Distribution-oblivious Online Algorithms for Age-of-Information Penalty Minimization

Cho-Hsin Tsai and Chih-Chun Wang, *Senior Member, IEEE*

***Abstract*—The ever-increasing needs of supporting real-time applications have spurred new studies on minimizing Age-of-Information (AoI), a novel metric characterizing the data freshness of the system. This work studies the single-queue information update system and strengthens the seminal results of Sun *et al.* on the following fronts: (i) When designing the optimal offline schemes with full knowledge of the delay distributions, a new *fixed-point-based* method is proposed with *quadratic convergence rate*, an order-of-magnitude improvement over the state-of-the-art; (ii) When the distributional knowledge is unavailable (which is the norm in practice), two new low-complexity online algorithms are proposed, which provably attain the optimal average AoI penalty; and (iii) the online schemes also admit a modular architecture, which allows the designer to *upgrade* certain components to handle additional practical challenges. Two such upgrades are proposed for the situations: (iii.1) The AoI penalty function is also unknown and must be estimated on the fly, and (iii.2) the unknown delay distribution is Markovian instead of i.i.d. The performance of our schemes is either provably optimal or within 3% of the omniscient optimal offline solutions in all simulation scenarios.**

***Index Terms*—Age-of-information, online algorithm, fixed-point equation, stochastic approximation algorithm.**

I. Introduction

Thanks to the accelerating growth of networked systems in the past decades, the capability of providing real-time status updates has been the cornerstone of many important practical systems. Examples include remote health monitoring, GPS location tracking and closed-loop drone control. Recent development of the Internet of Things (IoT) also promises real-time communication between numerous devices [2].

Since stale data is often of less value, it is crucial to optimize the *data freshness* of the system. An elementary approach is to transmit as many updates as possible. This, however, may clog the network and consume excessive energy. Recently, Age-of-Information (AoI) was introduced to characterize the level of information freshness [3], which has since been the foundation of many studies on data freshness control.

Early AoI minimization works studied the model where update packets arrive at the destination according to specific stochastic processes. [4], [5] studied the *generate-at-will* model and showed that to minimize the average AoI, the source node often has to *wait before sending the next packet* even when the channel/queue is currently idle. [6], [7] unified the freshness control [4] and remote estimation settings of [8] under a new *remote control* setting and derived the optimal joint source-&-destination policy. [9] found the optimal scheduling policy of a joint network cost and AoI minimization problem when multiple independent queues may share a common network cost constraint.

This work revisits and significantly strengthens the existing results [4], [5] with the following contributions: (i) When designing the optimal offline schemes with full knowledge of the delay distributions, all existing results [4]–[9] used a bisection search to find the optimal policy, which exhibits linear convergence rate. In contrast, *we propose a fixed-point-based method of computing the optimal policy under any arbitrarily given AoI penalty function, which exhibits quadratic convergence rate, an order-of-magnitude improvement over the state of the art.*

(ii) In most prior works [4]–[6], [8]–[12], the knowledge of delay distribution is required before one can numerically find the optimal waiting policy. In practice it may be difficult to know the delay distribution *a priori* since each sample (i.e., packet transmission) takes a full round-trip-time to complete and one may need many samples to accurately estimate the probability density function, which can be exceedingly time consuming. Furthermore, the delay distribution is constantly subject to network topology changes and traffic fluctuations [13], which further complicate the task of learning the distribution. To address these issues, *this work derives two new low-complexity online algorithms for arbitrarily given AoI penalty functions, and they provably converge to the optimum without knowing the delay distribution*, a result that could have substantial impact on practical protocol designs.

(iii) The new online schemes admit a modular architecture, which allows the designer to *upgrade* certain components to tackle additional practical challenges. Two such upgrades are proposed for the following two useful situations: Situation #1: Existing works [4], [5] assumed that the AoI penalty function $\gamma(\cdot)$ is known *a priori*. However, in practice, transmission decision is often made at the source but the penalty is often incurred at the destination. Therefore, it could be difficult for the source to know the AoI penalty function $\gamma(\cdot)$ a priori. A real-life analogy is that a vendor s (stands for the source) understands that less fresh produce will make his/her customer d (stands for the destination) unhappy but s may not know how unhappy d would be until d eventually receives the (not-so-fresh) produce. Furthermore, even d may not know how he/she will react to the stale produce until he/she actually receives the

This work was supported in parts by NSF under Grant CCF-1816013, Grant CCF-2008527 and Grant CNS-2107363, and also by National Spectrum Consortium (NSC) under grant W15QKN-15-9-1004.

Part of this work [1] was presented in 2020 IEEE International Symposium on Information Theory (ISIT) [DOI: 10.1109/ISIT44484.2020.9174306]. The supplementary material of this work is available on IEEE Xplore.

Cho-Hsin Tsai was with Purdue ECE, West Lafayette, IN 47907, USA. He is now with MathWorks Inc., Natick, MA 01760, USA. Chih-Chun Wang is with Purdue ECE (e-mail: chohsintsai@gmail.com; chihw@purdue.edu).

delivery. As a result, the AoI penalty function is not known in advance and our goal is to design a near-optimal online scheme with zero knowledge of either the delay distribution or the $\gamma(\cdot)$ penalty function. In this work, we design such a scheme by leveraging the *monotonic regression* method to estimate any arbitrary, non-linear $\gamma(\cdot)$, which further broadens the applicability.

Situation #2: The online schemes in (ii) provably converge to optimality for any unknown i.i.d. delay. In practice, the delay process may exhibit some memory/Markov behavior. We have devised a more versatile scheme based on K-nearest-neighbors (KNN). In our extensive simulation, even under the most challenging setting of arbitrary unknown Markov delay distribution and zero knowledge of $\gamma(\cdot)$, the performance of the resulting scheme is always within 3% of the omniscient offline optimal solution.

The rest of the paper is organized as follows. In Sec. II, we make detailed comparison to existing works. In Sec. III, we present our system model and problem formulation. In Sec. IV, we derive the analytical results for the optimal offline policy. Sec. V describes two online algorithms that are provably convergent to the optimum under any unknown i.i.d. delay distributions. Some practical issues are addressed in Secs. VI (under i.i.d. delay setting) and VII (under Markov delay setting). Numerical results are reported in Sec. VIII, and we conclude our work in Sec. IX. Due to the space limit, most of the proofs will be provided in the appendices of the supplementary material, which is available on IEEE Xplore.

II. Related Works

One approach of handling unknown delay distribution is to apply reinforcement learning (RL) [14], [15]. However, none of these RL-based AoI minimization schemes has a provable optimality guarantee and can be strictly suboptimal in many cases. For example, while exhibiting some promising performance, the RL scheme in [15] is unable to converge to the optimal scheme in any of the experiments in [15]. In contrast, this work proposes two adaptive schemes that converge to optimality both analytically and empirically.

Additionally, some previous works proposed online algorithms with bounded regret or provable performance [16], [17]. However, they all studied the simplest *linear* age penalty function or assumed the transmission delay is *deterministic*. For example, a provably optimal online algorithm was derived in [17] with the focus exclusively on linear AoI penalty function. This works allows for *non-linear* AoI penalty function and *random* transmission delay simultaneously. In terms of AoI regret minimization, since our online schemes converge to optimality, they achieve sub-linearly growing regret, the same performance as in [17]. There are other related works on AoI regret analysis, but the focus is on multiple sources [16] or multiple channels [18] with *deterministic* delay, which is very different from the two-way *random* delay setting in this work.

III. Model and Formulation

A. System Model with Two-way Delay

Consider the system in Fig. 1, which comprises a source, a destination, a forward source-to-destination (s2d) channel and a backward destination-to-source (d2s) channel. We assume the following ACK-based model: At any time instant $t \in \mathbb{R}_+$, the source can generate a (status) update packet and transmit it to the destination, the *generate-at-will* model [4], [5], [19]. When the destination receives the update packet, an ACK is transmitted back to the source immediately. Once the source receives the ACK, then it can either transmit the next update packet immediately or wait for an arbitrary (but finite) amount of time.[1] After the next transmission, it again waits for ACK. The process repeats itself indefinitely.

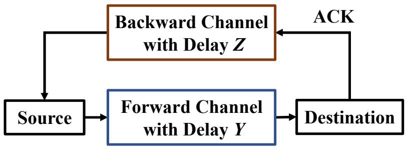


Fig. 1: Our system model with two-way delay.

Both the s2d and d2s channels incur some random delay.[2] We assume all packets are time stamped and describe the detailed system evolution as follows.

Time sequences: The system consists of three discrete-indexed real-valued non-negative random processes X_i, Y_i, and Z_i, for all $i \geq 0$. X_i is the waiting time of the i-th update packet at the source; Y_i (resp. Z_i) is the random delay for the i-th use of the s2d (resp. d2s) channel.

The instant when the i-th waiting time is over is denoted by S_i. That is, at time S_i, the i-th packet is generated and immediately transmitted. It is delivered to the destination at time D_i. The source will receive its ACK at time A_i. The values of (S_i, D_i, A_i) refer to the absolute time instants while the values of (X_i, Y_i, Z_i) represent the lengths of the intervals. They are related by the following equations: Initialize $A_0 = X_0 = Y_0 = Z_0 = 0$. For all $i \geq 1$, we have $S_i = A_{i-1} + X_i$, $D_i = S_i + Y_i$, and $A_i = D_i + Z_i$. We call the time interval $[A_{i-1}, A_i)$ as the *i-th round*, which consists of the i-th waiting time X_i at the source, the i-th forward delay Y_i and backward delay Z_i. See Fig. 2. The AoI $\Delta(t)$ is defined by

$$\Delta(t) \triangleq t - \max\{S_i : D_i \leq t\}. \tag{1}$$

The *AoI penalty function* $\gamma(\cdot) : [0, \infty) \to [0, \infty)$ quantifies the cost of stale data. Three popular choices are: (i) linear $\gamma_{\text{lin}}(\Delta) = \Delta$ [4]; (ii) exponential $\gamma_{\text{exp}}(\Delta) = e^{a\Delta} - 1$ for some constant $a > 0$ [5]; and (iii) quadratic $\gamma_{\text{qdr}}(\Delta) = \Delta^2$ [4]. Our results hold for any choice of $\gamma(\cdot)$ satisfying the technical assumption described in the next paragraph.

Technical assumptions: We assume (i) there exist finite y_{max}, z_{max}, and $yz_{\text{min}} > 0$ such that $\mathbb{P}(Y_i \leq y_{\text{max}}) = \mathbb{P}(Z_i \leq z_{\text{max}}) = \mathbb{P}(Y_i + Z_i \geq yz_{\text{min}}) = 1$; (ii) (Y_i, Z_i) can be of arbitrary joint distribution $\mathbb{P}_{YZ}$ but the vector random process $\{(Y_i, Z_i) : i \geq 1\}$ is stationary, Markov and ergodic; (iii) The AoI penalty

[1] In this work, *only after receiving the ACK* can the source transmit the next packet, a restrictive assumption that is widely used in the literature [1], [4]–[6], [9]. For comparison, [20] removed this assumption and studied the benefits of preemptively sending packets *before* ACK has arrived. Provably near-optimal schemes were developed in [20] for general two-way delay settings.

[2] If we assume the d2s delay is zero with probability one, then the setting is identical that of [4]. The consideration of random d2s delay is to provide additional flexibility if needed.

function $\gamma(\cdot) : [0,\infty) \to [0,\infty)$ is a continuous and *strictly increasing* function satisfying $\gamma(0) = 0$.

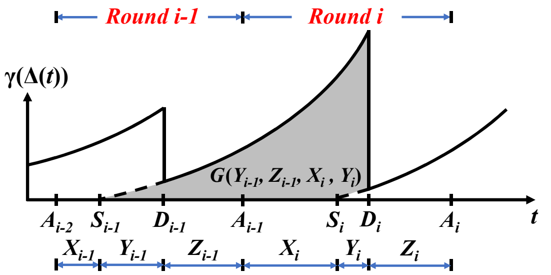


Fig. 2: Evolution of the AoI penalty function $\gamma(\Delta(t))$.

B. The Objective

Our goal is to minimize the long-term average AoI penalty:

$$\beta^* \triangleq \inf_{\{X_i\}} \limsup_{T\to\infty} \frac{1}{T}\int_0^T \mathbb{E}\left\{\gamma(\Delta(t))\right\} dt. \tag{2}$$

To simplify (2), we define two deterministic functions:

$$G(y', z', x, y) \triangleq \int_0^{y'+z'+x+y} \gamma(t)dt - \int_0^y \gamma(t)dt \tag{3}$$

$$G_1(y', z', x) \triangleq \mathbb{E}\{G(y', z', x, Y_i)|Y_{i-1} = y', Z_{i-1} = z'\} \tag{4}$$

where $G_1(y', z', x)$ is the conditional expectation of $G(y', z', x, Y_i)$ over Y_i given $Y_{i-1} = y'$ and $Z_{i-1} = z'$. The intuition behind (3) is that the shaded area in Fig. 2 is characterized by $G(Y_{i-1}, Z_{i-1}, X_i, Y_i)$. By noticing that the overall area underneath $\gamma(\Delta(t))$ can be decomposed as a summation of smaller sub-areas with shapes similar to the shaded area $G(Y_{i-1}, Z_{i-1}, X_i, Y_i)$ in Fig. 2, the optimization problem in (2) can be rewritten as

$$\beta^* = \inf_{\{X_i\}} \limsup_{n\to\infty} \frac{\displaystyle\sum_{i=1}^{n} \mathbb{E}\left\{G(Y_{i-1}, Z_{i-1}, X_i, Y_i)\right\}}{\displaystyle\sum_{i=1}^{n} \mathbb{E}\left\{Y_{i-1} + Z_{i-1} + X_i\right\}}. \tag{5}$$

Since (5) is a Markov decision problem with stationary, Markov and ergodic $\{(Y_i, Z_i)\}$, it suffices to find the optimal policy for the *single-round* optimization problem instead (see [4], [5] for the detailed derivation). The optimization problem (5) can thus be simplified as

$$\beta^* = \inf_{X_i} \frac{\mathbb{E}\left\{G_1(Y_{i-1}, Z_{i-1}, X_i)\right\}}{\mathbb{E}\left\{Y_{i-1} + Z_{i-1} + X_i\right\}} \tag{6}$$

where the numerator of (6) follows from (4).

We conclude this subsection by defining a constant β_{ZW} that would be useful for subsequent discussion.

$$\beta_{\text{ZW}} \triangleq \frac{\mathbb{E}\left\{G_1(Y_{i-1}, Z_{i-1}, 0)\right\}}{\mathbb{E}\left\{Y_{i-1} + Z_{i-1} + 0\right\}}. \tag{7}$$

That is, β_{ZW} is the objective function value in (6) when evaluated using a Zero-Wait policy.

Lemma 1: We must have $\beta_{\text{ZW}} < \infty$.

Proof: By the assumption about $y_{\max}$, $z_{\max}$, $yz_{\min}$, and the monotonicity of $\gamma(\cdot)$, we have $\mathbb{E}\{Y_{i-1}\} + \mathbb{E}\{Z_{i-1}\} \geq yz_{\min} > 0$ and $\mathbb{E}\{G_1(Y_{i-1}, Z_{i-1}, 0)\} \leq \mathbb{E}\{G_1(y_{\max}, z_{\max}, 0)\} < \infty$. As a result, $\beta_{\text{ZW}} < \infty$. ■

IV. Analytical Results

A. A Hitting-time-based Policy

At time A_{i-1}, the source has the knowledge of the past delays Y_{i-1} and Z_{i-1} since all packets are time stamped. As a result, we can write any waiting time rule $X_i = \phi(Y_{i-1}, Z_{i-1})$ as a function of (Y_{i-1}, Z_{i-1}).

Definition 1: We say a scheme $\mathcal{S}$ is *of finite expected duration* (FED) if $\mathbb{E}\{\phi(Y_{i-1}, Z_{i-1})\} < \infty$.

Practically speaking, it is crucial that the waiting time of each packet transmission has finite expectation. We thus limit the domain of the optimization problem of (6) to FED schemes only and ignore schemes that are not of FED.

Once we specify a waiting time function $\phi(Y_{i-1}, Z_{i-1})$, the resulting[3] averaged AoI penalty, not necessarily the minimum one, becomes

$$\text{Avg. AoI Penalty: } \frac{\mathbb{E}\left\{G_1\left(Y_{i-1}, Z_{i-1}, \phi(Y_{i-1}, Z_{i-1})\right)\right\}}{\mathbb{E}\left\{Y_{i-1} + Z_{i-1} + \phi(Y_{i-1}, Z_{i-1})\right\}}. \tag{8}$$

For any FED scheme $\mathcal{S}$, we denote its average AoI penalty by $\beta_{\mathcal{S}}$, which is evaluated by (8).

Define

$$\beta_{\text{UB}} \triangleq \lim_{t\to\infty} \gamma(t) \tag{9}$$

Note that the constant β_{UB} can be either a finite constant $0 < \beta_{\text{UB}} < \infty$ or infinity $\beta_{\text{UB}} = \infty$ if $\gamma(\cdot)$ grows unbounded. We then have the following lemma.

Lemma 2: For any arbitrary FED scheme $\mathcal{S}$ with average AoI penalty $\beta_{\mathcal{S}}$, if $\beta_{\mathcal{S}} < \infty$, then $\beta_{\mathcal{S}} \in [0, \beta_{\text{UB}})$.

Proof: See Appendix A of the supplementary material. ■

The intuition of Lemma 2 is that β_{UB} is a strict upper bound for the performance of any reasonable scheme $\mathcal{S}$ (excluding those poorly designed schemes having infinite average AoI penalty $\beta_{\mathcal{S}} = \infty$).

Note that since zero-wait is a FED policy, by combining Lemmas 1 and 2, we have

$$0 \leq \beta^* \leq \beta_{\text{ZW}} < \beta_{\text{UB}} \leq \infty. \tag{10}$$

We now describe a special scheme Γ_β. For any given $\beta \in [0, \beta_{\text{UB}})$, scheme Γ_β has the following special decision rule:

$$\begin{aligned} X_i &= \phi_{\Gamma,\beta}(Y_{i-1}, Z_{i-1}) \\ &\triangleq \inf\left\{t > 0 : \frac{d}{dt} G_1(Y_{i-1}, Z_{i-1}, t) > \beta\right\}. \end{aligned} \tag{11}$$

By (4), $G_1(Y_{i-1}, Z_{i-1}, t)$ is the expected AoI penalty (the shaded area in Fig. 2) if the i-th waiting time is $X_i = t$. Therefore, the decision rule $\phi_{\Gamma,\beta}$ in (11) essentially chooses the hitting time for which the *growth rate*[4] of the expected AoI penalty $G_1(Y_{i-1}, Z_{i-1}, t)$ first hits the threshold β.

[3] The scheduling rule ϕ can be deterministic or randomized. In case of the latter, the expectation in (8) takes the average over the randomness in ϕ.

[4] As will be shown in Lemma 3, for any given (y', z'), $G_1(y', z', t)$ is differentiable with respect to t.

An important remark is that *the input parameter β of the above scheme Γ_β must be within* $[0, \beta_{UB})$. The reason is due to the following lemmas.

Lemma 3: For any given $y', z', t < \infty$, we have

$$\frac{d}{dt}G_1(y', z', t) \\ = \mathbb{E}\left\{\gamma(y' + z' + t + Y_i)|Y_{i-1} = y', Z_{i-1} = z'\right\}. \quad (12)$$

Proof: See Appendix B of the supplementary material. ■

Lemma 4: For any arbitrary $\beta \in [0, \beta_{\text{UB}})$, there exists a $t_{\text{UB}} < \infty$ such that $\frac{d}{dt}G_1(y', z', t_{\text{UB}}) > \beta$ for all y', z'. Conversely, for any $\beta \geq \beta_{\text{UB}}$, $\frac{d}{dt}G_1(y', z', t) \leq \beta$ for all finite $t, y', z' < \infty$ values. See Appendix C of the supplementary material for the proof.

Lemma 4 implies that for any $\beta \in [0, \beta_{\text{UB}})$, the X_i value computed by (11) satisfies $X_i \leq t_{\text{UB}} < \infty$ almost surely, and the resulting Γ_β is thus a FED scheme. On the other hand, for any $\beta \geq \beta_{\text{UB}}$, the second half of Lemma 4 implies that the X_i value computed by (11) is always infinite. The resulting scheme thus has $\mathbb{P}(X_i = \infty) = 1$ and is catastrophic to the system. As a result, every time we describe/use the Γ_β scheme, it is critical to ensure the input parameter satisfying $\beta \in [0, \beta_{\text{UB}})$.

For this scheme Γ_β, we use $f_\Gamma(\beta)$ to denote its average AoI penalty, which can be computed by substituting the ϕ in (8) with the $\phi_{\Gamma,\beta}$ in (11). The input argument "(β)" highlights the fact that the average AoI penalty of the decision rule $\phi_{\Gamma,\beta}$ is a function of the hitting time threshold β.

Proposition 1: For any FED scheme $\mathcal{S}$ with scheduling rule $\phi_{\mathcal{S}}$ and the corresponding average AoI penalty $\beta_{\mathcal{S}} < \infty$, the following inequality must hold: $f_\Gamma(\beta_{\mathcal{S}}) \leq \beta_{\mathcal{S}}$.

The physical interpretation of this proposition is as follows. For scheme $\mathcal{S}$ that satisfies $\beta_{\mathcal{S}} < \infty$, its average AoI penalty must also satisfy $\beta_{\mathcal{S}} \in [0, \beta_{\text{UB}})$ by Lemma 2. Since the new scheme Γ_β in (11) can take any arbitrary $\beta \in [0, \beta_{\text{UB}})$ as input, we can use $\beta_{\mathcal{S}}$ as the hitting time threshold in (11). Then $f_\Gamma(\beta_{\mathcal{S}})$, the AoI penalty of the new scheme $\Gamma_{\beta_{\mathcal{S}}}$, will be no worse than the average AoI penalty $\beta_{\mathcal{S}}$ of the original scheme $\mathcal{S}$.

Proof: We provide high-level sketches. The details are relegated to Appendix D of the supplementary material.

For schemes $\mathcal{S}$ and $\Gamma_{\beta_{\mathcal{S}}}$, recall that $\phi_{\mathcal{S}}(Y_{i-1}, Z_{i-1})$ and $\phi_{\Gamma,\beta_{\mathcal{S}}}(Y_{i-1}, Z_{i-1})$ are the waiting times for schemes $\mathcal{S}$ and $\Gamma_{\beta_{\mathcal{S}}}$, respectively. For simplicity, we use $\phi_{\mathcal{S}}$ and $\phi_{\Gamma,\beta_{\mathcal{S}}}$ as shorthand by dropping the input arguments (Y_{i-1}, Z_{i-1}).

Suppose we are in the event of $\phi_{\Gamma,\beta_{\mathcal{S}}} \leq \phi_{\mathcal{S}}$, i.e., the scheme $\Gamma_{\beta_{\mathcal{S}}}$ sends the i-th update earlier than the scheme $\mathcal{S}$. *During the interval* $(\phi_{\Gamma,\beta_{\mathcal{S}}}, \phi_{\mathcal{S}}]$*, the growth rate of* $G_1(Y_{i-1}, Z_{i-1}, t)$ *is strictly higher than* $\beta_{\mathcal{S}}$. The reason is as follows. By the definition of $\phi_{\Gamma,\beta_{\mathcal{S}}}$ in (11), the growth rate of $G_1(Y_{i-1}, Z_{i-1}, t)$ at time $t = \phi_{\Gamma,\beta_{\mathcal{S}}}$ is greater than or equal to $\beta_{\mathcal{S}}$. Since the growth rate of $G_1(Y_{i-1}, Z_{i-1}, t)$ is strictly increasing (due to strictly increasing $\gamma(\cdot)$ and by Lemma 3), the growth rate of $G_1(Y_{i-1}, Z_{i-1}, t)$ is strictly larger than $\beta_{\mathcal{S}}$ during $(\phi_{\Gamma,\beta_{\mathcal{S}}}, \phi_{\mathcal{S}}]$. Compared to the original scheme $\mathcal{S}$, the new scheme $\Gamma_{\beta_{\mathcal{S}}}$ avoids "higher-than-$\beta_{\mathcal{S}}$" AoI penalty accumulation rates during the interval $(\phi_{\Gamma,\beta_{\mathcal{S}}}, \phi_{\mathcal{S}}]$, which in turn helps make its average AoI penalty $f_\Gamma(\beta_{\mathcal{S}})$ smaller than the benchmark $\beta_{\mathcal{S}}$.

Similarly, in the event of $0 \leq \phi_{\mathcal{S}} < \phi_{\Gamma,\beta_{\mathcal{S}}}$, during the interval $(\phi_{\mathcal{S}}, \phi_{\Gamma,\beta_{\mathcal{S}}}]$, the new scheme $\Gamma_{\beta_{\mathcal{S}}}$ will experience "no-higher-than-$\beta_{\mathcal{S}}$" AoI penalty accumulation rates since the growth rate of $G_1(Y_{i-1}, Z_{i-1}, t)$ has not hit $\beta_{\mathcal{S}}$ yet during $t \in (\phi_{\mathcal{S}}, \phi_{\Gamma,\beta_{\mathcal{S}}}]$, which again helps make $f_\Gamma(\beta_{\mathcal{S}})$ lower than $\beta_{\mathcal{S}}$.

Since in either case the average AoI penalty of $\Gamma_{\beta_{\mathcal{S}}}$ has improved over the benchmark $\beta_{\mathcal{S}}$, we have proven Proposition 1. ■

Recall that β^* is the minimum of (6). Since $\beta^* \in [0, \beta_{\text{UB}})$ by (10), Proposition 1 implies $\beta^* \geq f_\Gamma(\beta^*)$. Since Γ_{β^*} is yet another scheme, (6) implies $\beta^* \leq f_\Gamma(\beta^*)$. Jointly we have

Corollary 1: The minimum average AoI penalty value β^* is a root of the fixed-point equation

$$\beta = f_\Gamma(\beta) \quad (13)$$

over the domain $[0, \beta_{\text{UB}})$.

One can complement this corollary by the following Lemma.

Lemma 5: For any given penalty function $\gamma(\cdot)$, the equation $\beta = f_\Gamma(\beta)$ has a unique root in the domain $\beta \in [0, \beta_{\text{UB}})$.

The proof of Lemma 5 is relegated to Appendix E of the supplementary material. Jointly Corollary 1 and Lemma 5 show that

Proposition 2: β^* can be found by solving the fixed-point equation (13) over the domain $[0, \beta_{\text{UB}})$.

Moreover, if we know the value of β^*, then we can obtain the optimal policy by plugging β^* into the hitting time rule $\phi_{\Gamma,\beta}(\cdot,\cdot)$ in (11). Namely, the fixed-point equation not only finds the β^* but also finds a β^*-attaining optimal policy.

Remark 1: Proposition 2 is of similar form to [4, Theorem 3] and [5, Theorem 1]. However, the way we derive Proposition 2 is new. In [4], [5], the authors first defined the corresponding Lagrangian, then reformulated and solved it as a convex optimization problem, and finally showed that it admits no duality gap. In contrast, we first prove an intuitive result in Proposition 1 and the optimality conditions then follow suit naturally.

Remark 2: The function $f_\Gamma(\beta)$ can be computed easily by (3), (4), (8), (11), together with the complete knowledge of distribution $\mathbb{P}_{Y_{i-1},Z_{i-1}}$.

Remark 3: Since β for the scheme Γ_β must satisfy $\beta \in [0, \beta_{\text{UB}})$, the corresponding average penalty value $f_\Gamma(\beta)$ is defined only over the domain $[0, \beta_{\text{UB}})$. It is possible to extend the domain of $f_\Gamma(\beta)$ beyond $[0, \beta_{\text{UB}})$ by defining $f_\Gamma(\beta_{\text{UB}}) \triangleq \lim_{\beta \to \beta_{\text{UB}}} f_\Gamma(\beta)$. Under this extended domain $[0, \beta_{\text{UB}}]$, it is possible to have another fixed point $\beta_{\text{UB}} = f_\Gamma(\beta_{\text{UB}})$ as observed in [21]. At the same time, as explained in Lemma 4, any $\beta \geq \beta_{\text{UB}}$ will lead to schemes with $\mathbb{P}(X_i = \infty) = 1$ and such extended domain $[0, \beta_{\text{UB}}]$ is thus considered practically irrelevant.

B. Fast Fixed-point Iteration for Computing β^*

We now present a new way of computing β^* using (13).

Proposition 3: Set $\beta_0 = 0$ and iteratively compute $\beta_i = f_\Gamma(\beta_{i-1})$ for all $i = 1, 2, 3, \cdots$. The resulting sequence

$\{\beta_i : i \geq 1\}$ is non-increasing and converges to the optimal β^*. Furthermore, if we also assume $f_\Gamma(\beta)$ is doubly continuously differentiable in an open neighborhood of β^*,[5] then the convergence speed of this iterative computation is quadratic.

Proof: We first note that our function $f_\Gamma(\beta)$ is defined only over the domain $[0, \beta_{\text{UB}})$. As a result, we first need to prove that $\beta_i \in [0, \beta_{\text{UB}})$ for all $i \geq 1$. Since $\beta_0 = 0 \in [0, \beta_{\text{UB}})$, the corresponding scheme $X_i = \phi_{\Gamma,\beta}(Y_{i-1}, Z_{i-1})$ in (11) always leads to $X_i = 0$, the zero-wait policy. As a result, $\beta_1 = f_\Gamma(0) = \beta_{\text{ZW}}$. By (10), we have $\beta_1 \in [0, \beta_{\text{UB}})$.

We now prove that $\beta_{i+1} \leq \beta_i \leq \beta_1 < \beta_{\text{UB}}$ for all $i \geq 1$. For any $i \geq 1$, since $\beta_i = f_\Gamma(\beta_{i-1})$, the value of β_i is the average AoI penalty for the scheme $\Gamma_{\beta_{i-1}}$. If we temporarily call the scheme $\Gamma_{\beta_{i-1}}$ as Scheme $\mathcal{S}$, then we can apply Proposition 1 and obtain

$$f_\Gamma(\beta_i) \leq \beta_i. \tag{14}$$

The sequence $\{\beta_i : i \geq 1\}$ is thus non-increasing. Since $\beta_i \geq \beta^*$ for all $i \geq 1$, the sequence converges and we also have $\lim_{i\to\infty} \beta_i \in [0, \beta_{\text{UB}})$.

Since $\lim_{i\to\infty} \beta_i$ must be a root of $\beta = f_\Gamma(\beta)$, Lemma 5 implies $\lim_{i\to\infty} \beta_i = \beta^*$.

We now establish the quadratic convergence by proving the following inequality for all $i \geq i_0$, where i_0 is the first time β_i enters the neighborhood of β^* for which $f_\Gamma(\beta)$ is doubly differentiable.

Applying Taylor's expansion to $f_\Gamma(\beta)$ near β^*, we have:

$$\begin{aligned}&\beta_{i+1} - \beta^* = f_\Gamma(\beta_i) - \beta^* \\ &= \left(f_\Gamma(\beta^*) + (\beta_i - \beta^*) f'_\Gamma(\beta^*) + \frac{f''_\Gamma(z_i)}{2}(\beta_i - \beta^*)^2 \right) - \beta^*\end{aligned}$$

for some $z_i \in [\beta^*, \beta_i]$. Note that since $f_\Gamma(\beta)$ is doubly continuous in an open neighborhood containing β^* and since β^* minimizes $f_\Gamma(\beta)$, we must have $f'_\Gamma(\beta^*) = 0$. Then, by (i) $f_\Gamma(\beta^*) = \beta^*$ and (ii) $f'_\Gamma(\beta^*) = 0$ we have

$$\beta_{i+1} - \beta^* = \frac{f''_\Gamma(z_i)}{2}(\beta_i - \beta^*)^2. \tag{15}$$

Since $z_i \in [\beta^*, \beta_i] \subseteq [0, \beta_1]$, (15) implies

$$|\beta_{i+1} - \beta^*| \leq \left(\max_{z \in [\beta^*, \beta_1]} \frac{|f''_\Gamma(z)|}{2} \right) \cdot |\beta_i - \beta^*|^2, \ \forall i \geq i_0 \tag{16}$$

We can further relax the condition $i \geq i_0$ by noting that

$$\begin{aligned}&|\beta_{i+1} - \beta^*| \\ &\leq \max\left(\max_{z \in [\beta^*, \beta_1]} \frac{|f''_\Gamma(z)|}{2}, \frac{|\beta_1 - \beta^*|}{(\beta_{i_0-1} - \beta^*)^2} \right) \cdot |\beta_i - \beta^*|^2, \\ &\quad \forall i \geq 1\end{aligned} \tag{17}$$

which uses the fact that for all $i \in [1, i_0)$, we have

$$|\beta_{i+1} - \beta^*| \leq |\beta_1 - \beta^*|, \text{ and } \frac{|\beta_i - \beta^*|^2}{|\beta_{i_0-1} - \beta^*|^2} \geq 1 \tag{18}$$

due to the monotonicity of $\{\beta_i\}$. The inequality in (17) implies quadratic convergence rate of $\{\beta_i\}$. ■

V. Two Distribution-oblivious Online Algorithms for the I.I.D. Delay Setting

In the sequel, we propose two online algorithms that do not need the detailed probability distribution $\mathbb{P}_{Y_{i-1}, Z_{i-1}}$. This section considers the simpler setting in which the delay (vector) process $\{(Y_i, Z_i) : i \geq 1\}$ is *i.i.d.,* and we derive two online algorithms that are provably convergent to the optimal solution. Some practical issues will then be discussed in Sec. VI. The general case where $\{(Y_i, Z_i) : i \geq 1\}$ can be any ergodic stationary *Markov* process is considered in Sec. VII, where we explain how the designed algorithms can be seamlessly modified to accommodate the Markovian delay even though we no longer have provable convergence.

Before proceeding, we introduce a few new notations necessary when describing the algorithm. For any $\beta \in [0, \beta_{\text{UB}})$ and any $0 \leq y', z' < \infty$, we define

$$g_1(y', z', \beta) \triangleq G_1(y', z', \phi_{\Gamma,\beta}(y', z')) \tag{19}$$

$$g_2(y', z', \beta) \triangleq y' + z' + \phi_{\Gamma,\beta}(y', z') \tag{20}$$

$$\overline{g}_1(\beta) \triangleq \mathbb{E}_{Y_{i-1}, Z_{i-1}} \{ g_1(Y_{i-1}, Z_{i-1}, \beta) \} \tag{21}$$

$$\overline{g}_2(\beta) \triangleq \mathbb{E}_{Y_{i-1}, Z_{i-1}} \{ g_2(Y_{i-1}, Z_{i-1}, \beta) \} \tag{22}$$

Recalling that $f_\Gamma(\beta)$ is the average AoI penalty when $X_i = \phi_{\Gamma,\beta}(Y_{i-1}, Z_{i-1})$, by (8) we have

$$f_\Gamma(\beta) = \frac{\overline{g}_1(\beta)}{\overline{g}_2(\beta)}, \quad \forall \beta \in [0, \beta_{\text{UB}}). \tag{23}$$

A. Algorithm 1: Fixed-point-iteration-based Solution

The detailed step is described in Algorithm 1. At the beginning of the i-th round (Line 4), the algorithm updates the value β_i, see Lines 6 and 8, and then use (11) to compute the waiting time $X_i = \phi_{\Gamma,\beta_i}(Y_{i-1}, Z_{i-1})$ (Line 10), and update two register values (Lines 12 and 13). Then wait for X_i time before sending out the i-th packet. After sending the packet, source waits for the ACK of the i-th packet (Line 14) and the iteration continues.

We now elaborate how to compute β_i in Algorithm 1 in Lines 6 and 8, which would then be used to find X_i in Line 10. We initialize $\beta_1 = \beta_2 = 0$ in Line 6. For $i \geq 3$, we use the $g_1(y', z', \beta)$ and $g_2(y', z', \beta)$ functions defined in (19) and (20), respectively, and compute

$$\beta_i = \frac{\sum_{j=1}^{i-1} g_1(Y_{j-1}, Z_{j-1}, \beta_j)}{\sum_{j=1}^{i-1} g_2(Y_{j-1}, Z_{j-1}, \beta_j)} = \frac{\sum_{j=1}^{i-1} g_1(Y_{j-1}, Z_{j-1}, \beta_j)}{S_{i-1}}. \tag{24}$$

The denominator of (24) is derived by noting that

$$\begin{aligned}&g_2(Y_{j-1}, Z_{j-1}, \beta_j) = Y_{j-1} + Z_{j-1} + \phi_{\Gamma,\beta_j}(Y_{j-1}, Z_{j-1}) \\ &= Y_{j-1} + Z_{j-1} + X_j = S_j - S_{j-1}, \ \forall j \in [1, i-1]\end{aligned} \tag{25}$$

[5] For instance, if (i) γ is doubly continuously differentiable and (ii) Y_i and Z_i are discrete random variables with $N_Y < \infty$ and $N_Z < \infty$ points having strictly positive probabilities, then $f_\Gamma(\beta)$ is doubly continuously differentiable for the entire domain $(0, \beta_{\text{UB}})$ expect for up to $N_Y \cdot N_Z$ points. Then as long as the optimal β^* does not fall into any of the $N_Y \cdot N_Z$ points, then this assumption holds. Another scenario for which such assumption holds is if both Y and Z are well-behaved continuous random variables, e.g., both being exponential or both being log-normal, etc. In this scenario, $f_\Gamma(\beta)$ is doubly continuously differentiable for the entire domain $(0, \beta_{\text{UB}})$ except for a single point $\beta_{\text{singular}} = \sup\{\beta \geq 0 : f_\Gamma(\beta) = \beta_{\text{ZW}}\}$.

and hence the denominator of (24) can be simplified by the telescoping argument $\sum_{j=1}^{i-1} g_2(Y_{j-1}, Z_{j-1}, \beta_j) = S_{i-1}$. That is why we use the register $\mathbf{snd.time}$ to record the latest "send time" in Line 13 and then use this value as the denominator in Line 8.

In fact, one can prove that the $\mathbf{snd.time}$ of round 1 is always $S_1 = 0$ and the $\mathbf{snd.time}$ of round 2 is $S_2 > 0$. Note that in the beginning of round i, we use the $\mathbf{snd.time}$ value of the *previous round* as the denominator of β_i in Line 8, also see (24). Therefore, only at round $i = 3$ can we start to have a strictly positive denominator in Line 8 (and in (24)). That is why we hardwire $\beta_1 = \beta_2 = 0$ in Line 6.

Algorithm 1 Fixed-point-iteration-based online algorithm

Universal input for every round: A set of statistics of Y, denoted as SY_γ

Per-round input: (Y_{i-1}, Z_{i-1})

Per-round output: Waiting time X_i in the i-th round

```
1: Initialize $Y_0 = Z_0 = A_0 = 0$ (see Sec. III-A)
2: Maintain two scalar registers for $\mathbf{snd.time}$ and $\mathbf{sum.AoI.pnlty}$
3: Initialize $\mathbf{snd.time} = \mathbf{sum.AoI.pnlty} = 0$
4: for time instant $A_{i-1}$, i.e., the beginning of round $i = 1, 2, 3, \cdots$ do
5:     if $i \leq 2$ then
6:         $\beta_i = 0$
7:     else
8:         $\beta_i = \frac{\mathbf{sum.AoI.pnlty}}{\mathbf{snd.time}}$, which implements (24)
9:     end if
10:    Use (11) and $\mathsf{SY}_\gamma$ to compute $X_i = \phi_{\Gamma,\beta_i}(Y_{i-1}, Z_{i-1})$
11:    Use (19) and $\mathsf{SY}_\gamma$ to compute $g_1(Y_{i-1}, Z_{i-1}, \beta_i)$
12:    Update $\mathbf{sum.AoI.pnlty} = \mathbf{sum.AoI.pnlty} + g_1(Y_{i-1}, Z_{i-1}, \beta_i)$
13:    Update $\mathbf{snd.time} = \mathbf{snd.time} + Y_{i-1} + Z_{i-1} + X_i$
14:    Wait for $X_i$ time, send the $i$-th packet, and wait for the ACK to start the next round
15: end for
```

The numerator of (24) can also be simplified. That is, there is no need to repeat the summation $\sum_{j=1}^{i-1} g_1(Y_{j-1}, Z_{j-1}, \beta_j)$ for each i. Instead, we only need to "update" the sum by adding the increment from the previous upper limit $i-2$ to the new upper limit $i-1$, as shown in Line 12 of Algorithm 1. By combining these two simplifications, the actual update of β_i is carried out in Line 8 of Algorithm 1.

B. Intuition of Algorithm 1

Note that each $g_1(Y_{j-1}, Z_{j-1}, \beta_j)$ term in the summation can be viewed as the *empirical AoI penalty* experienced during time interval (S_{j-1}, S_j). As a result, (24) computes the ratio of the past total AoI penalty over the past duration $[0, S_{i-1}]$, which is essentially the *empirical average AoI penalty.* We then use it as the new threshold β_i to decide the $X_i = \phi_{\Gamma,\beta_i}(Y_{i-1}, Z_{i-1})$ for the i-th round. This closely follows the spirit of the fixed-point iteration

$$\beta_i = f_\Gamma(\beta_{i-1}) = \frac{\overline{g}_1(\beta_{i-1})}{\overline{g}_2(\beta_{i-1})} \tag{26}$$

in Proposition 3. The differences between (24) and (26) are (i) (24) not only depends on β_{i-1} but also on $\{\beta_j : j \leq i-1\}$ and (ii) (24) uses the empirical $g_1(Y_{j-1}, Z_{j-1}, \beta_j)$ and $g_2(Y_{j-1}, Z_{j-1}, \beta_j)$ rather than the expectations $\overline{g}_1(\beta_{i-1})$ and $\overline{g}_2(\beta_{i-1})$. Therefore, $\{\beta_i\}$ in (24) is a *random process* but $\{\beta_i\}$ in Proposition 3 (also in (26)) is a deterministic sequence.

C. Knowledge Required to Run Algorithm 1

In order to run Algorithm 1, we need to compute $\phi_{\Gamma,\beta}(y', z', \beta)$ and $g_1(y', z', \beta)$ using (11) and (19), respectively. Recall that in this section (Sec. V) we only consider the *i.i.d.* vector random process $\{(Y_i, Z_i) : i \geq 1\}$. By combining (3), (4), (11), (12) and (19), we have a simplified version

$$\phi_{\Gamma,\beta}(y', z') = \inf\left\{t > 0 : \mathbb{E}\left\{\gamma(y' + z' + t + Y)\right\} > \beta\right\} \tag{27}$$

$$g_1(y', z', \beta) = \mathbb{E}_Y\left\{\int_Y^{y'+z'+\phi_{\Gamma,\beta}(y',z')+Y} \gamma(t)dt\right\}. \tag{28}$$

Both still require some knowledge of the statistics of Y. That is why in Algorithm 1, we use a notation SY_γ to denote the needed Statistics of Y (SY) and we clearly indicate that SY_γ is needed in Lines 10 and 11. As we will see, the set of needed statistics depends on the AoI penalty function $\gamma(\cdot)$ and this is why we have γ in the subscript of SY_γ.

We discuss the cases of the most popular penalty functions $\gamma_{\text{lin}}(\cdot)$, $\gamma_{\text{qdr}}(\cdot)$, and $\gamma_{\text{exp}}(\cdot)$. Note that the proposed algorithm is not limited to the above choices and can be tailored for other choices of $\gamma(\cdot)$ satisfying the technical assumption (iii) described in Sec. III-A. One just has to analyze the needed SY_γ separately for other classes of penalty functions.

Case 1: Linear penalty $\gamma_{lin}(\Delta) = \Delta$. We use $\phi^{\text{SY}}_{\text{lin},\beta}(y', z')$ to denote the waiting time function $\phi_{\Gamma,\beta}(y', z')$ specialized for $\gamma_{\text{lin}}(\Delta)$. Similarly, $g^{\text{SY}}_{\text{lin},1}(y', z', \beta)$ denotes the $g_1(y', z', \beta)$ specialized for $\gamma_{\text{lin}}(\Delta)$. The superscript "SY" indicates that this function requires (knowing) some Statistics of $\mathbb{P}_Y$. Applying simple calculus to (3), (4), (11), and (19) shows that

$$\phi^{\text{SY}}_{\text{lin},\beta}(y', z') = \max\left(\beta - \mathbb{E}\{Y\} - y' - z', 0\right) \tag{29}$$

$$g^{\text{SY}}_{\text{lin},1}(y', z', \beta) = \frac{\left(y' + z' + \phi^{\text{SY}}_{\text{lin},\beta}(y', z')\right)^2}{2} + \left(y' + z' + \phi^{\text{SY}}_{\text{lin},\beta}(y', z')\right)\mathbb{E}\{Y\}. \tag{30}$$

From (29) and (30), it is clear that to calculate $\phi^{\text{SY}}_{\text{lin},\beta}(y', z')$ and $g^{\text{SY}}_{\text{lin},1}(y', z', \beta)$, the only statistical knowledge we need is a scalar $\mathsf{SY}_\gamma = \mathbb{E}\{Y\}$.

Remark 4: In the case of the *i.i.d.* $\{(Y_i, Z_i) : i \geq 1\}$ and the linear penalty $\gamma_{\text{lin}}(\Delta) = \Delta$, the following modification of Algorithm 1 will lead to a provably-convergent-to-optimality online algorithm that completely waives the need of statistical knowledge $\mathsf{SY}_\gamma = \mathbb{E}\{Y\}$. To that end, we note that the objective function (6) can be simplified as

$$\beta^* = \inf_{X_i} \frac{\mathbb{E}\left\{\frac{1}{2}(Y_{i-1}+Z_{i-1}+X_i)^2\right\}}{\mathbb{E}\left\{Y_{i-1}+Z_{i-1}+X_i\right\}} + \mathbb{E}\{Y\} \tag{31}$$

in the case of *i.i.d.* $\{(Y_i, Z_i) : i \geq 1\}$ and the linear penalty $\gamma_{\text{lin}}(\Delta) = \Delta$, which is also observed in [4], [17]. Since the term $\mathbb{E}\{Y\}$ on the right-hand side of (31) is not dependent on the decision variable X_i, if we define an auxiliary variable $\zeta \triangleq \beta - \mathbb{E}\{Y\}$, then (31) becomes

$$\zeta^* = \beta^* - \mathbb{E}\{Y\} = \inf_{X_i} \frac{\mathbb{E}\left\{\frac{1}{2}(Y_{i-1}+Z_{i-1}+X_i)^2\right\}}{\mathbb{E}\left\{Y_{i-1}+Z_{i-1}+X_i\right\}}. \tag{32}$$

Using the auxiliary variable ζ, the modified waiting time function and empirical AoI penalty in Algorithm 1 then become

$$\tilde{\phi}_{\text{lin},\zeta}(y', z') = \max\left(\zeta - y' - z', 0\right) \tag{33}$$

$$\tilde{g}_{\text{lin},1}(y', z', \zeta) = \frac{\left(y' + z' + \tilde{\phi}_{\text{lin},\zeta}(y', z')\right)^2}{2} \tag{34}$$

where $\tilde{\phi}_{\text{lin},\zeta}(y', z')$ and $\tilde{g}_{\text{lin},1}(y', z', \zeta)$ no longer need the knowledge of $\mathbb{E}\{Y\}$. Comparing (33), (34) with (29) and (30), this is as if we are facing a system with $\mathbb{E}\{Y\} = 0$ being known. Therefore, we can apply the same Algorithm 1 on finding ζ^* with no modification other than (33) and (34). However, the above modification using the auxiliary variable only applies to the case of *i.i.d.* $\{(Y_i, Z_i) : i \geq 1\}$ and the linear penalty $\gamma_{\text{lin}}(\Delta) = \Delta$, where $\mathbb{E}\{Y\}$ is *separated* from other terms (see (31)).

Case 2: Exponential penalty $\gamma_{exp}(\Delta) = e^{a\Delta} - 1$ for a constant a that is known globally. Similar to the previous case, we use $\phi^{\mathsf{SY}}_{\text{exp},\beta}(y', z')$ and $g^{\mathsf{SY}}_{\text{exp},1}(y', z', \beta)$ to describe the $\phi_{\Gamma,\beta}(y', z')$ and $g_1(y', z', \beta)$ specialized for $\gamma_{\text{exp}}(\Delta)$. Again, applying simple calculus to (3), (4), (11), and (19) shows that

$$\begin{aligned}
&\phi^{\mathsf{SY}}_{\text{exp},\beta}(y', z') \\
&\quad= \max\left(\frac{\ln(\beta+1) - \ln\left(\mathbb{E}\{e^{aY}\}\right)}{a} - y' - z', 0\right)
\end{aligned} \tag{35}$$

$$\begin{aligned}
&g^{\mathsf{SY}}_{\text{exp},1}(y', z', \beta) \\
&\quad= \frac{\left(e^{a\left(y'+z'+\phi^{\mathsf{SY}}_{\text{exp},\beta}(y',z')\right)} - 1\right)\mathbb{E}\{e^{aY}\}}{a} \\
&\quad\quad - \left(y' + z' + \phi^{\mathsf{SY}}_{\text{exp},\beta}(y', z')\right).
\end{aligned} \tag{36}$$

From (35) and (36), it is clear that to calculate $\phi^{\mathsf{SY}}_{\text{exp},\beta}(y', z')$ and $g^{\mathsf{SY}}_{\text{exp},1}(y', z', \beta)$, the only statistical knowledge we need is a scalar $\mathsf{SY}_\gamma = \mathbb{E}\{e^{aY}\}$ where a is the exponent of the $\gamma_{\text{exp}}(\cdot)$ that is known globally in advance.

Case 3: Quadratic $\gamma_{qdr}(\Delta) = \Delta^2$. We use $\phi^{\mathsf{SY}}_{\text{qdr},\beta}(y', z')$ and $g^{\mathsf{SY}}_{\text{qdr},1}(y', z', \beta)$ to describe the $\phi_{\Gamma,\beta}(y', z')$ and $g_1(y', z', \beta)$ specialized for $\gamma_{\text{qdr}}(\Delta)$. We then have

$$\begin{aligned}
w_{\text{qdr}}(\beta) \triangleq\ &\mathbb{1}_{\{\beta + (\mathbb{E}\{Y\})^2 \geq \mathbb{E}\{Y^2\}\}} \cdot \\
&\left(\sqrt{\beta + (\mathbb{E}\{Y\})^2 - \mathbb{E}\{Y^2\}} - \mathbb{E}\{Y\}\right)
\end{aligned} \tag{37}$$

$$\phi^{\mathsf{SY}}_{\text{qdr},\beta}(y', z') = \max\left(w_{\text{qdr}}(\beta) - y' - z', 0\right) \tag{38}$$

$$\begin{aligned}
g^{\mathsf{SY}}_{\text{qdr},1}(y', z', \beta) = &\frac{\left(y' + z' + \phi^{\mathsf{SY}}_{\text{qdr},\beta}(y', z')\right)^3}{3} \\
&+ \left(y' + z' + \phi^{\mathsf{SY}}_{\text{qdr},\beta}(y', z')\right)^2 \mathbb{E}\{Y\} \\
&+ \left(y' + z' + \phi^{\mathsf{SY}}_{\text{qdr},\beta}(y', z')\right) \mathbb{E}\{Y^2\}.
\end{aligned} \tag{39}$$

From (37) to (39), it is clear that to calculate $\phi^{\mathsf{SY}}_{\text{qdr},\beta}(y', z')$ and $g^{\mathsf{SY}}_{\text{qdr},1}(y', z', \beta)$, the only statistical knowledge we need is a pair $\mathsf{SY}_\gamma = \left(\mathbb{E}\{Y\}, \mathbb{E}\{Y^2\}\right)$.

Depending on which $\gamma(\cdot)$ is considered, the corresponding $\phi_{\Gamma,\beta}(\cdot,\cdot)$ and $g_1(\cdot,\cdot,\cdot)$ functions in Lines 10 and 11 are different. For ease of exposition, we introduce the following notation.

(i) $\Xi^{\mathsf{SY}}_{\text{lin}}$ denotes Algorithm 1 when specialized for $\gamma_{\text{lin}}(\Delta) = \Delta$. In this case $\mathsf{SY}_\gamma = \mathbb{E}\{Y\}$.

(ii) $\Xi^{\mathsf{SY}}_{\text{qdr}}$ denotes Algorithm 1 when specialized for $\gamma_{\text{qdr}}(\Delta) = \Delta^2$. In this case $\mathsf{SY}_\gamma = \left(\mathbb{E}\{Y\}, \mathbb{E}\{Y^2\}\right)$.

(iii) $\Xi^{\mathsf{SY}}_{\text{exp}}$ denotes Algorithm 1 when specialized for $\gamma_{\text{exp}}(\Delta) = e^{a\Delta} - 1$. In this case $\mathsf{SY}_\gamma = \mathbb{E}\{e^{aY}\}$.

One remarkable feature of Algorithm 1 is that instead of requiring the knowledge of the entire delay distribution (e.g., pdf or cdf or pmf), it requires only a scalar statistic (Cases 1 and 2) or a pair of statistics (Case 3).

D. Feasibility and Convergence of Algorithm 1

One implicit but key assumption in Algorithm 1 is that when describing the $\phi_{\Gamma,\beta}$ scheme, we require the input parameter β to be in the range $[0, \beta_{\text{UB}})$. Therefore, the feasibility of Algorithm 1 hinges on that all β_i computed in Line 8 are in the range $[0, \beta_{\text{UB}})$. We affirm this feasibility condition as follows.

Lemma 6: For any $\gamma(\cdot)$ function, the random process β_i computed by the iterative formula (24) satisfies

$$\sup\{\beta_i : i \in [1, \infty)\} \leq \beta_{\max} < \beta_{\text{UB}} \tag{40}$$

almost surely for some constant $\beta_{\max}$.

Proof: See Appendix G of the supplementary material. ■

For example, for the linear, quadratic, and exponential $\gamma(\cdot)$, we have $\beta_{\text{UB}} = \infty$. Since β_i is the empirical average AoI penalty (which remains finite all the time), the condition $\beta_i < \beta_{\text{UB}}$ is trivially true. However, for the signal-agnostic sampling of the Ornstein-Uhlenbeck (OU) process [21], the equivalent AoI penalty is $\gamma(\Delta) = \frac{\sigma^2}{2\theta}(1 - e^{-2\theta\Delta})$, and the corresponding $\beta_{\text{UB}} = \frac{\sigma^2}{2\theta}$. As a result, the inequality $\beta_i < \beta_{\text{UB}}$ becomes a non-trivial condition that needs to carefully examined. Lemma 6 guarantees $\beta_i < \beta_{\text{UB}}$ always holds regardless whether $\beta_{\text{UB}} \triangleq \lim_{\Delta\to\infty} \gamma(\Delta)$ is infinite or finite.

We now present the optimality results.

Proposition 4 (Convergence in probability): There exist $\alpha \in (0, 0.5)$ and $c_1, c_2, c_3, c_4 > 0$ such that $\forall i \geq 1$,

$$\mathbb{P}\left(\beta_{i+1} < \beta^* - c_1 \cdot i^{-(0.5-\alpha)}\right) \leq c_2 \cdot \exp\left(-c_3 \cdot i^{2\alpha}\right) \quad (41)$$

$$\mathbb{E}\{\beta_i - \beta^*\} \leq c_4 \cdot i^{-(0.5-\alpha)} \quad (42)$$

Proof: We provide proof sketches in Appendix A. See Appendices H and I of the supplementary material for detailed proof. ■

Corollary 2 (Convergence in the 2nd mean (the L^2 norm)): The random process $\{\beta_i\}$ computed in (24) converges to β^* in the 2nd mean (or equivalently the L^2 norm).

Proof: See Appendix J of the supplementary material. ■

E. Algorithm 2: A Root-finding-based Online Algorithm

A close look at Algorithm 1 shows that it consists of two components. Firstly, create a (random) sequence β_i that eventually converges to the optimal β^*. Secondly, use each β_i to compute the waiting time $X_i = \phi_{\Gamma,\beta_i}(Y_{i-1}, Z_{i-1})$ for the i-th round. This decoupled structure immediately prompts the following question: Can we design a different online algorithm of computing β_i that also converges to β^*? If so, then the scheme will eventually have the optimal β^* and the waiting time $X_i = \phi_{\Gamma,\beta^*}(Y_{i-1}, Z_{i-1})$ will also become optimal.

This observation prompts the second online algorithm that uses the Robbins-Monro algorithm to compute/update β_i. All the subsequent discussion for this new algorithm assumes $\beta_{\text{UB}} = \lim_{t\to\infty} \gamma(t) = \infty$. The reason why we impose this non-trivial assumption will be provided in Sec. V-G.

By Lemma 5, β^* is the unique root of (13). Since $\overline{g}_1(\beta)$, and $\overline{g}_2(\beta)$ in (21) and (22) are both finite for any $\beta \in [0, \beta_{\text{UB}}) = [0, \infty)$, β^* is also the unique root of the equation

$$\beta \cdot \overline{g}_2(\beta) - \overline{g}_1(\beta) = 0. \quad (43)$$

Since $\overline{g}_1(\beta)$ and $\overline{g}_2(\beta)$ take the *expectations* of the functions $g_1(\cdot,\cdot,\cdot)$ and $g_2(\cdot,\cdot,\cdot)$, respectively, the task of finding β^* can be solved by the Robbins-Monro algorithms [22], [23] that find the root of (43), which results in our new Algorithm 2.

Algorithms 1 and 2 are very similar. Specifically, both use the first two rounds $i \leq 2$ for initialization. The computed β_i is then used to compute the waiting time X_i (see Lines 8 and 10 of Algorithm 1 and Lines 8 and 10 of Algorithm 2) in the same way. The main difference between Algorithms 1 and 2 is how β_i is computed. For any step-size parameter $\eta > 0$, for all $i \geq 3$, Line 8 of Algorithm 2 essentially computes

$$\beta_i = \beta_{i-1} - \frac{\eta}{i} \cdot \left(\beta_{i-1} \cdot g_2(Y_{i-2}, Z_{i-2}, \beta_{i-1}) - g_1(Y_{i-2}, Z_{i-2}, \beta_{i-1})\right) \quad (44)$$

where $g_1(\cdot,\cdot,\cdot)$ and $g_2(\cdot,\cdot,\cdot)$ are defined in (19) and (20), respectively.

Algorithm 2 Root-finding-based (Robbins-Monro) online algorithm

Universal input for every round: η and SY_γ (a set of statistics of Y)
Per-round input: (Y_{i-1}, Z_{i-1})
Per-round output: Waiting time X_i in the i-th round

```
1: Initialize $Y_0 = Z_0 = A_0 = 0$ (see Sec. III-A)
2: Maintain two scalar registers $\mathbf{curr.g1}$ and $\mathbf{curr.g2}$
3: Initialize $\mathbf{curr.g1} = \mathbf{curr.g2} = 0$
4: for time instant $A_{i-1}$, i.e., the beginning of round $i = 1, 2, 3, \cdots$ do
5:     if $i \leq 2$ then
6:         $\beta_i = 0$
7:     else
8:         $\beta_i = \beta_{i-1} - \frac{\eta}{i} \cdot (\beta_{i-1} \cdot \mathbf{curr.g2} - \mathbf{curr.g1})$ based on (44)
9:     end if
10:    Use (11) and $\mathsf{SY}_\gamma$ to compute $X_i = \phi_{\Gamma,\beta_i}(Y_{i-1}, Z_{i-1})$
11:    Use (19), $\beta_i$, $\mathsf{SY}_\gamma$ to compute $\mathbf{curr.g1} = g_1(Y_{i-1}, Z_{i-1}, \beta_i)$
12:    $\mathbf{curr.g2} = Y_{i-1} + Z_{i-1} + X_i$ (see (20))
13:    Wait for $X_i$ time, then send the $i$-th packet, and wait for the ACK to start the next round
14: end for
```

The update rule in (44) follows from standard Robbins-Monro algorithm proposed in [22] since conditioning on the previous β_{i-1} value, we have

$$\begin{aligned}&\mathbb{E}\{\beta_{i-1} \cdot g_2(Y_{i-2}, Z_{i-2}, \beta_{i-1}) - g_1(Y_{i-2}, Z_{i-2}, \beta_{i-1}) | \beta_{i-1}\} \\ &= \beta_{i-1} \cdot \mathbb{E}\{g_2(Y_{i-2}, Z_{i-2}, \beta_{i-1}) | \beta_{i-1}\} \\ &\quad - \mathbb{E}\{g_1(Y_{i-2}, Z_{i-2}, \beta_{i-1}) | \beta_{i-1}\} \qquad (45) \\ &= \beta_{i-1} \cdot \overline{g}_2(\beta_{i-1}) - \overline{g}_1(\beta_{i-1}) \qquad (46)\end{aligned}$$

where (46) follows from the facts that (i) $\{(Y_{i-2}, Z_{i-2})\}$ is independent of $\{(Y_j, Z_j)\}_{i=0}^{j-3}$; (ii) β_{i-1} in (44) was computed by the history of $\{(Y_j, Z_j)\}_{i=0}^{j-3}$ and is thus independent of Y_{i-2} and Z_{i-2}; and (iii) the definitions in (21) and (22).

Blum [23] proved that the standard Robbins-Monro algorithm (i.e., $\{\beta_i\}$ computed by (44)) converges to the unique root (i.e., β^*) almost surely, provided that the following three conditions are met.

(i) $\{\beta_i\}$ computed by (44) is uniformly bounded.
(ii) $\beta \cdot \overline{g}_2(\beta) - \overline{g}_1(\beta)$ is non-decreasing.
(iii) $0 < \frac{d}{d\beta}\left(\beta \cdot \overline{g}_2(\beta) - \overline{g}_1(\beta)\right)\Big|_{\beta=\beta^*} < \infty$.

By proving that all three conditions hold in our AoI penalty minimization setting, we have

Proposition 5 (Almost sure convergence): For any $\eta > 0$, the sequence $\{\beta_i\}$ computed in (44) converges to β^* almost surely.

Proof: See Appendix M of the supplementary material. ■

F. Knowledge Required to Run Algorithm 2

Algorithm 2 requires the computation of $g_1(y', z', \beta)$, $g_2(y', z', \beta)$ and $\phi_{\Gamma,\beta}(y', z')$ for any $(Y_{i-1} = y', Z_{i-1} = z')$.

Since $g_2(y', z', \beta) = y' + z' + \phi_{\Gamma,\beta}(y', z')$ (see (20)), it is clear that Algorithm 2 only needs to compute $g_1(y', z', \beta)$ and $\phi_{\Gamma,\beta}(y', z')$, which is also needed by Algorithm 1. From the discussion in V-C, we conclude that Algorithm 2 requires the same statistics of Y (i.e., SY_γ) as Algorithm 1.[6]

For ease of future reference, we introduce another three notations for the corresponding online algorithms.

(i) $\Lambda^{\mathsf{SY}}_{\text{lin}}$ denotes Algorithm 2 when specialized for the linear AoI penalty function $\gamma_{\text{lin}}(\Delta) = \Delta$;

(ii) $\Lambda^{\mathsf{SY}}_{\text{qdr}}$ denotes Algorithm 2 when specialized for the quadratic AoI penalty function $\gamma_{\text{qdr}}(\Delta) = \Delta^2$;

(iii) $\Lambda^{\mathsf{SY}}_{\text{exp}}$ denotes Algorithm 2 when specialized for the exponential AoI penalty function $\gamma_{\text{exp}}(\Delta) = e^{a\Delta} - 1$

where we use the Λ schemes for the Robbins-Monro-based solution in this subsection (Ξ is reserved for the fixed-point-based scheme in Sec. V-A). The superscript SY indicates that these three schemes require the Statistics of Y.

G. Two Critical Differences between Algorithms 1 and 2

Difference #1: Algorithm 2 can be applied only if the AoI penalty function $\gamma(\cdot)$ satisfies $\beta_{\text{UB}} = \lim_{t\to\infty} \gamma(t) = \infty$ while Algorithm 1 does not require this restrictive assumption. The reason is as follows. Suppose $\beta_{\text{UB}} < \infty$. We note that the update rule (44) starts from the previous β_{i-1} value and then adds a random disturbance term. In the initial rounds (when i is still small), the step size $\frac{\eta}{i}$ is still large. Therefore, the new β_i after random perturbation may be outside the target range $[0, \beta_{\text{UB}})$. The impact of possibly having $\beta_i \geq \beta_{\text{UB}}$ is catastrophic since it immediately results in an infinite waiting time $X_i = \phi_{\Gamma,\beta_i}(\cdot,\cdot)$ (see the discussion on β_{UB} in Sec. IV-A) that halts the entire system. This is the reason why Algorithm 2, at least in its current form, is feasible only under the assumption $\beta_{\text{UB}} = \infty$, which guarantees $\beta_i < \beta_{\text{UB}} = \infty$ will always be within the right range.

One way to avoid having $\beta_i \geq \beta_{\text{UB}}$ is to choose a small η to begin with, see (44). However, choosing a small η will adversely affect the convergence speed even though eventually it still converges to optimum. This leads to the second main difference.

Difference #2: Finding the right step size $\frac{\eta}{i}$ that balances the convergence speed and stability is important for the Robbins-Monroe algorithms. For comparison, there is no step-size parameter in the fixed-point-based solution in Algorithm 1. In a broad sense, the fixed-point update rule in Algorithm 1 which uses the old empirical average as the new β_i is able to "self-regulate" the perturbation of each update step without an explicitly specified step size needed in a Robbins-Monro algorithm.

Remark 5: For the scenario of $\beta_{\text{UB}} < \infty$, it is possible to add a projection operation to the Robbins-Monro algorithm when updating β_i in (44). One natural choice is to project β_i back to $[0, \beta_{\text{UB}}]$ so that all $\beta_i \leq \beta_{\text{UB}}$. However, such immediate modification still does not work since we need $\beta_i < \beta_{\text{UB}}$ with strict inequality (note that $\beta_i = \beta_{\text{UB}}$ still leads to $X_i = \infty$). As will be seen in the numerical results, Algorithm 1 offers superior/equal performance to Algorithm 2 while being more robust on all scenarios without the need of fine tuning step sizes. We thus leave the generalization of Algorithm 2 for the scenario of $\beta_{\text{UB}} < \infty$ as future work.

VI. Addressing Practical Issues in the I.I.D. Delay Setting

A. Using Running Average to Replace SY_γ

Note that both Algorithm 1 (Sec. V-A) and Algorithm 2 (Sec. V-E) require the knowledge of SY_γ, some statistics of Y, to run. Since estimating a statistic is much easier than learning the entire distribution $\mathbb{P}_Y$, we can further substitute the values of SY_γ in Algorithms 1 and 2 by the running empirical averages to obtain truly distribution-oblivious online algorithms.

For example, consider the linear $\gamma_{\text{lin}}(\Delta) = \Delta$, in which case $\mathsf{SY}_\gamma = \mathbb{E}\{Y\}$. We first set the window size $N_{\text{RA}} > 0$. Then, for $i = 1$ (at the beginning of the first round), we use the value 0 as the estimate of $\mathbb{E}\{Y\}$. For $i \geq 2$, we compute the running empirical average that has a fixed window size N_{RA}:

$$\frac{1}{\min(i-1, N_{\text{RA}})} \sum_{j=\min(1, i-N_{\text{RA}})}^{i-1} Y_j. \tag{47}$$

The complexity of computing the running empirical average is $O(1)$ per slot, since for every round one can simply add the latest term Y_{i-1} and subtract the oldest term $Y_{i-N_{\text{RA}}}$.

Once we replace the statistics SY_γ by its running average, we can carry out the computation of Lines 10 and 11 of Algorithm 1 (resp. Algorithm 2) without knowing the true value of SY_γ. Similar RA-based substitution can be applied to other non-linear $\gamma(\cdot)$ as well, including $\gamma_{\text{qdr}}(\cdot)$ and $\gamma_{\text{exp}}(\cdot)$. We denote the resulting online algorithms by $\Xi^{\text{RA}}_{\text{lin}}$, $\Xi^{\text{RA}}_{\text{qdr}}$ and $\Xi^{\text{RA}}_{\text{exp}}$ (resp. $\Lambda^{\text{RA}}_{\text{lin}}$, $\Lambda^{\text{RA}}_{\text{qdr}}$ and $\Lambda^{\text{RA}}_{\text{exp}}$) for the fixed-point-based (resp. Robbins-Monro-based) algorithm.

B. The Case of Unknown AoI Penalty Function $\gamma(\cdot)$

In all the previous sections, we implicitly assume that the source has perfect knowledge of the AoI penalty function $\gamma(\cdot)$, which may not always hold in real world. In particular, the AoI penalty depends on the specific application that is currently running *at the destination*. Although the source is able to compute the value of the AoI using the time stamps, it may not have access to the application layer ($\gamma(\Delta)$ value) *at the destination*. Even the destination may not have full knowledge of its own AoI penalty function. See the discussion in Sec. I. This section addresses this important practical need of estimating $\gamma(\cdot)$ on the fly.

For any i, when the destination receives the update packet at time $S_i + Y_i$, if we denote $(Y_{i-1} = y', Z_{i-1} = z', X_i = x, Y_i = y)$ and define $p_i \triangleq y' + z' + x + y$, then the peak AoI penalty at that time is $\gamma(p_i)$ (see Fig. 2). Suppose the destination does not have full knowledge of $\gamma(\cdot)$. Instead,

[6] As is discussed in Remark 4, in the case of *i.i.d.* $\{(Y_i, Z_i) : i \geq 1\}$ and the linear penalty $\gamma_{\text{lin}}(\Delta) = \Delta$, $\mathbb{E}\{Y\}$ is separated from other terms in the objective function (see (31)). Hence, one can design a Robbins-Monro online algorithm based on Algorithm 2 that does not require SY_γ by using $\tilde{\phi}_{\text{lin},\zeta}(y', z')$ and $\tilde{g}_{\text{lin},1}(y', z', \zeta)$ (see (33) and (34)). Also see [17] for a similar Robbins-Monro-based online algorithm that waives the need of SY_γ based on the observation $\mathbb{E}\{Y\}$ being separated from other terms.

destination can observe how good/poor the system state is at that time instant and use it to estimate the scalar AoI penalty value $\gamma(p_i)$. For example, say at time $S_i + Y_i$, the system is on the brink of major disruption due to the inability to receive update packets for a long time (large p_i value), then the AoI penalty $\gamma(p_i)$ value is likely to be large. On the contrary, if at the time of receiving the latest packet, the system is still functioning normally, then the $\gamma(p_i)$ value is likely to be small. Since destination estimates $\gamma(p_i)$ by observing the system state, we assume that destination knows the value of $q_i = \gamma(p_i) + n_i$, where n_i's are i.i.d. zero-mean Gaussian random variables that represent the estimation/observation error. The values of the pair (p_i, q_i) is then fed back to the source via ACK.

At the i-th round, the source maintains the set of the past $N_{\hat{\gamma}}$ observations $\mathcal{S}_{\hat{\gamma}} = \{(p_j, q_j)\}_{j=i-N_{\hat{\gamma}}}^{i-1}$. Any (p_i, q_i) outside of this set is considered *too old* and is excluded from consideration. Our goal is to estimate the true $\gamma(\cdot)$ using the noisy observations $\mathcal{S}_{\hat{\gamma}}$.

Step 1: Since the penalty function $\gamma(\cdot)$ is non-negative and satisfies $\gamma(0) = 0$, we first add the point $(0, 0)$ to the set $\mathcal{S}_{\hat{\gamma}}$.

Step 2: We sort and relabel the elements of $\mathcal{S}_{\hat{\gamma}}$ in an ascending order of p_i and thus we have $0 = p_1 \leq p_2 \leq \cdots \leq p_{N_{\hat{\gamma}}+1}$. Note that since $\gamma(\cdot)$ is an increasing function, the sequence $\{\gamma(p_i)\}$ after sorting is also increasing. However, with noisy observation $q_i = \gamma(p_i) + n_i$ the corresponding $\{q_i\}_{i=1}^{N_{\hat{\gamma}}+1}$ may not be an increasing sequence.

Step 3: Since the AoI analysis and scheduler designs rely heavily on the assumption that $\gamma(\cdot)$ is monotonically increasing, our plan is to first solve the following quadratic programming problem

$$\min_{\{\hat{q}_i\}_{i=1}^{N_{\hat{\gamma}}+1}} \sum_{i=1}^{N_{\hat{\gamma}}+1} (\hat{q}_i - q_i)^2$$
$$\text{subject to } \hat{q}_i \leq \hat{q}_{i+1}, \ 1 \leq i \leq N_{\hat{\gamma}} \tag{48}$$

that gives us a new sequence of pairs $(p_i, \hat{q}_i)$ that is non-decreasing in both coordinates. We then set the estimated AoI penalty function $\hat{\gamma}(\cdot)$ to be a piece-wise linear function with $N_{\hat{\gamma}}$ pieces. For all $i \in [1, N_{\hat{\gamma}} - 1]$, the i-th line segment is connecting the two pairs $(p_i, \hat{q}_i)$ and $(p_{i+1}, \hat{q}_{i+1})$. The last segment $(i = N_{\hat{\gamma}})$ is starting from $(p_{N_{\hat{\gamma}}}, \hat{q}_{N_{\hat{\gamma}}})$, going through $(p_{N_{\hat{\gamma}}+1}, \hat{q}_{N_{\hat{\gamma}}+1})$, and extending all the way to infinity. The mathematical expression of this piecewise-linear $\hat{\gamma}(\cdot)$ can be easily derived, see (49) in the supplementary material, and we thus omit the expression of $\hat{\gamma}(\cdot)$ herein.

In the literature, performing Step 3 to obtain the estimate $\hat{\gamma}(\cdot)$ is termed *monotonic regression* or *isotonic regression.* [7]

Step 4: After obtaining $\hat{\gamma}(\cdot)$, we can carry out the online Algorithms 1 and 2 easily by substituting $\gamma(\cdot)$ by $\hat{\gamma}(\cdot)$. Specifically, since the $\hat{\gamma}(\cdot)$ is piece-wise linear, two major substitutions are needed that are different from our previous discussion of γ_{lin}, γ_{exp}, and γ_{qdr} in Sec. V-C. First, we replace the waiting time $X_i = \phi_{\Gamma,\beta_i}(y', z')$ in Algorithms 1 and 2 with

$$\begin{aligned} X_i &= \phi^{\text{RA}}_{\hat{\gamma},\beta_i}(y', z') \\ &\triangleq \inf\left\{t > 0 : \frac{\sum_{j=i-N_{\text{sum}}}^{i-1} \hat{\gamma}\left(y' + z' + t + Y_j\right)}{N_{\text{sum}}} > \beta_i\right\} \end{aligned} \tag{49}$$

where $N_{\text{sum}} \leq N_{\hat{\gamma}}$ samples from the past Y_i are plugged into the estimated $\hat{\gamma}(\cdot)$ to approximate the expectation $\mathbb{E}(\gamma(y' + z' + t + Y))$ in (11) and (12).

Step 5: The second substitution is for computing $g_1(y', z', \beta_i)$. We first note that by (4) and (19), the value $g_1(y', z', \beta_i)$ in Algorithms 1 and 2 is *the expectation of an integral with random upper and lower limits.* As a result, we again use the combination of running average (RA) and estimated $\hat{\gamma}(\cdot)$ to estimate the $g_1(y', z', \beta_i)$, and we have

$$\begin{aligned} &g^{\text{RA}}_{\hat{\gamma},1}(y', z', \beta_i) \\ &\triangleq \frac{\sum_{j=i-N_{\text{sum}}}^{i-1} \left(\mathsf{Cumulative.trapezoidal.integral}_j\right)}{N_{\text{sum}}} \end{aligned} \tag{50}$$

where the summation and division are to compute the running empirical average, and we use the following cumulative trapezoidal integral

$$\begin{aligned} &\mathsf{Cumulative.trapezoidal.integral}_j \\ &\triangleq \sum_{n=1}^{N_{\text{trapezoid}}} \frac{\hat{\gamma}(x_j^{n-1}) + \hat{\gamma}(x_j^n)}{2} \delta_j \end{aligned} \tag{51}$$

in which $\{x_j^n\}_{n=0}^{N_{\text{trapezoid}}}$ is a uniformly spaced partition points of $[Y_j, Y_j + y' + z' + \phi^{\text{RA}}_{\hat{\gamma},\beta_i}(y', z')]$ (based on the j-th sample Y_j and the previously computed $\phi^{\text{RA}}_{\hat{\gamma},\beta_i}(y', z')$ value in (49)) and δ_j is the corresponding spacing between x_j^{n-1} and x_j^n. Mathematically,

$$\delta_j \triangleq \frac{y' + z' + \phi^{\text{RA}}_{\hat{\gamma},\beta_i}(y', z')}{N_{\text{trapezoid}}} \tag{52}$$

$$x_j^n \triangleq Y_j + n\delta_j, \ \forall n \in [0, N_{\text{trapezoid}}]. \tag{53}$$

After the modification described in Steps 1 to 5, both Algorithms 1 and 2 can be carried out for arbitrary unknown $\gamma(\cdot)$, not limited to the previous $\gamma_{\text{lin}}(\cdot)$, $\gamma_{\text{qdr}}(\cdot)$ or $\gamma_{\text{exp}}(\cdot)$.

We now analyze the complexity incurred of estimating $\hat{\gamma}(\cdot)$. In Step 1, we add a point $(0, 0)$ to the set $\mathcal{S}_{\hat{\gamma}}$ and hence the complexity of $O(1)$. For Step 2, we only need to *update* the sorted list so that per-time-slot complexity is only $\log N_{\hat{\gamma}}$. To carry out the monotonic regression in Step 3, one may use the active set algorithms in [24] that have a complexity of $O(N_{\hat{\gamma}})$.

For Step 4, given (y', z', β_i), evaluating the summation in (49) takes $O(N_{\text{sum}})$ time, and to compute the infimum t value in (49) we use the bisection method over the interval $[0, t_{\text{large}}]$ for a sufficiently large $t_{\text{large}} > 0$. Depending on the desired precision $\varrho > 0$, the combined complexity of (49) is $O(\log(t_{\text{large}}/\varrho)N_{\text{sum}})$. For Step 5, since the trapezoidal integral is done for each Y_j and we have N_{sum} such Y_j, together the complexity is $O(N_{\text{sum}} \cdot N_{\text{trapezoid}})$. In sum, the per-round complexity of this new modification is $O\left(N_{\hat{\gamma}} + \log N_{\hat{\gamma}} + N_{\text{sum}} \cdot (\log(t_{\text{large}}/\varrho) + N_{\text{trapezoid}})\right)$.

[7] Our derivation assumes $\gamma(\cdot)$ to be strictly increasing and Step 3 only guarantees a non-decreasing $\hat{\gamma}(\cdot)$. Nonetheless, if we have enough observations (i.e., large $N_{\hat{\gamma}}$), $\hat{\gamma}(\cdot)$ is very well-behaved (very close to being strictly increasing). As will be shown in the simulation results (Sec. VIII), the online algorithms using $\hat{\gamma}(\cdot)$ still achieve satisfactory performance.

We denote $\Xi^{\text{RA}}_{\hat{\gamma}}$ (resp. $\Lambda^{\text{RA}}_{\hat{\gamma}}$) to be the online algorithm that uses the above 5 steps to modify the original Algorithm 1 (resp. Algorithm 2). The resulting algorithm is fully distribution- and AoI-penalty-oblivious.

VII. Addressing Practical Issues in the Markov Delay Setting

In this section, we turn our attention to the case where $\{(Y_i, Z_i)\}$ is a stationary and ergodic Markov process (not necessarily i.i.d.), and we propose leveraging the k-nearest neighbors (KNN) algorithm to design an empirically-AoI-optimal scheduling scheme in this new setting.

By examining the proposed $\Xi^{\text{RA}}_{\hat{\gamma}}$ (resp. $\Lambda^{\text{RA}}_{\hat{\gamma}}$) in Sec. VI, we notice that the proposed schemes have a desirable "modular" structure. Namely, the estimation of the unknown $\gamma(\cdot)$ is separated from the actual evaluation of the functions $\phi_{\Gamma,\beta}$ and g_1 in (49) and (50), respectively. With a Markov delay setting, we can thus reuse the first 3 steps to estimate $\gamma(\cdot)$ and only modify Steps 4 and 5 accordingly.

In the i.i.d.-based computation (49) and (50), we use all the past N_{sum} observations $\{Y_j\}_{j=i-N_{\text{sum}}}^{i-1}$ to compute the running empirical average. For the Markovian delay setting, we propose using the k-nearest neighbors (KNN) algorithm that computes $\mathbb{P}_{Y_i|Y_{i-1},Z_{i-1}}$ by considering a set of k points that are the nearest to the given $(Y_{i-1}, Z_{i-1}) = (y', z')$ [25].

Specifically, at the i-th round, among all the past N_{sum} observed delay values $\{(Y_{j-1}, Z_{j-1}, Y_j)\}_{j=i-N_{\text{sum}}}^{i-1}$, we select N_{KNN} points such that the first two coordinates (Y_{j-1}, Z_{i-1}) are of the shortest Euclidean distance to the latest observation $(Y_{i-1}, Z_{i-1}) = (y', z')$. Then the *Modified Step 4* computes X_i in Algorithms 1 and 2 using

$$X_i = \phi^{\text{KNN}}_{\hat{\gamma},\beta_i}(y', z')$$
$$\triangleq \inf\left\{ t > 0 : \frac{\sum_{j=1}^{N_{\text{KNN}}} \hat{\gamma}\left(y' + z' + t + Y_j^{\text{select}}\right)}{N_{\text{KNN}}} > \beta_i \right\}. \tag{54}$$

Eq. (54) is almost identical to (49) except that Y_j^{select} are the third coordinates of the $N_{\text{KNN}} \leq N_{\text{sum}}$ samples nearest to (y', z') instead of all N_{sum} past samples.

Modified Step 5: Replace $g_1(y', z', \beta_i)$ in Algorithms 1 and 2 with

$$g^{\text{KNN}}_{\hat{\gamma},1}(y', z', \beta_i) \triangleq \frac{\sum_{j=1}^{N_{\text{KNN}}} \left(\mathsf{Cumulative.trapezoidal.integral}_j\right)}{N_{\text{KNN}}} \tag{55}$$

where the cumulative trapezoidal integral is computed in the same way as in (51), but replaces the Y_j used in (51) with the Y_j^{select} from the N_{KNN} samples nearest to (y', z'). Since the modified Steps 4 and 5 are very similar to the Steps 4 and 5 in Sec. VI-B, the KNN version has similar complexity as the RA version.

We use $\Xi^{\text{KNN}}_{\hat{\gamma}}$ (resp. $\Lambda^{\text{KNN}}_{\hat{\gamma}}$) to denote the online algorithm that uses the modified Algorithm 1 (resp. modified Algorithm 2). Table I summarizes the notations used for the fixed-point-based online algorithms. All the algorithms can be divided to two major categories, depending on whether it is developed exclusively for the i.i.d. delay models or for the more general Markov models; and whether it assumes the full knowledge of $\gamma(\cdot)$ or involves the penalty estimation $\hat{\gamma}(\cdot)$ component. Finally, we also distinguish the cases based on whether some statistics of $\mathbb{P}_Y$ are known *a priori* versus the truly distribution-oblivious setting for which the algorithm has absolutely zero knowledge of the distribution or statistics.

As can be seen, four major sets of algorithms are developed. We start from the most basic version that requires both the statistics and $\gamma(\cdot)$. Next we introduce the running average version that still requires $\gamma(\cdot)$. Then we introduce the component of estimated $\hat{\gamma}(\cdot)$. Finally, we relax the i.i.d. assumption and extend the results to the Markov delay models. Four other combinations are not considered in this work and they correspond to the "– – –" parts in Table I. If desired, our approaches can be easily applied to those settings as well.

Remark 6: The online scheme using the KNN algorithm described in this section works well in simulation, as will be shown in Sec. VIII-B. The reason why we do not have an analytical proof of convergence is the lack of martingales due to the memory property inherent in the Markov delay process (see Lemma 7 in Appendix A where martingales can be constructed in the case of *i.i.d.* delay).

VIII. Simulation Results

A. I.I.D. Delay

In this section, we consider *i.i.d.* $\{(Y_i, Z_i) : i \geq 1\}$ with Y_i and Z_i being independent log-normal random variables with $(\mu_Y, \sigma_Y^2) = (0.5, 0.25)$ and $(\mu_Z, \sigma_Z^2) = (0.5, 0.5)$. We consider the quadratic AoI penalty function $\gamma_{\text{qdr}}(\Delta) = \Delta^2$.

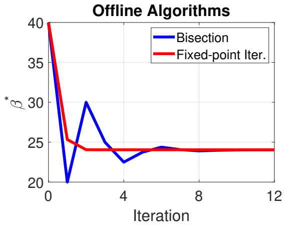


Fig. 3: Offline computation for β^* under the i.i.d. delay.

The trajectories of the *offline* fixed-point computation $\beta_{i+1} = f_\Gamma(\beta_i)$, described in Sec. IV-B, versus the bisection method are plotted in Fig. 3. The advantage of our scheme is twofold. Firstly it converges faster than the bisection method. Secondly, as proved in Proposition 3, the sequence $\{\beta_i\}$ is non-increasing and thus does not fluctuate as in the case of the bisection search.

We also run the *fixed-point-iteration-based* online algorithm $\Xi^{\text{SY}}_{\text{qdr}}$. Fig. 4a plots the evolution of β_i versus i and benchmarks β_i against β^* (the red dashed line). The three curves in Fig. 4a are generated by different random seeds. For each curve, β_i is within 8% of β^* after just 10^3 iterations. Since it is an online algorithm, *it means that using our distribution-oblivious scheme, after sending just 1000 update packets, the average AoI penalty of the underlying system (over the last 1000 packets) is already within* 8% *of the best offline solution that requires complete knowledge of the delay distributions.* The gap is less than 4% after 10^4 iterations.

TABLE I: Summary of the fixed-point-based online algorithms. Each of the fixed-point-based algorithm Ξ has a stochastic approximation Robbins-Monro-based counterpart, denoted by the symbol Λ (instead of Ξ).

	Derived under i.i.d. delay		Derived under Markov delay	
	Requiring statistics SY_γ	Estimating statistics SY_γ	Requiring statistics SY_γ	Estimating statistics SY_γ
γ	• $\phi^{\text{SY}}_{\text{lin},\beta_i}$, $\phi^{\text{SY}}_{\text{qdr},\beta_i}$, $\phi^{\text{SY}}_{\text{exp},\beta_i}$ derived according to (29), (37), (35) • Policies: $\Xi^{\text{SY}}_{\text{lin}}$, $\Xi^{\text{SY}}_{\text{qdr}}$, $\Xi^{\text{SY}}_{\text{exp}}$	• $\phi^{\text{RA}}_{\text{lin},\beta_i}$, $\phi^{\text{RA}}_{\text{qdr},\beta_i}$, $\phi^{\text{RA}}_{\text{exp},\beta_i}$ use running emp. avg. to replace SY_γ, according to Sec. VI-A • Policies: $\Xi^{\text{RA}}_{\text{lin}}$, $\Xi^{\text{RA}}_{\text{qdr}}$, $\Xi^{\text{RA}}_{\text{exp}}$	- - - - - - - - -	- - - - - - - - - - - - -
$\hat{\gamma}$	- - - - - - - - - - - - - - -	• $\phi^{\text{RA}}_{\hat{\gamma},\beta_i}$ estimates $\gamma(\cdot)$ and uses running emp. avg. according to (49) • Policy: $\Xi^{\text{RA}}_{\hat{\gamma}}$	- - - - - - - - -	• $\phi^{\text{KNN}}_{\hat{\gamma},\beta_i}$ estimates $\gamma(\cdot)$ and uses KNN according to (54) • Policy: $\Xi^{\text{KNN}}_{\hat{\gamma}}$

Note that Fig. 4a traces the evolution of the β_i computed by $\Xi^{\text{SY}}_{\text{qdr}}$. The value of β_i is then fed into (11) to compute the waiting time X_i. Fig. 4a shows that β_i converges to the optimal choice β^* but does not evaluate how close the empirical AoI penalty, resulting from these choices β_i, is to the optimal/minimal AoI penalty. To directly examine the penalty performance, we compute the *observed avg. AoI penalty* $\frac{\int_0^{D_i}\gamma(t)dt}{D_i}$ for every $i \geq 1$. The red horizontal dashed line is the AoI penalty achieved by the best possible offline algorithm. Similar to Fig. 4a, the observed avg. AoI penalty is within 7% of β^* after just 10^3 iterations, and the difference is less than 3% after 10^4 iterations.

Next we run the *root-finding-based* online algorithm $\Lambda^{\text{SY}}_{\text{qdr}}$ where we choose the step-size to be $\frac{0.5}{i}$ for the i-th update. The results are presented in Figs. 4c and 4d. Compared with the fixed-point-iteration-based scheme $\Xi^{\text{SY}}_{\text{qdr}}$, β_i using $\Lambda^{\text{SY}}_{\text{qdr}}$ is generally higher for the first 1000 iterations (see Fig. 4a versus Fig. 4c) and eventually converges to the optimal β^*. We also directly examine the resulting empirical AoI in Fig. 4d. Even though the β_i chosen by the Robbins-Monro algorithm is generally larger, their impact on the average AoI performance is not significant. That is, after the 100 iterations, the empirical AoI of both $\Xi^{\text{SY}}_{\text{qdr}}$ and $\Lambda^{\text{SY}}_{\text{qdr}}$ are very close to each other. This relative insensitivity to the β_i choice could be explained as follows. Recall that both Ξ^{SY} and Λ^{SY} choose their waiting time X_i based on the same water-filling rule (11). Therefore, for a wide range of β_i we will choose to *zero-wait* $\phi^{\text{SY}}_{\text{qdr},\beta_i}(y',z') = 0$ when y' and z' in (11) are large. As a result, different β_i values have impacts only in the scenarios of small $(Y_{i-1}, Z_{i-1}) = (y', z')$ and thus the actual AoI penalty performance is not very sensitive to the β_i value.

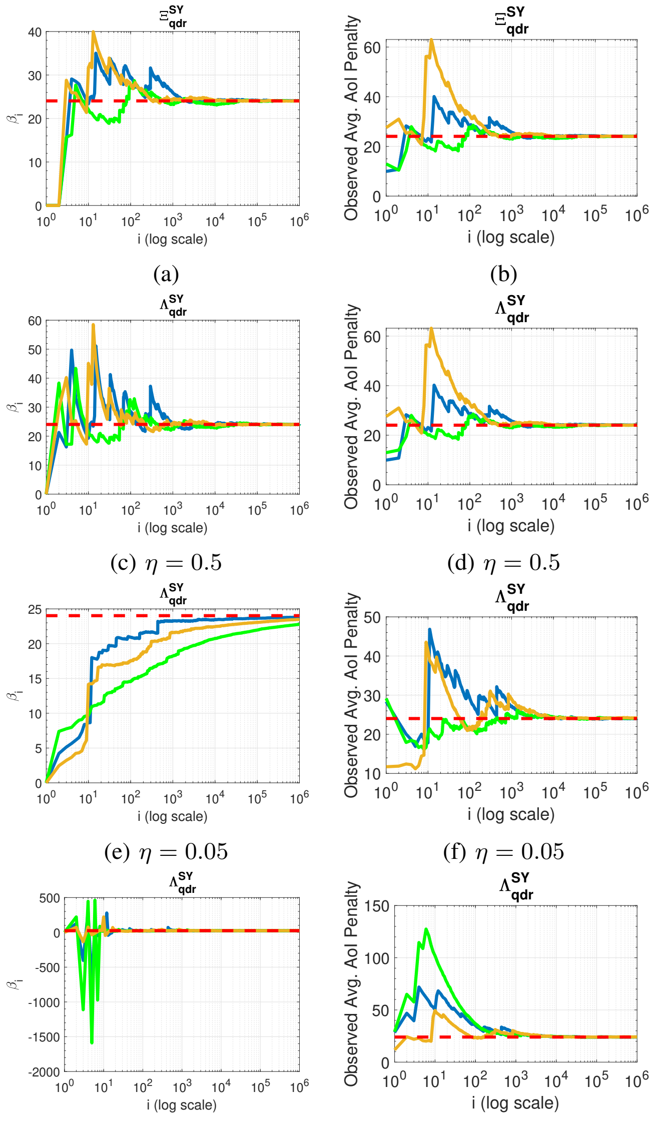
However, the step-size $\frac{\eta}{i}$ used in the Robbins-Monro-based online algorithm $\Lambda^{\text{SY}}_{\text{qdr}}$ has to be carefully determined to balance the convergence speed and numerical stability. For example, if we set η to be 0.05, then β_i is not always convergent to β^* even after 10^6 iterations (see Fig. 4e). On the other hand, if $\eta = 1.5$ is picked, then a large β_i is sometimes observed in our simulation (see β_i fluctuate between $[-1500, 500]$ in Fig. 4g). If $\beta_{\text{UB}} < \infty$, then any β_i that is strictly larger than β_{UB} immediately results in an infinite waiting time $X_i = \phi_{\Gamma,\beta_i}(\cdot,\cdot)$ that suspends future control actions. This is why we need the non-trivial assumption that $\beta_{\text{UB}} = \infty$ for the Robbins-Monro-based online algorithm. The fixed-point-based online algorithm, however, does not have such an issue. Also see the discussion in Sec. V-G.

Since the fixed-point-iteration-based and root-finding-based online algorithms (with a carefully chosen $\frac{\eta}{i}$) achieve similar performance, for the rest of the paper we will only present the results of the *fixed-point-iteration-based* online algorithms.

We then consider the case of using the running average-based online algorithm $\Xi^{\text{RA}}_{\text{qdr}}$. We set $N_{\text{RA}} = 10^3$ and the results are plotted in Figs. 5a and 5b. Comparing $\Xi^{\text{SY}}_{\text{qdr}}$ and $\Xi^{\text{RA}}_{\text{qdr}}$, we observe that for the first 100 iterations, with an insufficient number of samples, $\Xi^{\text{RA}}_{\text{qdr}}$ does not have an accurate estimate of SY_γ, and hence the resulting β_i is slightly higher than $\Xi^{\text{SY}}_{\text{qdr}}$ (see Figs. 4a and 5a). However, the slightly higher parameter β_i choices do not impact much on the actual empirical avg. AoI penalty. The curves in Figs. 4b and 5b are almost identical. After the first 100 iterations, both $\Xi^{\text{SY}}_{\text{qdr}}$ and $\Xi^{\text{RA}}_{\text{qdr}}$ have similar β_i and empirical AoI penalty, and eventually converge to the optimal value. This confirms the benefits of using the running empirical average as a substitute of the delay statistics SY_γ.

A more interesting scenario is when $\gamma(\cdot)$ is unknown. In our simulation, the destination observes $q_i = \gamma(p_i) + n_i$ where $p_i = Y_{i-1} + Z_{i-1} + X_i + Y_i$. We assume $\gamma(\Delta) = \Delta^2$ but this fact is unknown to the source/destination pair, and we also assume the observation error n_i's are i.i.d. Gaussian random variables with mean zero and variance 0.05. Other parameters are set as follows: $N_{\hat{\gamma}} = 10^3$, $N_{\text{sum}} = 200$ and $N_{\text{trapezoid}} = 10^3$. Fig. 6a plots the resulting $\hat{\gamma}(\cdot)$ and the true underlying $\gamma(\cdot)$ (those scattered red points are the elements in the set $\mathcal{S}_{\hat{\gamma}}$). As can be seen from Figs. 6a and 6b (magnified version), $\hat{\gamma}(\cdot)$ is non-decreasing and sufficiently close to $\gamma(\cdot)$. At the end of iteration ($i = 10^6$), β_i from every curve in Fig. 6c is within 4% of β^*. Meanwhile, because the actual AoI penalty performance is less sensitive to the β_i value, the observed avg. AoI penalty is within 1% of the best offline algorithm that knows both the distribution $\mathbb{P}_Y$ and the $\gamma(\cdot)$. This demonstrates the superior performance of $\Xi^{\text{RA}}_{\hat{\gamma}}$, which is not only distribution-oblivious, but it is also able to estimate $\gamma(\cdot)$ in an online manner.

Finally, even though $\Xi^{\text{KNN}}_{\hat{\gamma}}$ is originally derived assuming Markov delay (see Sec. VII and Table I), we use it here to examine its performance under the i.i.d. delay scenario. The associated parameters are set as follows: $N_{\hat{\gamma}} = 10^3$, $N_{\text{KNN}} = 100$ and $N_{\text{trapezoid}} = 10^3$. At $i = 10^6$ iteration, for each random seed β_i is 15% away from β^*, see Fig. 7a. The reason is that with N_{KNN} set to be a small number 100, the estimation of the expectations is not as accurate as the running average scheme (which has $N_{\text{sum}} = 200$ observations), which leads to larger error of β_i, see see Figs. 5a and 7a. Nonetheless, the observed avg. AoI penalty is only 3% away from the offline optimum (see Fig. 7b), which shows the effectiveness and robustness of $\Xi^{\text{KNN}}_{\hat{\gamma}}$ even under the i.i.d. delay setting.



(a) (b)

(c) $\eta = 0.5$ (d) $\eta = 0.5$

(e) $\eta = 0.05$ (f) $\eta = 0.05$

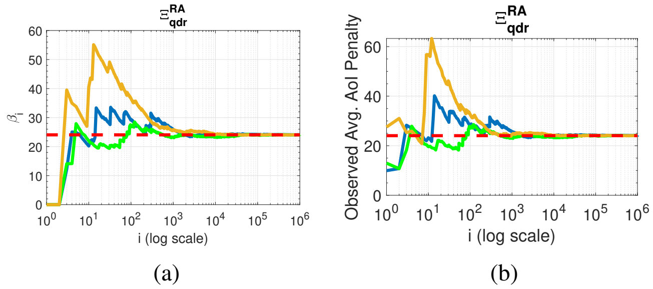
(g) $\eta = 1.5$ (h) $\eta = 1.5$

Fig. 4: Left: Evolution of β_i using the online algorithms Right: Evolution of the observed AoI penalty using the online algorithms Different curves represent different random seeds. The horizontal red dashed line represents β^*.

B. Markov Delay

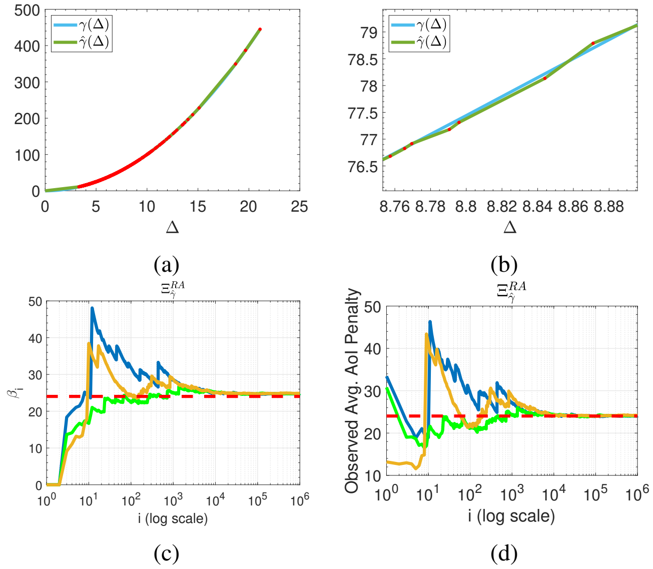
In this section, we simulate *Markov* $\{(Y_i, Z_i) : i \geq 1\}$. Specifically, a stationary discrete Markov chain is considered: We set $\mathbb{P}(Z_i = 1) = 1$, $\mathbb{P}(Y_i = 1) = \mathbb{P}(Y_i = 2) = \mathbb{P}(Y_i = 3) = \frac{1}{3}$ and the transition matrix for Y_i is

$$\begin{bmatrix} 0.95 & 0.025 & 0.025 \\ 0.025 & 0.95 & 0.025 \\ 0.025 & 0.025 & 0.95 \end{bmatrix}. \tag{56}$$



(a) (b)

Fig. 5: Simulation results using $\Xi^{\text{RA}}_{\text{qdr}}$ under the i.i.d. delay.



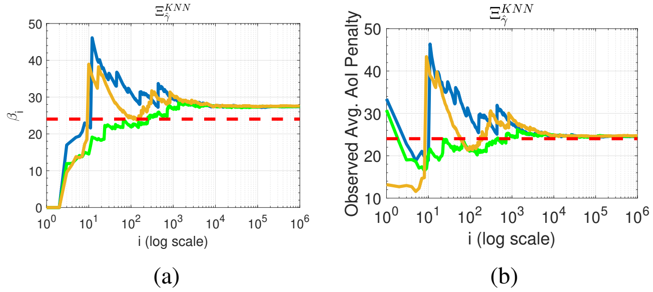
(a) (b)

(c) (d)

Fig. 6: Simulation results using $\Xi^{\text{RA}}_{\hat{\gamma}}$ under the i.i.d. delay.

The exponential AoI penalty function $\gamma_{\exp}(\Delta) = e^{2\Delta} - 1$ is considered. The offline optimal hitting time threshold β^* and the corresponding optimal average AoI penalty is found by the fixed-point computation $\beta_{i+1} = f_\Gamma(\beta_i)$ in Sec. IV-B.

Next, we examine the performance degradation when the delay process is *Markov*, but the source *wrongly believes that the process is i.i.d.* That is, we run the best i.i.d.-delay-assuming online algorithm $\Xi^{\text{SY}}_{\text{exp}}$ while directly feeding the true value of $\text{SY}_\gamma = \mathbb{E}\{e^{aY}\}$ to the algorithm, and plot the trace of the observed avg. AoI penalty. Here we do not plot the trace of β_i for $\Xi^{\text{SY}}_{\text{exp}}$ scheme anymore. The reason is that even if



(a) (b)

Fig. 7: Simulation results using $\Xi^{\text{KNN}}_{\hat{\gamma}}$ under the i.i.d. delay.

there is a genie that gives $\Xi^{\text{SY}}_{\text{exp}}$ the ideal β^* value, the scheme will still compute a suboptimal X_i since when computing X_i, the i.i.d.-delay-assuming scheme (incorrectly) takes the expectation of the marginal distribution $\mathbb{P}_Y$ in (11) and (12) while an optimal Markov-delay-assuming scheme, given the ideal β^* value, would take the expectation over the conditional distribution $\mathbb{P}_{Y_i|Y_{i-1},Z_{i-1}}$ instead. As a result, how close β_i is to β^* has little indication of how good the performance of the i.i.d.-based $\Xi^{\text{SY}}_{\text{exp}}$ scheme is when applied to a Markov setting. The only meaningful metric is to directly measure the observed avg. AoI penalty of different schemes. As shown in Fig. 8, at the end of iteration the avg. AoI penalty of the i.i.d.-based $\Xi^{\text{SY}}_{\text{exp}}$ is 11% away from optimal offline Markov scheme.

We then run the online algorithm $\Xi^{\text{KNN}}_{\hat{\gamma}}$ with the associated parameters $N_{\hat{\gamma}} = 10^3$, $N_{\text{KNN}} = 100$ and $N_{\text{trapezoid}} = 10^3$. Without knowing that we are dealing with Markovian delay and without the knowledge of the penalty function $\gamma(\cdot)$, our scheme $\Xi^{\text{KNN}}_{\hat{\gamma}}$ performs extremely well. Fig. 9a plots the resulting $\hat{\gamma}(\cdot)$ and the true underlying $\gamma(\cdot)$ (those scattered red points are the elements in the set $\mathcal{S}_{\hat{\gamma}}$). This time, the estimator $\hat{\gamma}(\cdot)$ automatically adapts to a different underlying $\gamma(\cdot)$. The scheme $\Xi^{\text{KNN}}_{\hat{\gamma}}$ leads to 11% higher β_i compared with β^* (see Fig. 9c) while the observed avg. AoI penalty, arguably the more important metric, is within 2% of best possible offline solution (see Fig. 9d). This again shows the strength of the online algorithm $\Xi^{\text{KNN}}_{\hat{\gamma}}$.

C. *Bounded $\gamma(\cdot)$ for the OU Process*

One critical difference between the fixed-point-based Algorithm 1 and the Robbins-Monro-based Algorithm 2 is that the former is capable of handling bounded $\gamma(\cdot)$ while the latter is not. In this section, we consider the bounded $\gamma_{\text{OU}}(t) = \frac{\sigma^2}{2\theta}(1 - e^{-2\theta t})$ corresponding to signal-agnostic sampling for the OU process [21] and the corresponding $\beta_{\text{UB}} = \lim_{\Delta\to\infty} \gamma(\Delta) = \frac{\sigma^2}{2\theta} < \infty$.

Following the same manner in Sec.V-C, we use $\phi^{\text{SY}}_{\text{OU},\beta}(y', z')$ to denote the waiting time function $\phi_{\Gamma,\beta}(y', z')$ specialized for the OU-process penalty. Similarly, $g^{\text{SY}}_{\text{OU},1}(y', z', \beta)$ denotes the empirical AoI penalty function $g_1(y', z', \beta)$ specialized for the OU-process $\gamma_{\text{OU}}(\cdot)$. Applying simple calculus to (3), (4), (11), and (19) shows that for $\beta < \beta_{\text{UB}} = \frac{\sigma^2}{2\theta}$, we have

$$\begin{aligned}&\phi^{\text{SY}}_{\text{OU},\beta}(y', z')\\&= \max\left(\frac{1}{2\theta}\ln\left(\frac{\mathbb{E}\{e^{-2\theta Y}\}}{1 - \frac{2\theta}{\sigma^2}\beta}\right) - y' - z', 0\right) \quad (57)\end{aligned}$$

$$\begin{aligned}&g^{\text{SY}}_{\text{OU},1}(y', z', \beta) = \frac{\sigma^2}{2\theta}\left(y' + z' + \phi^{\text{SY}}_{\text{OU},\beta}(y', z')\right)\\&- (\frac{\sigma}{2\theta})^2 \cdot \left(1 - e^{-2\theta\left(y'+z'+\phi^{\text{SY}}_{\text{OU},\beta}(y',z')\right)}\right) \cdot \mathbb{E}\{e^{-2\theta Y}\}. \quad (58)\end{aligned}$$

From (57) and (58), it is clear that to calculate $\phi^{\text{SY}}_{\text{OU},\beta}(y', z')$ and $g^{\text{SY}}_{\text{OU},1}(y', z', \beta)$, the only statistical knowledge we need is a scalar $\text{SY}_\gamma = \mathbb{E}\{e^{-2\theta Y}\}$, which can be well estimated in practice. $\Xi^{\text{SY}}_{\text{OU}}$ denotes Algorithm 1 when specialized for the OU-process AoI Penalty function $\gamma_{\text{OU}}(\Delta) = \frac{\sigma^2}{2\theta}(1 - e^{-2\theta\Delta})$.

We consider the same log-normal delay as in Sec. VIII-A and set $\sigma = 4$ and $\theta = 0.5$. The simulation results running $\Xi^{\text{SY}}_{\text{OU}}$ are presented in Figs. 10a and 10b.

As shown in Fig. 10a, we always have $\beta_i < \beta_{\text{UB}} = 16$ (as proved by Lemma 6), which demonstrates the applicability of Algorithm 1 under the bounded $\gamma(\cdot)$. Moreover, compared with the unbounded $\gamma_{\text{qdr}}(\cdot)$ (Figs. 4a and 4b), the convergence rate of Algorithm 1 for the bounded γ_{OU} seems even faster. Specifically, β_i is already within 1% of β^* after just 10 iterations, and the difference is less than 0.4% after 10^2 iterations. The observed avg. AoI penalty (Fig. 10b) is within 1.3% of offline optimum after 10^2 iterations.

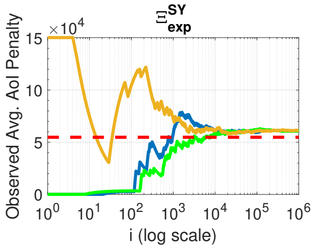
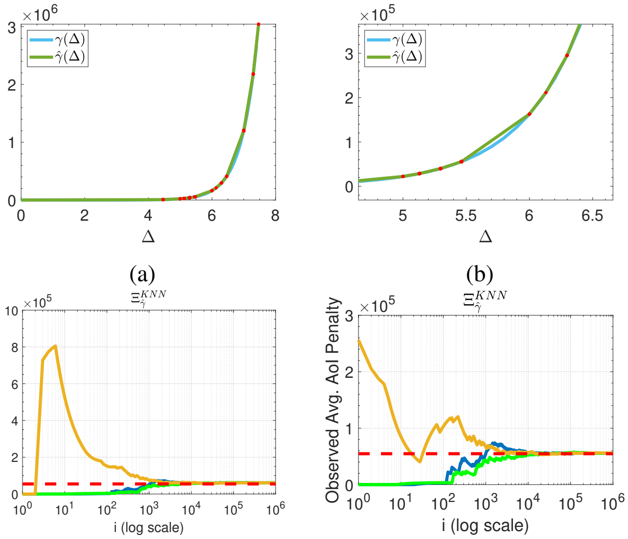


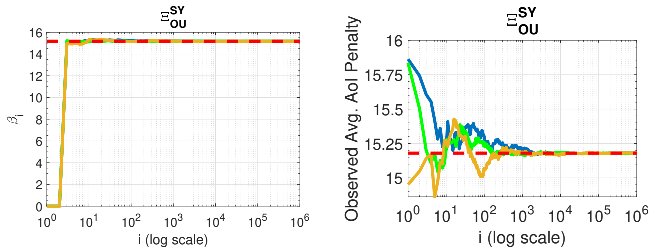
Fig. 8: Simulation results using $\Xi^{\text{SY}}_{\text{exp}}$ under the Markov delay.



(a) (b)

(c) (d)

Fig. 9: Simulation results using $\Xi^{\text{KNN}}_{\hat{\gamma}}$ under the Markov delay.



(a) (b)

Fig. 10: Simulation results using the bounded $\gamma_{\text{OU}}(\cdot)$ corresponding to the OU process.

IX. Conclusion

We have studied the AoI minimization problem based on a new fixed-point-based framework, and derived the corresponding optimal waiting policy. We have also developed the first provably optimal distribution-oblivious online algorithms on AoI minimization for arbitrary AoI penalty functions, which may be bounded or unbounded. Additionally, we have addressed several practical issues in the i.i.d. delay and Markov delay settings, including proposing an effective solution to estimating the AoI penalty function $\gamma(\cdot)$ using monotonic regression. Simulation results verify the effectiveness of the proposed schemes.

Appendix A
Proof of (41) in Proposition 4

Here we provide proof sketches for (41) in Proposition 4. The full proof can be found in the Appendix H of the supplementary material. We define two random processes M_i and N_i as follows. Set $M_0 = N_0 = 0$ and for any $i \geq 1$,

$$M_i = \sum_{j=1}^{i-1} \left(g_1(Y_{j-1}, Z_{j-1}, \beta_j) - \overline{g}_1(\beta_j)\right) \tag{59}$$

$$N_i = \sum_{j=1}^{i-1} \left(g_2(Y_{j-1}, Z_{j-1}, \beta_j) - \overline{g}_2(\beta_j)\right). \tag{60}$$

Define $\mathcal{F}_i \triangleq \{(Y_j, Z_j) : j \leq i-2\}$ as the set of all the previous forward and backward channel delays up to the $(i-2)$-th packet.

Lemma 7: $\{M_i\}$ and $\{N_i\}$ are martingales with respect to $\mathcal{F}_i$.

Proof: First, since $\{Y_i\}$, $\{Z_i\}$ and $\{\beta_i\}$ are all bounded, we have $\mathbb{E}\{|M_i|\} < \infty$ and $\mathbb{E}\{|N_i|\} < \infty$. We then have

$$\begin{aligned}
&\mathbb{E}\{M_i - M_{i-1} | \mathcal{F}_{i-1}\} \\
&= \mathbb{E}\{g_1(Y_{i-2}, Z_{i-2}, \beta_{i-1}) - \overline{g}_1(\beta_{i-1}) | \mathcal{F}_{i-1}\} \qquad (61) \\
&= \mathbb{E}\{g_1(Y_{i-2}, Z_{i-2}, \beta_{i-1}) | \mathcal{F}_{i-1}\} - \overline{g}_1(\beta_{i-1}) = 0 \qquad (62)
\end{aligned}$$

where the first equality in (62) follows from the fact that β_{i-1} is completely determined by $\mathcal{F}_{i-1}$ (see (24) and the definition of $\mathcal{F}_i$ in the above); and the second equality in (62) follows from $\{(Y_{i-2}, Z_{i-2})\}$ being i.i.d. and independent of $\mathcal{F}_{i-1}$. Similar reasoning gives $\mathbb{E}\{N_i - N_{i-1} | \mathcal{F}_{i-1}\} = 0$. ■

Lemma 8: For $\alpha > 0$ and for all $i \geq 1$, there exist two positive constants $k_1, k_2 > 0$ such that (63) and (64) hold.

$$\mathbb{P}\left(\sum_{j=1}^{i} g_1(Y_{j-1}, Z_{j-1}, \beta_j) < -i^{(0.5+\alpha)} + \sum_{j=1}^{i} \overline{g}_1(\beta_j)\right) \leq \exp\left(\frac{-i^{2\alpha}}{2(k_1)^2}\right) \tag{63}$$

$$\mathbb{P}\left(\sum_{j=1}^{i} g_2(Y_{j-1}, Z_{j-1}, \beta_j) > i^{(0.5+\alpha)} + \sum_{j=1}^{i} \overline{g}_2(\beta_j)\right) \leq \exp\left(\frac{-i^{2\alpha}}{2(k_2)^2}\right). \tag{64}$$

Though admitting a complicated form, the intuition of Lemma 8 is simple. Because M_i and N_i are martingales, the growth rates of both M_i and N_i should be within $\pm i^{0.5+\alpha}$ with close-to-one probability. Nonetheless because we only need one side of it, we bound the probability of M_i being too small in (63) and bound the probability of N_i being too large in (64). Since $\{M_i\}$ and $\{N_i\}$ are martingales, by Azuma's inequality [26], there exist positive constants k_1 and k_2 such that (63) and (64) hold. See the supplementary material for the full proof.

Lemma 9: For $\alpha > 0$ and $i \geq 1$, there exists a positive constant $k_3 > 0$ such that

$$\mathbb{P}\left(\frac{\sum_{j=1}^{i} g_1(Y_{j-1}, Z_{j-1}, \beta_j)}{\sum_{j=1}^{i} g_2(Y_{j-1}, Z_{j-1}, \beta_j)} < \frac{-i^{(0.5+\alpha)} + \sum_{j=1}^{i} \overline{g}_1(\beta_j)}{i^{(0.5+\alpha)} + \sum_{j=1}^{i} \overline{g}_2(\beta_j)}\right) \leq 2\exp\left(\frac{-i^{2\alpha}}{2(k_3)^2}\right). \tag{65}$$

Proof: We first define 3 events.

$$A_1 \triangleq \left\{\sum_{j=1}^{i} g_1(Y_{j-1}, Z_{j-1}, \beta_j) < -i^{(0.5+\alpha)} + \sum_{j=1}^{i} \overline{g}_1(\beta_j)\right\}$$

$$A_2 \triangleq \left\{\sum_{j=1}^{i} g_2(Y_{j-1}, Z_{j-1}, \beta_j) > i^{(0.5+\alpha)} + \sum_{j=1}^{i} \overline{g}_2(\beta_j)\right\}$$

$$A \triangleq \left\{\frac{\sum_{j=1}^{i} g_1(Y_{j-1}, Z_{j-1}, \beta_j)}{\sum_{j=1}^{i} g_2(Y_{j-1}, Z_{j-1}, \beta_j)} < \frac{-i^{(0.5+\alpha)} + \sum_{j=1}^{i} \overline{g}_1(\beta_j)}{i^{(0.5+\alpha)} + \sum_{j=1}^{i} \overline{g}_2(\beta_j)}\right\}$$

Eq. (65) is equivalent to $\mathbb{P}(A) \leq 2\exp\left(\frac{-i^{2\alpha}}{2\cdot(k_3)^2}\right)$.

We then observe that $(A_1)^c \cap (A_2)^c \subseteq A^c$, which implies $\mathbb{P}(A) \leq P(A_1 \cup A_2)$. By the union bound and by choosing $k_3 = \max(k_1, k_2)$, we have completed the proof. ■

Lemma 10: Given any $\alpha \in (0, 0.5)$, define

$$I_1 \triangleq \left\lceil \overline{g}_1(0)^{\left(\frac{-1}{0.5-\alpha}\right)} \right\rceil. \tag{66}$$

Then, for all $i \geq I_1$, we have $\overline{g}_1(0) \cdot i \geq i^{(0.5+\alpha)}$, which follows from the fact that $0.5 + \alpha < 1$.

Lemma 11: Given any $\alpha \in (0, 0.5)$, for $i \geq I_1$,

$$\mathbb{P}\left(\frac{\sum_{j=1}^{i} g_1(Y_{j-1}, Z_{j-1}, \beta_j)}{\sum_{j=1}^{i} g_2(Y_{j-1}, Z_{j-1}, \beta_j)} < \frac{\sum_{j=1}^{i} \overline{g}_1(\beta_j)}{\sum_{j=1}^{i} \overline{g}_2(\beta_j)} - \frac{i^{-(0.5-\alpha)}}{\overline{g}_2(0)} - \frac{\overline{g}_1(\beta_{\max}) \cdot i^{-(0.5-\alpha)}}{(\overline{g}_2(0))^2}\right) \leq 2\exp\left(\frac{-i^{2\alpha}}{2(k_3)^2}\right). \tag{67}$$

The proof is a directly combination of Lemmas 9 to 10. Since

$$\beta^* = \min_{\beta} \frac{\overline{g}_1(\beta)}{\overline{g}_2(\beta)} \leq \frac{\overline{g}_2(\beta_j)}{\overline{g}_2(\beta_j)} \tag{68}$$

for any β_j (see the discussions in Proposition 1 and Corollary 1), we must have

$$\beta^* \leq \frac{\sum_{j=1}^{i} \overline{g}_1(\beta_j)}{\sum_{j=1}^{i} \overline{g}_2(\beta_j)}. \tag{69}$$

Continuing from (67) and (69), we have

Lemma 12: Given any $\alpha \in (0, 0.5)$, for $i \geq I_1$, there exists a constant k_4 such that

$$\mathbb{P}\left(\frac{\sum_{j=1}^{i} g_1(Y_{j-1}, Z_{j-1}, \beta_j)}{\sum_{j=1}^{i} g_2(Y_{j-1}, Z_{j-1}, \beta_j)} < \beta^* - k_4 \cdot i^{-(0.5-\alpha)}\right) \leq 2\exp\left(\frac{-i^{2\alpha}}{2(k_3)^2}\right) \quad (70)$$

Ineq. (70) follows directly from (67) and (69) by setting

$$k_4 \triangleq \frac{1}{\overline{g}_2(0)} + \frac{\overline{g}_1(\beta_{\max})}{(\overline{g}_2(0))^2}. \quad (71)$$

By the β_i update rule in (24), Lemma 12 can be rewritten as

$$\mathbb{P}\left(\beta_{i+1} < \beta^* - k_4 \cdot i^{-(0.5-\alpha)}\right) \leq 2\exp\left(\frac{-i^{2\alpha}}{2(k_3)^2}\right), \quad \forall i > I_1. \quad (72)$$

The detailed proofs of Lemmas 10 to 12 can be found in the appendices of the supplementary material. A quick recap of the intuition is as follows. Because M_i and N_i are martingales, the empirical observations $\sum_{j=1}^{i} g_k(Y_{j-1}, Z_{j-1}, \beta_j)$ must concentrate around its mean for $k = 1$ and 2, see Lemmas 9 to 11. Therefore, their ratio, which is the iteratively computed parameter value β_{i+1}, must not be too far below the optimal β^*, see Lemma 12.

Finally, given any $\alpha \in (0, 0.5)$, we set the positive constants in (41) in the following way:

$$c_1 \triangleq k_4 = \frac{1}{\overline{g}_2(0)} + \frac{\overline{g}_1(\beta_{\max})}{(\overline{g}_2(0))^2} \quad (73)$$

$$c_2 \triangleq \max\left(2, \exp\left(c_3 \cdot (I_1)^{2\alpha}\right)\right) \quad (74)$$

$$c_3 \triangleq \frac{1}{2(k_3)^2}. \quad (75)$$

The above specific choices of c_1 to c_3 plus the inequality (72), we have proven (41).

References

[1] C.-H. Tsai and C.-C. Wang, "Age-of-Information revisited: Two-way delay and distribution-oblivious online algorithm," in *2020 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2020, pp. 1782–1787.

[2] A. V. Dastjerdi and R. Buyya, "Fog computing: Helping the Internet of Things realize its potential," *Computer*, vol. 49, no. 8, pp. 112–116, 2016.

[3] X. Song and J. W.-S. Liu, "Performance of multiversion concurrency control algorithms in maintaining temporal consistency," in *Proceedings., Fourteenth Annual International Computer Software and Applications Conference*. IEEE, 1990, pp. 132–139.

[4] 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.

[5] Y. Sun and B. Cyr, "Sampling for data freshness optimization: Non-linear age functions," *Journal of Communications and Networks*, vol. 21, no. 3, pp. 204–219, 2019.

[6] C.-H. Tsai and C.-C. Wang, "Unifying AoI minimization and remote estimation—–Optimal sensor/controller coordination with random two-way delay," in *IEEE INFOCOM 2020-IEEE Conference on Computer Communications*. IEEE, 2020, pp. 466–475.

[7] ——, "Unifying AoI minimization and remote estimation—Optimal sensor/controller coordination with random two-way delay," *IEEE/ACM Transactions on Networking*, vol. 30, no. 1, pp. 229–242, 2022.

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

[9] C.-H. Tsai and C.-C. Wang, "Jointly minimizing AoI penalty and network cost among coexisting source-destination pairs," in *2021 IEEE International Symposium on Information Theory (ISIT)*, 2021, pp. 3255–3260.

[10] L. Huang and E. Modiano, "Optimizing Age-of-Information in a multi-class queueing system," in *2015 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2015, pp. 1681–1685.

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

[12] M. Costa, M. Codreanu, and A. Ephremides, "Age of Information with packet management," in *2014 IEEE International Symposium on Information Theory*. IEEE, 2014, pp. 1583–1587.

[13] H. Pucha, Y. Zhang, Z. M. Mao, and Y. C. Hu, "Understanding network delay changes caused by routing events," in *ACM SIGMETRICS Performance Evaluation Review*, vol. 35, no. 1. ACM, 2007, pp. 73–84.

[14] E. T. Ceran, D. Gündüz, and A. György, "Reinforcement learning to minimize Age of Information with an energy harvesting sensor with HARQ and sensing cost," in *IEEE INFOCOM 2019-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 2019, pp. 656–661.

[15] C. Kam, S. Kompella, and A. Ephremides, "Learning to sample a signal through an unknown system for minimum AoI," in *IEEE INFOCOM 2019-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 2019, pp. 177–182.

[16] E. U. Atay, I. Kadota, and E. Modiano, "Aging bandits: Regret analysis and order-optimal learning algorithm for wireless networks with stochastic arrivals," *arXiv preprint arXiv:2012.08682*, 2020.

[17] H. Tang, Y. Chen, J. Sun, J. Wang, and J. Song, "Sending timely status updates through channel with random delay via online learning," in *IEEE INFOCOM 2022-IEEE Conference on Computer Communications*. IEEE, 2022, pp. 1819–1827.

[18] S. Fatale, K. Bhandari, U. Narula, S. Moharir, and M. K. Hanawal, "Regret of Age-of-Information bandits," *IEEE Transactions on Communications*, vol. 70, no. 1, pp. 87–100, 2021.

[19] R. D. Yates, "Lazy is timely: Status updates by an energy harvesting source," in *2015 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2015, pp. 3008–3012.

[20] C.-C. Wang, "How useful is delayed feedback in AoI minimization—–A study on systems with queues in both forward and backward directions," in *2022 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2022, pp. 3192–3197.

[21] T. Z. Ornee and Y. Sun, "Sampling and remote estimation for the Ornstein-Uhlenbeck process through queues: Age of Information and beyond," *IEEE/ACM Transactions on Networking*, vol. 29, no. 5, pp. 1962–1975, 2021.

[22] H. Robbins and S. Monro, "A stochastic approximation method," *The Annals of Mathematical Statistics*, pp. 400–407, 1951.

[23] J. R. Blum *et al.*, "Approximation methods which converge with probability one," *The Annals of Mathematical Statistics*, vol. 25, no. 2, pp. 382–386, 1954.

[24] M. J. Best and N. Chakravarti, "Active set algorithms for isotonic regression: A unifying framework," *Mathematical Programming*, vol. 47, no. 1, pp. 425–439, 1990.

[25] N. S. Altman, "An introduction to kernel and nearest-neighbor non-parametric regression," *The American Statistician*, vol. 46, no. 3, pp. 175–185, 1992.

[26] K. Azuma, "Weighted sums of certain dependent random variables," *Tohoku Mathematical Journal, Second Series*, vol. 19, no. 3, pp. 357–367, 1967.