Balancing Current and Historical State Information in Remote Tracking Systems: A Randomized Update Approach

Sunjung Kang, Chengzhang Li, Atilla Eryilmaz, and Ness B. Shroff

***Abstract*—The traditional goal in remote tracking of a dynamic source is to keep the current estimate at the destination as close as possible to the true state. However, in domains such as surveillance applications, the destination is also interested in reconstructing the past trajectory of states for further processing. This requires striking a balance between providing current versus past state information so that the destination can optimize the trade-off between the metrics of freshness and reconstruction queue length. In this work, we propose a randomized update policy that decides between head-of-line versus tail-of-line packets in the update queue. As such, our policy combines the strength of Last-Come-First-Serve (LCFS) service discipline (which aims at reducing the age) with the strength of First-Come-First-Serve (FCFS) service discipline (which aims at reducing the reconstruction delay). We evaluate the performance of our proposed policy in terms of its randomization parameter, which can be optimized given the system parameters to achieve a better trade-off.**

I. Introduction

With the emergence of Internet of Things (IoT), remote tracking systems including for surveillance and healthcare monitoring are expected to become increasingly popular [1]. These systems rely on sensors transmitting update packets with time-varying data to a remote monitor, ensuring the continuous tracking of objects. Timely and accurate updates are essential for maintaining up-to-date information in these systems.

Resource constraints in IoT systems, such as limited communication and energy, can prevent timely updates. Addressing this, there have been various studies conducted in the fields of Age of Information (AoI) [2]–[7] and Remote Estimation (RE) [8]–[14], where the value of information is measured with freshness and accuracy, respectively. The *age* is a quantitative measure used as a performance metric to assess the freshness of information. It is defined as the time elapsed since the latest data available at the destination was generated at its source. Similarly, in the context of RE, the *estimation error* is employed to measure the accuracy of information held by a remote monitor, in comparison to the actual information.

In this work, we go beyond achieving freshness of updates to investigate real-time remote tracking systems integrated with trajectory reconstruction. This integration aims to enable accurate and efficient monitoring and analysis of objects with time-varying characteristics. Remote tracking systems have versatile applications. In surveillance and security, they monitor public spaces and critical facilities, providing real-time updates for immediate threat detection and employing trajectory reconstruction to analyze movement patterns for unusual behavior. In healthcare, wearable devices leverage these systems to track physical activities and health metrics. Timely information updates facilitate real-time health monitoring and instant feedback, while trajectory reconstruction aids in analyzing exercise trends, progress tracking, and tailoring personalized fitness plans.

To achieve real-time tracking of the most recent state, the transmitter selectively maintains the freshest packet while discarding older packets. Conversely, for trajectory reconstruction, it needs to send historical packets, facilitating the reconstruction of object trajectories for further analysis. Therefore, the transmitter is required to make scheduling decisions among the packets available in its queue to balance between real-time tracking and trajectory reconstruction.

A. Related Works

One key challenge in AoI research is to determine the optimal strategies for sampling and scheduling [2]–[7] Specifically, this involves identifying the ideal time for data sampling and transmission with the objective of minimizing the age metric, while considering any relevant constraints. There have been studies focused on sampling strategies for a single source-receiver pair under various conditions and constraints [2], [3] Additionally, there have also been studies focused on wireless networks with multiple sources that are tracked by a common receiver over a shared wireless channel [4], [5] Further, the AoI framework has found applications in various domains, including UAV-assisted networks [6] and vehicular networks [7]

Similar to the challenges faced in AoI research, determining optimal strategies for sampling and scheduling is also a challenge in the context of remote estimation. Several works have addressed this challenge, studying single source-receiver pairs under various conditions and constraints [8], [9] Other research focuses on network scenarios with multiple sources updating a common receiver [10]–[12] In these studies, the state of each source is modeled as a Linear Time Invariant

S. Kang, C. Li and A. Eryilmaz are with the Department of Electrical and Computer Engineering, The Ohio State University, OH 43210, USA. E-mail: {kang.853,li.13488, eryilmaz.2}@osu.edu

N. B. Shroff is with the Department of Electrical and Computer Engineering and Computer Science and Engineering, The Ohio State University, OH 43210, USA. E-mail: shroff.11@osu.edu

This work has been supported in part by NSF grants: NSF AI Institute (AI-EDGE) 2112471, CNS-NeTS-2106679, CNS-NeTS-2007231, CNS-2312836, CNS- 2223452, CNS-2225561, CNS- 2106933, CNS- 2106932; the ONR Grant N00014-19-1-2621; and a grant from the Army Research Office: W911NF-21-1-0244, and was sponsored by the Army Research Laboratory under Cooperative Agreement Number W911NF-23-2-0225. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

(LTI) system with independent zero-mean Gaussian noise [10] , an Ornstein-Uhlenbeck (OU) process [11], or a zero-mean independent and identically distributed random process [12].

B. Contributions

In this work, we consider a remote tracking system with a pair of source/transmitter and a remote monitor/receiver, where the monitor remotely tracks not only the current state of the source but also the trajectory of its evolution over time. Applying existing works [2]–[5] directly to our problem is not feasible as the existing solutions primarily address the freshness of information, which fails to capture the requirements of trajectory reconstruction.

Our contributions can be summarized as follows.

- We introduce the concept of a *reconstruction queue* and focus on its queue length as a quantitative measure for evaluating the cost of trajectory reconstruction. This cost can measure the memory space, computational load or processing time delay of the reconstruction operation, depending on how the reconstruction is performed. This metric, combined with the concept of the age, allows us to establish an optimization problem balancing timely updates and trajectory reconstruction in Section II.
- We show that neither a First-Come First-Serve (FCFS) nor Last-Come First-Serve (LCFS) policy can effectively balance the costs associated with the freshness of information (age) and trajectory reconstruction (reconstruction queue length) in Section III-A.
- To improve performance in both aspects, we propose a randomized update policy that probabilistically chooses between the LCFS and FCFS. We analyze the performance of the proposed policy in Section III-B.
- We evaluate the proposed randomization policies through numerical simulations in Section IV.

II. System Model

We consider a remote monitoring system, represented in Fig. 1, which consists of a source (or transmitter) and a remote estimator (or receiver). In this system, the remote estimator tracks the time-varying state of the source, wherein its focus is twofold: (i) the most recent state and (ii) the historical evolution of the state. Consequently, the transmitter is responsible for effective scheduling between the freshest update packet and the staled packets, acknowledging the significance of both aspects to the receiver.

The state of the source, denoted as $x(t)$, evolves over time. The transmitter randomly samples $x(t)$, generating update packets following a Poisson process with rate $\lambda > 0$. The generated update packets are stored in a queue held by the transmitter, where $Q_s(t)$ denotes its queue length at time t. The transmitter's queue consists of a buffer with an infinite capacity and a server handling one packet at a time. The server's service times follow an exponential distribution with rate $\mu > 0$, where $\lambda < \mu$.

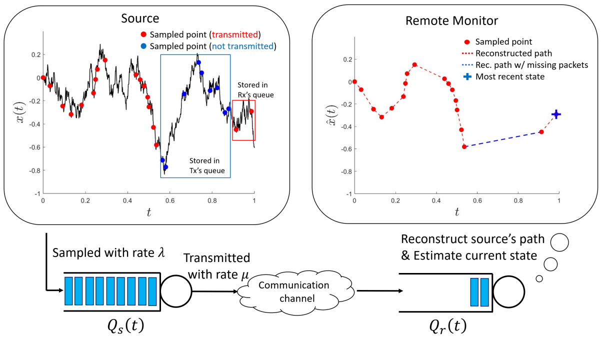

Fig. 1: System model.

The receiver also holds a queue, referred to as the *reconstruction queue*, designed to store packets delivered from the transmitter. Its length at time t is $Q_r(t)$ with average length $\bar{Q}^r = \lim_{t\to\infty} \frac{1}{t}\int_0^t Q_r(s)ds$. Upon receiving an update packet, the receiver can update its estimate for the source's state if this packet is fresher than the previously held information. This aids in addressing the first objective of maintaining up-to-date information. For reconstruction purposes, an update packet j is stored in this queue if there are preceding packets generated earlier that have not yet been delivered due to out-of-order transmissions from the source queue (as explained above). This approach ensures chronological accuracy in data reconstruction, as the packet j will wait in the queue until all older packets are received, preserving the sequence of events.

In Fig. 1, the monitor reconstructs the trajectory of the source's state using the received update packets. This involves estimating the state values at intervals between the sampling points. An increased number of undelivered packets between two sampling points can lead to greater uncertainty in these estimations. However, as these missing packets are eventually delivered and processed, the uncertainty associated with these estimates is expected to decrease, leading to a more accurate reconstruction of the state trajectory [15]. In this case, a high $Q_r(t)$ may suggest greater uncertainty in the reconstruction. Additionally, the monitor relies on the freshest update packet to estimate the current state of the source.

Update packets are generated at times $t_1, t_2, ...$ and received by the receiver at times $t'_1, t'_2, ...$ We assume that $t_i < t_j$ for $i < j$, while the order of delivery times t'_i may not align with the generation order depending on the chosen service discipline. The age $A(t)$ at time t is defined as the time elapsed since the last received fresh packet has been generated, i.e.,

$$A(t) := \min_i \{t - t_i \mid t'_i \le t\} \tag{1}$$

with the initial age $A(0) = A_0$.

Let i_k denote the index associated with the k-th fresh packet delivered to the receiver. Then, upon the receipt of k^{th} fresh packet by the receiver, the *peak age*[1] A_k^{peak} is defined as the maximum age attained immediately prior to receiving the fresh packet: $A_k^{peak} := A(t'^{-}_{i_k})$, where $A(t^-) = \lim_{s\to t^-} A(s)$. The average peak age $\bar{A}^{peak}$ is defined as

[1] In this paper, we consider the peak age instead of the average age, the latter being the metric most typically used in the literature, due to mathematical tractability.

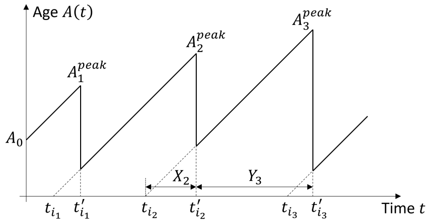

Fig. 2: Age evolution over time, where t_{i_k} and t'_{i_k} is the generation time and delivery time of k^{th} fresh packet, respectively.

$$\bar{A}^{peak} := \limsup_{K\to\infty} \mathbb{E}\left[\frac{1}{K}\sum_{k=1}^{K} A_k^{peak}\right], \tag{2}$$

where the expectation $\mathbb{E}[\cdot]$ is taken over generation and delivery times of the update packets.

The time each fresh packet spends in the system, from its generation to delivery, is represented as $X_k = t'_{i_k} - t_{i_k}$ for the k^{th} fresh packet. The inter-delivery time between successive fresh packets is denoted as $Y_k = t'_{i_k} - t'_{i_{k-1}}$. Fig. 2 shows a sample path of age evolution over time. Further, for the ergodic A_k^{peak}, we can express $\bar{A}^{peak}$ as:

$$\bar{A}^{peak} = \mathbb{E}[X_{k-1}] + \mathbb{E}[Y_k]. \tag{3}$$

We assume a preemptive server, whereby the transmitter can replace a packet in service with another waiting in the queue. In such cases, if the preempted packet is served again, the transmission of the packet starts anew. Let π denote a update policy and let Π denote the set of all possible update policies. Our objective is to design a policy that minimizes the expected weighted sum of age and reconstruction queue length:

$$\min_{\pi\in\Pi} J_\pi = \beta\bar{A}_\pi^{peak} + (1-\beta)\bar{Q}_\pi^r, \tag{4}$$

where $\bar{A}_\pi^{peak}$ and $\bar{Q}_\pi^r$ is the average peak age and the average reconstruction queue length, respectively, that are associated with update policy π, and $\beta \in [0, 1]$ is a weight parameter, which determines the relative importance of the fresh information and reconstruction cost.

III. Update Policies

In this section, we discuss different update policies for the transmission of packets in the system. We begin by examining two commonly used policies: FCFS and LCFS. Subsequently, we introduce a novel update policy called Head-Tail Randomization for balancing freshness and reconstruction queue length.

A. First-Come First-Serve and Last-Come First-Serve

As an extreme case, it can be observed that under FCFS policy, the reconstruction queue is always empty due to sequential transmission, while high traffic intensity (i.e., λ/μ approaching 1) leads to increased age. On the opposite end, LCFS policy with preemption benefits packet freshness, but compromises reconstruction queue performance as the packets are transmitted in a reverse-sequential manner.

Fig. 3 shows the average peak age $\bar{A}_\pi^{peak}$ and average reconstruction queue length $\bar{Q}_\pi^r$ under FCFS and LCFS, where the service rate μ is set to 1 and the traffic intensity $\rho = \lambda/\mu$ varies from 0 and 1. The results are averaged over 10^5 packets.

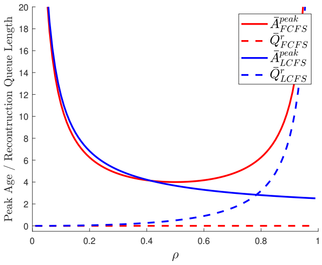

Fig. 3: The average peak age $\bar{A}_\pi^{peak}$ and reconstruction queue length $\bar{Q}_\pi^r$ under FCFS and LCFS policies.

As expected, at higher ρ values, FCFS results in a greater age increase compared to LCFS, while LCFS leads to poorer queue performance than FCFS. Note that, for lower ρ values, the performance gap between FCFS and LCFS is insignificant due to the infrequency of packet generation. This indicates neither policy alone effectively minimizes the weighted sum of average age and reconstruction queue length in (4).

B. Head-Tail Randomization Policy

In this section, we introduce a randomized update policy that chooses between LCFS and FCFS policies probabilistically to achieve a good trade-off between the two objectives of freshness and reconstruction.

Scheduling at the transmitter: Upon the generation of a new packet at t_i, the transmitter probabilistically chooses either the freshest (tail) or oldest (head) packet, with probabilities α and $1-\alpha$ respectively. The parameter α can be intentionally designed to optimize the objective stated in (4). Further, when a packet departs at t'_i, the head packet is chosen with probability 1.

Status update and reconstruction at the receiver: Under the proposed scheduling policy, the receiver can distinguish whether a received packet is the head packet or the tail packet. When a tail packet is received, indicating the most recent information, it leads to a reduction in the age as the receiver adjusts its state estimate. This tail packet is stored in the reconstruction queue until all previously generated packets have also been delivered, thereby enhancing the accuracy of reconstructing prior trajectory events.

On the other hand, when a head packet is delivered, it enables the receiver to reconstruct the source's trajectory more accurately. The head packet, combined with other enqueued packets in their generation order, provides a comprehensive view of the source's history. Note that the head packet could reduce the age. However, for tractability in theoretical analysis, it is assumed that the receiver only relies on the tail packet to monitor the recent status of the source, resulting in a reduction in age upon the tail packet delivery. We refer to the proposed policy as π_{R1}. Nonetheless, we note that in practical scenarios, allowing the receiver to utilize the head packets for monitoring the recent status of the source can potentially enhance the age performance further. We discuss it in Section III-D.

We now analyze the average peak age $\bar{A}_{\pi_{R1}}^{peak}(\alpha)$ and average reconstruction queue length $\bar{Q}_{\pi_{R1}}^r(\alpha)$ under the proposed randomized update policy π_{R1} with the tail probability α.

The average peak age $\bar{A}^{peak}_{\pi}$: We can calculate the average peak age $\bar{A}^{peak}_{\pi_{R1}}(\alpha)$ by determining the expected system time $\mathbb{E}[X_{k-1}]$ and the expected inter-update time $\mathbb{E}[Y_k]$ using (3). Further, since the age is decreased only when the tail packet is delivered under the π_{R1} policy, the variables X_k and Y_k correspond to the system time and inter-delivery time of the tail packets, respectively. In the following lemma, we provide the the average peak age $\bar{A}^{peak}_{\pi_{R1}}(\alpha)$ of the head-tail randomization policy π_{R1} given α, λ and μ where $\lambda < \mu$.

Lemma 3.1: Given α, λ and μ where $\lambda < \mu$, the average peak age $\bar{A}^{peak}_{\pi_{R1}}$ of the π_{R1} policy is given by

$$\bar{A}^{peak}_{\pi_{R1}}(\alpha) = \tfrac{1}{\lambda+\mu} + \tfrac{1}{\alpha}\left(\tfrac{1}{\lambda} + \tfrac{1}{\mu}\right). \tag{5}$$

The proof is in Appendix A.

The average reconstruction queue length $\bar{Q}^{r}_{\pi}$: In the π_{R1} policy, when a tail packet departs the server, it is the freshest in the queue. If packet i is a tail packet, the number of packets in the queue at its departure is denoted by L_i. Packet i waits in the reconstruction queue until all L_i preceding packets are delivered. This waiting time equals the delivery time for these L_i packets. With the arrival rate λ, we apply Little's law to assess the reconstruction queue performance under this policy.

Lemma 3.2: Given α, λ and μ where $\lambda < \mu$, the average reconstruction queue length $\bar{Q}^{r}_{\pi_{R1}}(\alpha)$ of the π_{R1} policy is given by

$$\bar{Q}^{r}_{\pi_{R1}}(\alpha) = \tfrac{\alpha}{\mu+(1-\alpha)\lambda}\tfrac{\lambda^2}{\mu-\lambda}. \tag{6}$$

The proof is in Appendix B. From Lemmas 3.1 and 3.2, we can now obtain the average cost $J_{\pi_{R1}}(\alpha)$ and further optimize the tail probability α.

Proposition 3.1: Given α, λ and μ where $\lambda < \mu$, the expected weighted sum $J_{\pi_{R1}}(\alpha)$ of age and reconstruction queue length of the π_{R1} policy is given by $J_{\pi_{R1}}(\alpha)$

$$= \beta\left(\tfrac{1}{\lambda+\mu} + \tfrac{1}{\alpha}\left(\tfrac{1}{\mu} + \tfrac{1}{\lambda}\right)\right) + (1-\beta)\tfrac{\alpha}{\mu+(1-\alpha)\lambda}\tfrac{\lambda^2}{\mu-\lambda}. \tag{7}$$

Let $\alpha_1^*(\beta, \lambda)$ denote the optimal tail probability α that minimizes the cost under the π_{R1} policy given the weight $\beta \in [0, 1]$ and $\lambda \in (0, \mu)$, which is convex in α. By solving the convex problem, we have

$$\alpha_1^*(\beta, \lambda) = \min\left\{1, \tfrac{\sqrt{\beta}(1+\mu/\lambda))}{\sqrt{\beta}+\sqrt{(1-\beta)/(1/\lambda-1/\mu)}}\right\}. \tag{8}$$

Note that when $\beta = 0$, the cost J_π matches the expected reconstruction queue length $\bar{Q}^{r}_{\pi}$. In this case, according to (8), we find that $\alpha_1^*(0, \lambda) = 0$. Consequently, the transmitter always uses head packets, resulting in $J_{\pi_{R1}} = 0$, which aligns with our expectations. In contrast, when $\beta = 1$, the cost J_π is equivalent to the expected peak age $\bar{A}^{peak}_{\pi}$. In this situation, we find that $\alpha_1^*(1, \lambda) = 1$. This implies that the freshest packet always takes priority, and if there is no fresh packet in the buffer, head packets are served until a new packet is generated.

C. Comparison with FCFS and LCFS

Section III-A highlights challenges in remote tracking systems using FCFS or LCFS policies for trajectory reconstruction. To tackle these, Section III-B introduces the π_{R1} policy, which better balances age and reconstruction queue length by alternating between fresh and old packet transmissions. Although not an optimal solution for minimizing J_π in (4), π_{R1} effectively addresses the trade-off between these factors. In the following corollary, we present key findings regarding the performance characteristics of different policies under high traffic intensity.

Corollary 3.1: For $\beta \in (0, 1)$, as $\rho \to 1$, the average costs of the FCFS and LCFS policies increase in the order of $1/(1-\rho)$:

$$J_{FCFS} = O\left(\tfrac{1}{1-\rho}\right) \quad \textit{and} \quad J_{LCFS} = O\left(\tfrac{1}{1-\rho}\right), \tag{9}$$

while the average cost of the π_{R1} policy with optimal tail probability α^* increases in the order of $1/\sqrt{1-\rho}$:

$$J_{\pi_{R1}}(\alpha^*) = O\left(\tfrac{1}{\sqrt{1-\rho}}\right). \tag{10}$$

As can be seen in (20), the reconstruction queue length increases in the order of $1/(1-\rho)$ and also from [16], the average peak age for the FCFS policy is given by $1/(1-\rho)\mu + 1/\lambda$. Thus, we have (9) for a given $\beta \in (0, 1)$. Further, we can obtain (10) by substituting (8) into (7). Corollary 3.1 highlights the performance characteristics of different policies as the traffic intensity ρ approaches 1. This indicates that the head-tail randomization policy achieves a better performance trade-off between freshness and reconstruction as the system approaches high traffic intensity.

D. Improved Head-Tail Randomization

We recall that the π_{R1} policy, focusing solely on tail packets for monitoring the recent status of the source for analytical tractability, compromises age performance by not considering fresh head packets. To address this limitation, we introduce the π_{R2} policy, which incorporates head-tail randomization updates, allowing the receiver to utilize both packet types for more comprehensive freshness assessment. With π_{R2}, the average peak age, $\bar{A}^{peak}_{\pi_{R2}}$, tends to be lower or equal to that of π_{R1}, $\bar{A}^{peak}_{\pi_{R1}}$. Notably, the average reconstruction queue length, $\bar{Q}^{r}_{\pi_{R1}}(\alpha)$, remains equal to $\bar{Q}^{r}_{\pi_{R2}}(\alpha)$ across all α values. Additionally, in high-traffic conditions ($\rho \to 1$), both policies converge to similar average peak ages, as detailed in the subsequent proposition.

Proposition 3.2: For given $\alpha \in (0, 1)$, we have

$$\bar{Q}^{r}_{\pi_{R2}}(\alpha) = \bar{Q}^{r}_{\pi_{R1}}(\alpha) \; \textit{and} \; \bar{A}^{peak}_{\pi_{R2}}(\alpha) \leq \bar{A}^{peak}_{\pi_{R1}}(\alpha), \tag{11}$$

for any given $\rho \in (0, 1)$, and further

$$\lim_{\rho\to 1} |\bar{A}^{peak}_{\pi_{R2}}(\alpha) - \bar{A}^{peak}_{\pi_{R1}}(\alpha)| = 0. \tag{12}$$

To establish the proposition, it is sufficient to show that the average time fraction during which the head packet is fresher than the packets held by the receiver decreases as $\rho \to 1$. We omit the proof due to lack of space.

IV. Numerical Results

In this section, we evaluate the proposed head-tail randomization policy π_{R1}. We first compare π_{R1} against the FCFS, LCFS, and π_{R2} policies. For π_{R2}, optimal α is determined by numerical search. Results are obtained from 5×10^5 packets.

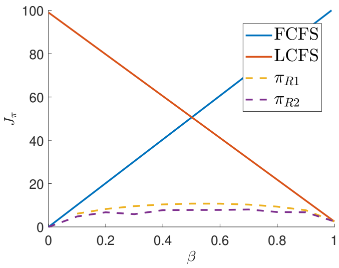

Fig. 4: Average cost J_π when $\mu = 1$ and $\lambda = 0.99$.

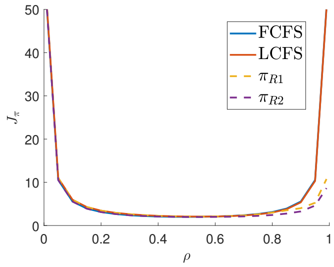

Fig. 5: Average cost J_π when $\mu = 1$ and $\beta = 0.5$.

Fig 4 illustrates the average cost of these four different polices for $\beta \in [0, 1]$ with $\mu = 1$ and $\lambda = 0.99$, showing π_{R1}'s superiority over FCFS and LCFS, especially at $\beta = 0.5$. Further, while the π_{R2} policy outperforms π_{R1} as predicted by Proposition 3.2, the gap in their performance is not substantial with a high traffic intensity of $\lambda/\mu = 0.99$.

Fig. 5 illustrates average costs for varying traffic intensities ($\rho = \lambda/\mu \in (0, 1)$) with $\mu = 1$ and $\beta = 0.5$. The figure reveals, as predicted by Corollary 3.1, that as $\rho \to 1$, the gap between the head-tail randomization policies (π_{R1} and π_{R2}) and FCFS/LCFS widens. Further, all policies exhibit increased costs as $\rho \to 0$ due to less frequent packet generation and thus higher age performance, leading to minimal performance differences in low-generation-rate scenarios.

V. Conclusion

In this paper, we investigated remote tracking systems with a single source and monitor, focusing on tracking the source's current state as well as past trajectory. We used reconstruction queue length to evaluate trajectory reconstruction performance and addressed the problem of minimizing the weighted sum of age and this length. Our findings revealed that neither FCFS nor LCFS policies alone effectively solves the minimization problem. To overcome this, we proposed a randomized policy that probabilistically selects between FCFS and LCFS policies and analyzed its performance. Our results showed that the proposed randomization update policy tackles the limitations of FCFS and LCFS policies and outperforms them. Notably, our proposed policy does not utilize state information such as current age or queue length. It will be interesting to explore open problems related to the design of state-dependent update policies or learning-based update policies when certain system parameters are unknown a priori.

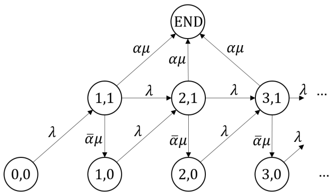

Fig. 6: A CTMC with state $(E_1(s), E_2(s))$ and END, where $\bar{\alpha} = 1 - \alpha$.

Appendix A
Proof of Lemma 3.1

We can calculate the average peak age $\bar{A}^{peak}_{\pi_{R1}}(\alpha)$ by evaluating the expected system time $\mathbb{E}[X_{k-1}]$ and the expected inter-update time $\mathbb{E}[Y_k]$. Note that, under the π_{R1} policy, the age is decreased only when a tail packet is delivered, making X_k and Y_k correspond to the tail packets' system time and inter-delivery time, respectively.

For inter-delivery time $\mathbb{E}[Y_k]$ of consecutive tail packets, let $E_1(s)$ denote the number of new packets generated by time s after the delivery of the $(k-1)^{th}$ tail packet. Let $E_2(s)$ denote the event that there exists a positive probability for the tail packet to be delivered at time $s \geq t'_{i_{k-1}}$. If a new packet at time s' leads to tail packet service with probability α, we have $E_2(s') = 1$, and 0 if the head packet is chosen with probabilty $1 - \alpha$. If a head packet is serviced, then $E_2(s) = 0$ until the next packet generation. The event END marks the k^{th} tail packet's delivery. Using these definitions, we form a continuous-time Markov chain (CTMC) with states $(E_1(s), E_2(s))$ for $s \geq t'_{i_{k-1}}$ and END as shown in Fig. 6.

Note that the holding time of each state $(a, 0)$ for $a \geq 0$ is equal to $1/\lambda$, while the holding time of each state $(a, 1)$ for $a \geq 1$ is equal to $1/(\lambda+\mu)$. Let us denote $H_{a,b}$ as the expected duration from state (a, b) to state END. Considering that the initial state at time $t'_{i_{k-1}}$ is $(0, 0)$, we can express the expected inter-delivery time of tail packets as $\mathbb{E}[Y_k] = H_{0,0}$. Using the Markov property, we can write the following equations:

$$H_{\text{END}} = 0, \; H_{a,0} = \tfrac{1}{\lambda} + H_{a+1,1}, \quad \text{for } a \geq 0, \tag{13}$$

$$H_{a,1} = \tfrac{1}{\lambda+\mu} + \tfrac{\lambda}{\lambda+\mu} H_{a+1,1} + \tfrac{(1-\alpha)\mu}{\lambda+\mu} H_{a+1,0} + \tfrac{\alpha\mu}{\lambda+\mu} H_{\text{END}},$$

for $a \geq 1$. By solving these equations, we can obtain the expected inter-delivery time as

$$\mathbb{E}[Y_k] = H_{0,0} = \tfrac{1}{\lambda} + \tfrac{1}{\alpha}\left(\tfrac{1}{\mu} + \tfrac{1}{\lambda}\right). \tag{14}$$

We now consider the expected system time $\mathbb{E}[X_{k-1}]$. According to the π_{R1} policy, the server initiates the service of a tail packet with a probability of α upon its generation. Further, for the tail packet to be successfully delivered to the receiver, it is imperative that no new packet generation occurs before its departure. Thus, we can express the expected system time as $\mathbb{E}[X_{k-1}] = \mathbb{E}[\text{Service time} \mid \text{departure before arrival}] \cdot \mathbb{P}(\text{dep. before arr.}) = \mathbb{E}[\text{Service time}]\mathbb{P}(\text{dep. before arr.})$, so

$$\mathbb{E}[X_{k-1}] = \tfrac{1}{\mu} \cdot \tfrac{\mu}{\lambda+\mu} = \tfrac{1}{\lambda+\mu}. \tag{15}$$

Therefore, from (14) and (15), we have

$$\bar{A}^{peak}_{\pi_{R1}}(\alpha) = \tfrac{1}{\lambda+\mu} + \tfrac{1}{\alpha}\left(\tfrac{1}{\mu} + \tfrac{1}{\lambda}\right). \tag{16}$$

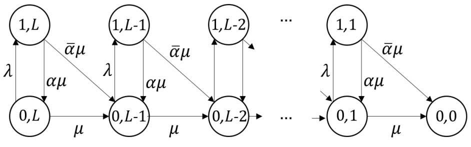

Fig. 7: A CTMC with state $(E_2(s), E_3(s))$, where $\bar{\alpha} = 1 - \alpha$.

Appendix B
Proof of Lemma 3.2

In the π_{R1} policy, when a tail packet departs, it is the freshest in the queue. We denote $L > 0$ as the number of packets in the queue at the tail packet's departure. This tail packet waits in the reconstruction queue until all L older packets are delivered, thus the waiting time for this packet in the reconstruction queue equals the time taken for all L preceding packets to be delivered. The expected waiting time for all L packets is denoted as $f_\alpha(L)$, and the event of a leaving packet being the tail packet is E_α. The average waiting time, $\bar{D}_{\pi_{R1}}(\alpha)$, given tail probability α, is:

$$\bar{D}_{\pi_{R1}}(\alpha) = \textstyle\sum_{l=1}^{\infty} \mathbb{P}(L = l)\mathbb{P}(E_\alpha) f_\alpha(L). \tag{17}$$

To analyze $f_\alpha(L)$, the expected duration for all L preceding packets to be delivered after a tail packet's transmission, we focus on two key events post the tail packet's departure at t'_{i_k}. First, the event $E_2(s)$ indicates the presence of a non-zero probability for the tail packet to be delivered to the receiver at time s with $E_2(t'_{i_k}) = 0$. Secondly, $E_3(s)$ tracks the number of older packets in the transmitter's queue at time s. By incorporating these elements, we can construct a CTMC characterized by states $(E_2(s), E_3(s))$ for $s \geq t'_{i_k}$ as in Fig. 7.

In this CTMC, the holding time for each state $(1, b)$, where $b \in \{1, ..., L\}$, is equal to $1/\mu$. For states $(0, b)$ within the same range, the holding time is $1/(\lambda + \mu)$. We denote $\bar{H}_{a,b}$ as the expected duration from state (a, b) to state $(0, 0)$, the latter representing the condition where all L older packets have been delivered, enabling the k^{th} tail packet to leave the queue. This implies that $f_\alpha(L) = \bar{H}_{0,L}$. Further, note that the transition time from each state $(0, b)$ to the next state $(0, b-1)$ is consistent for all $b \in \{1, ..., L\}$. We denote this uniform time as $\bar{H}_0$. Additionally, let $\bar{H}_1$ denote the expected duration from state $(0, b)$ to state $(1, b)$. Consequently, we can write $f_\alpha(L) = \bar{H}_{0,L} = L\bar{H}_0$. Using the Markov property, we can write the following equations:

$$\begin{aligned} \bar{H}_{END} = 0,\ \bar{H}_0 &= \tfrac{1}{\lambda+\mu} + \tfrac{\lambda}{\lambda+\mu}\bar{H}_1 + \tfrac{\mu}{\lambda+\mu}\bar{H}_{END}, \\ \bar{H}_1 &= \tfrac{1}{\mu} + \alpha\bar{H}_0 + (1-\alpha)\bar{H}_{END}, \end{aligned} \tag{18}$$

where END denotes the absorbing state, i.e., $(0, b-1)$. By solving these equations, we can obtain $\bar{H}_0 = \frac{1+\rho}{\mu+(1-\alpha)\lambda}$ and

$$f_\alpha(L) = \tfrac{(1+\rho)L}{\mu+(1-\alpha)\lambda}. \tag{19}$$

Next, we analyze $\mathbb{P}(E_\alpha)$, the probability of a departing packet being the tail packet. In a scenario where a new packet (packet A) is generated and chosen for service with probability α, two outcomes can occur: if a new packet arrives before packet A departs, packet A becomes a head packet; if packet A departs first, it is a tail packet. Thus, $\mathbb{P}(E_\alpha)$ combines the chance of packet A being selected and the chance of its departure preceding any new arrival, resulting in $\mathbb{P}(E_\alpha) = \frac{\alpha\mu}{\lambda+\mu}$.

Combining with (17) and (19), we have

$$\bar{D}_{\pi_{R1}}(\alpha) = \sum_{l=1}^{\infty} l\mathbb{P}(L = l)\mathbb{P}(E_\alpha) f_\alpha(1) = \mathbb{P}(E_\alpha) f_\alpha(1)\mathbb{E}[L].$$

Hence, the left is to obtain the expected number $\mathbb{E}[L]$ of packets in the transmitter's queue when a tail packet leaves the system, which is equal to the expected length of the transmitter's queue, i.e., $\mathbb{E}[L] = \frac{\lambda}{\mu-\lambda}$. Therefore, we have $\bar{D}_{\pi_{R1}}(\alpha) = \frac{\alpha}{\mu+(1-\alpha)\lambda}\frac{\lambda}{\mu-\lambda}$, and by Little's law, the average length of reconstruction queue is given by

$$\bar{Q}^r_{\pi_{R1}}(\alpha) = \tfrac{\alpha}{\mu+(1-\alpha)\lambda}\tfrac{\lambda^2}{\mu-\lambda}. \tag{20}$$

References

[1] L. Atzori, A. Iera, and G. Morabito, "The internet of things: A survey," *Computer networks*, vol. 54, no. 15, pp. 2787–2805, 2010.

[2] Y. Sun, E. Uysal-Biyikoglu, R. D. Yates, C. E. Koksal, and N. B. Shroff, "Update or wait: How to keep your data fresh," *IEEE Transactions on Information Theory*, vol. 63, no. 11, pp. 7492–7508, 2017.

[3] J. Pan, A. M. Bedewy, Y. Sun, and N. B. Shroff, "Age-optimal scheduling over hybrid channels," *IEEE Transactions on Mobile Computing*, 2022.

[4] I. Kadota, A. Sinha, and E. Modiano, "Optimizing age of information in wireless networks with throughput constraints," in *Proc. INFOCOM*, 2018.

[5] E. O. Gamgam and N. Akar, "Exact analytical model of age of information in multi-source status update systems with per-source queueing," *IEEE Internet of Things Journal*, vol. 9, no. 20, pp. 20 706–20 718, 2022.

[6] S. Zhang, H. Zhang, Z. Han, H. V. Poor, and L. Song, "Age of information in a cellular internet of uavs: Sensing and communication trade-off design," *IEEE Transactions on Wireless Communications*, vol. 19, no. 10, pp. 6578–6592, 2020.

[7] X. Qin, Y. Xia, H. Li, Z. Feng, and P. Zhang, "Distributed data collection in age-aware vehicular participatory sensing networks," *IEEE Internet of Things Journal*, vol. 8, no. 19, pp. 14 501–14 513, 2021.

[8] Y. Sun, Y. Polyanskiy, and E. Uysal-Biyikoglu, "Remote estimation of the wiener process over a channel with random delay," in *IEEE International Symposium on Information Theory*, 2017, pp. 321–325.

[9] T. Z. Ornee and Y. Sun, "Sampling for remote estimation through queues: Age of information and beyond," in *WiOpt*, 2019, pp. 1–8.

[10] S. Wu, X. Ren, S. Dey, and L. Shi, "Optimal scheduling of multiple sensors over shared channels with packet transmission constraint," *Automatica*, vol. 96, pp. 22–31, 2018.

[11] K. Chen, L. Ding, F. Yang, and L. Qian, "Mse minimized scheduling for multiple-source remote estimation with aoi constraints in iwsn," in *Proc. WCSP*, 2020, pp. 1119–1124.

[12] X. Zhang, M. M. Vasconcelos, W. Cui, and U. Mitra, "Distributed remote estimation over the collision channel with and without local communication," *IEEE Transactions on Control of Network Systems*, 2021.

[13] S. Kang, A. Eryilmaz, and N. B. Shroff, "Remote tracking of distributed dynamic sources over a random access channel with one-bit updates," in *WiOpt*, 2021, pp. 1–8.

[14] S. Kang, A. Eryilmaz, and C. Joo, "Comparison of decentralized and centralized update paradigms for remote tracking of distributed dynamic sources," in *Proc. INFOCOM*, 2021.

[15] E. Schulz, M. Speekenbrink, and A. Krause, "A tutorial on gaussian process regression: Modelling, exploring, and exploiting functions," *Journal of Mathematical Psychology*, vol. 85, pp. 1–16, 2018.

[16] S. Kaul, R. Yates, and M. Gruteser, "Real-time status: How often should one update?" in *IEEE INFOCOM*, 2012, pp. 2731–2735.