of carefully choosing K based on system design priorities and the relative significance of AoI versus time stamp accuracy.

Fig. 3.2 illustrates the relationship between AoI and time stamp error under varying server recovery rates (α), for a fixed arrival rate $\lambda = 0.5$, fixed service rate $\mu = 0.5$ and three values of server recovery rate $\alpha \in 0.1, 0.5, 0.9$ (fixed across all servers). The results are obtained after optimizing over k for each value of the trade-off parameter $\beta \in [0, 1]]$. That is, we solve problem (3.14).

The results demonstrate an inverse relationship across all values of α: as the timestamp error increases, the AoI decreases. Systems with higher server recovery rates (α) consistently achieve lower AoI, implying that faster recovery enhances information freshness. Conversely, lower α values lead to significantly higher AoI, particularly when error levels are low. The curves show that for small α, AoI drops steeply as error increases, while for larger α, the decline is more gradual.

These observations underscore the trade-off between AoI and timestamp accuracy and emphasize the necessity of carefully choosing α and the number of servers/channels

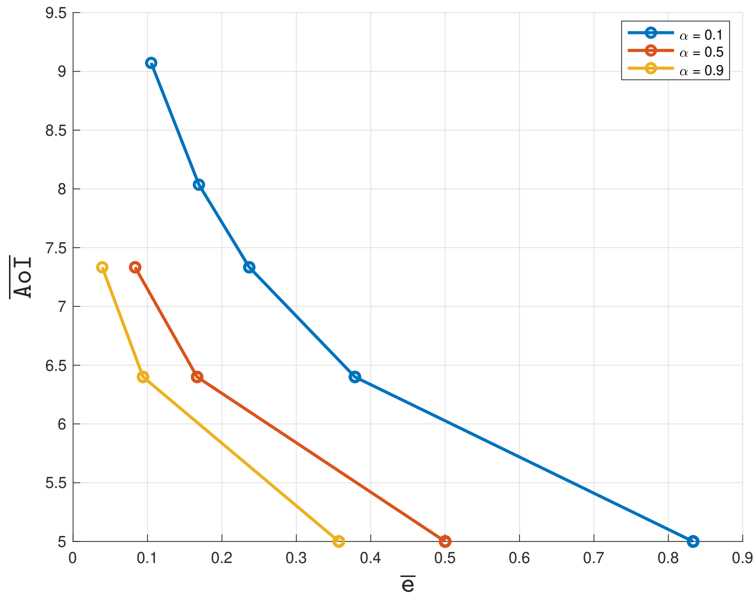

Figure 3.2: Average AoI vs. timestamp error for $\alpha \in \{0.1, 0.5, 0.9\}$, with K optimized over $[1, 30]$ for each $\beta \in [0.05{:}0.025{:}0.95]$.

(representing the redundancy level) K based on system constraints and performance objectives.

CHAPTER 4: CONCLUSIONS AND FUTURE WORK

In this thesis, we have investigated the impact of time stamp errors on the credibility of Age of Information (AoI) in status updating systems. The core of the study revolves around the analysis of how sensor-acquired data, processed by servers and transmitted through a shared channel, can be affected by time stamp errors. These errors, which arise due to inaccuracies in the timestamping process, are modeled as a function of the servers' busy times.

A key observation made in this research is the trade-off between time stamp errors and AoI. Specifically, we found that allowing servers to have more sleeping time, thereby providing them with more time for recovery, can reduce the occurrence of time stamp errors. However, this also results in an increase in AoI, as the system introduces a delay in the transmission of information. This trade-off is critical in designing systems that balance the need for accurate information with the desire to keep the system responsive.

To model this trade-off more rigorously, we formulated an optimization problem aimed at characterizing the optimal timeliness-credibility balance. The problem involved designing efficient scheduling and server sleep-wake policies that minimize AoI while controlling the rate of time stamp errors.

In the context of a single process system, we showed that the optimal sleep-wake policy follows a threshold structure, offering a clear decision rule for when servers should switch between sleep and wake states to achieve the desired trade-off. For multi-process systems, two distinct scheduling schemes were explored: round-robin (symmetric) scheduling and asymmetric scheduling. Our analysis revealed that the choice of scheduling policy plays a significant role in the system's performance, with round-robin scheduling providing a fairer distribution of server time but potentially leading to suboptimal performance in certain configurations.

Through extensive simulations and theoretical analysis, we demonstrated that the recovery rates of servers have a profound effect on the timeliness-credibility trade-off curves. Higher recovery rates are shown to improve the overall system performance by reducing time stamp errors, but they also require careful consideration of the system's sampling rates, channel service rate, and relative recovery rates across multiple servers.

Our results emphasize the importance of selecting appropriate scheduling policies based on these system parameters. Specifically, the trade-off between minimizing AoI and maintaining accurate time stamps should guide the choice of policy, ensuring that the system performs optimally under varying conditions. The insights from this work contribute to the development of more efficient status updating systems, particularly in applications where the timeliness and credibility of information are crucial, such as in real-time monitoring, Internet of Things (IoT) networks, and wireless communication systems.

Following this, we analyzed the impact of redundancy in a multi-server system on the AoI and time stamp error. By modeling a system with K servers, each transmitting updates through dedicated channels, our results demonstrated that increasing the number of servers improves time stamp accuracy by reducing time stamp error through averaging, but it also increases AoI due to longer channel occupancy. This trade-off highlights the necessity of optimizing K to balance freshness and accuracy. Through numerical validation, we established an optimal server count that minimizes a weighted combination of AoI and time stamp error, providing valuable insights into designing efficient multi-server status update systems.

The findings of this study provide a comprehensive understanding of the trade-offs involved in multi-server systems and offer a practical framework for optimizing system performance. By carefully selecting the number of servers K and the weighting parameter β, system designers can achieve the desired balance between data freshness and accuracy, tailored to the specific needs of their applications.

Future research can examine systems with multiple sensors, processes, and servers using dedicated channels. Key areas include analyzing AoI and time stamp errors, synchronizing updates, and optimizing server allocation to improve efficiency. Studying the effects of processing delays, varying sampling rates, and developing adaptive algorithms for resource allocation can enhance performance. These efforts aim to create scalable, efficient real-time monitoring systems for applications like industrial automation and autonomous networks.

REFERENCES

[1] R. D. Yates, Y. Sun, R. D. Brown, S. K. Kaul, E. Modiano, and S. Ulukus, "Age of information: An introduction and survey," *IEEE J. Sel. Areas Commun.*, vol. 39, pp. 1183–1210, May 2021.

[2] S. Kaul, R. D. Yates, and M. Gruteser, "Real-time status: How often should one update?," in *Proc. IEEE INFOCOM*, March 2012.

[3] A. Arafa, J. Yang, S. Ulukus, and H. V. Poor, "Timely status updating over erasure channels using an energy harvesting sensor: Single and multiple sources," *IEEE Trans. Green Commun. Netw.*, vol. 6, no. 1, pp. 6–19, 2022.

[4] H. Yang, A. Arafa, T. Q. S. Quek, and H. V. Poor, "Age-based scheduling policy for federated learning in mobile edge networks," in *Proc. IEEE ICASSP*, April 2020.

[5] R. D. Yates, "The age of gossip in networks," in *Proc. IEEE ISIT*, July 2021.

[6] J. Heand, Q. Ma, M. Zhang, and J. Huang, "Optimal fresh data sampling and trading," in *Proc. IEEE WiOpt*, October 2021.

[7] H. Sac, T. Bacinoglu, E. Uysal-Biyikoglu, and G. Durisi, "Age-optimal channel coding blocklength for an M/G/1 queue with HARQ," in *Proc. IEEE SPAWC*, June 2018.

[8] M. A. Abd-Elmagid, N. Pappas, and H. S. Dhillon, "On the role of age of information in the internet of things," *IEEE Commun. Mag.*, vol. 57, pp. 72–77, December 2019.

[9] F. Zhao, N. Pappas, C. Ma, X. Sun, T. Q. S. Quek, and H. H. Yang, "Optimizing information freshness in mobile networks with age-threshold aloha," in *Proc. IEEE ISIT*, July 2024.

[10] X. He, C. You, and T. Q. S. Quek, "Age-based scheduling for mobile edge computing: A deep reinforcement learning approach," *IEEE Trans. Mobile Comput.*, vol. 23, pp. 9881–9897, February 2024.

[11] A. Arafa and K. Banawan, "Private status updating with erasures: A case for retransmission without resampling," in *Proc. IEEE ICC*, May 2023.

[12] W. Minnaard, "Timestomping NFTS," Master's thesis, University of Amsterdam, July 2014.

[13] P. Kaswan and S. Ulukus, "Timestomping vulnerability of age-sensitive gossip networks," *IEEE Trans. Commun.*, vol. 72, pp. 4193–4205, July 2024.

[14] D. Ward, *Remote Estimation Over Use-Dependent Channels.* PhD thesis, University of Maryland, 2017.

[15] S. Boyd and L. Vandenberghe, *Convex optimization.* Cambridge university press, 2004.

[16] W. Dinkelbach, “"On nonlinear fractional programming",” *Manag. Sci.*, vol. 13, no. 7, pp. 492–498, 1967.

[17] K. Banawan, A. Arafa, and K. G. Seddik, “Timely multi-process estimation over erasure channels with and without feedback: Signal-independent policies,” *IEEE J. Sel. Areas Inf. Theory*, vol. 4, pp. 607–623, November 2023.

[18] H. A. David and H. N. Nagaraja, *Order statistics.* John Wiley & Sons, 2004.