Optimal AoI for Systems With Queueing Delay in Both Forward and Backward Directions

Chih-Chun Wang, *Fellow, IEEE*

***Abstract*— Age-Of-Information (AoI) is a metric that focuses directly on the application-layer objectives, and a canonical AoI minimization problem is the *update-through-queues* models. Existing results in this direction fall into two categories: The open-loop setting for which the sender is oblivious of the packet departure time, versus the closed-loop setting for which the decision is based on *instantaneous* Acknowledgment (ACK). Neither setting perfectly reflects modern networked systems, which almost always rely on *feedback that experiences some delay.* Motivated by this observation, this work subjects the ACK traffic to a second queue so that the closed-loop decision is made based on delayed feedback. Near-optimal schedulers have been devised, which smoothly transition from the instantaneous-ACK to the open-loop schemes depending on how long the feedback delay is. The results quantify the benefits of delayed feedback for AoI minimization in the update-through-queues systems.**

***Index Terms*— Age-of-information, semi-Markov decision process, two-way delay, network scheduling, update through queues.**

I. Introduction

SUPPORTING low-latency applications is a top mission of modern communication networks. One example application is remote control in cyber-physical systems (CPS). E.g., [2] studies *linear quadratic Gaussian* (LQG) control systems with random communication delay. The results show that the control performance deteriorates exponentially fast with respect to the *Age of* (the measurement) *Information* (AoI). The intuition is that any control action at time t based on measurements that are Δ-time old inevitably leaves the state disturbance accumulated during time interval $(t-\Delta, t]$ unchecked. This usually incurs exponential cost $e^{c\cdot\Delta}$ since for an *inherently unstable system*, the system state drifts exponentially away in time if left unchecked.

By exploring the connections between the staleness of the data and the efficacy of the control, many existing results have established strong relationships between AoI and the underlying system performance [3], [4], [5]. AoI minimization has since attracted significant research attentions on subjects like broadcast channels [6], random access channels [7], etc.

Manuscript received 1 June 2023; revised 2 January 2024; accepted 10 March 2024; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor S. Kompella. This work was supported in part by NSF under Grant CCF-1618475, Grant CCF-1816013, Grant CCF-2008527, Grant CNS-2107363, and Grant CCF-2309887; and in part by the National Spectrum Consortium (NSC) under Grant W15QKN-15-9-1004. Part of this work was presented in ISIT2022 [DOI: 10.1109/ISIT50566.2022.9834697].

The author is with the Department of ECE, Purdue University, West Lafayette, IN 47907 USA (e-mail: chihw@purdue.edu).

This article has supplementary downloadable material available at https://doi.org/10.1109/TNET.2024.3379895, provided by the author.

Digital Object Identifier 10.1109/TNET.2024.3379895

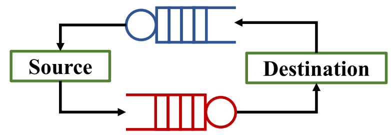


Fig. 1. Information Update System with 2-Way Queues.

One earliest canonical example of AoI minimization is the *update-through-queues* systems [1], [8], [9], [10], [12], [13], [14], [15], [16], [17], [19], [20]. Specifically, a source node s would like to send update packets through a queue to a destination node d. The AoI at d is defined as

$$\Delta(t) \triangleq t - \max\{S_i : \forall i \text{ s.t. } D_i < t\} \tag{1}$$

where S_i is the *send time* of the i-th packet P_i (the time of injecting P_i into the queue) and D_i is the *delivery time* (the time P_i departs the queue). The objective is to design $\{S_i : i\}$ that minimizes the *average AoI* or the *average peak AoI*.

Existing results of this model fall into two categories: The open-loop versus closed-loop settings. In the open-loop settings [8], [9], [16], [17], [19], the sender is oblivious of the packet departure time. Analysis has been conducted for different queue service policies, e.g., Last-Come-First-Serve, and the optimal scheme generally follows a stationary randomized design. In the closed-loop settings [10], [12], [13], [15], [20] s has instantaneous ACK of the departure time D_{i-1}. Optimal $\{S_i : i\}$ are analyzed for arbitrary AoI penalty functions [10], [12], transmission cost [13], [15], and provably optimal distribution-oblivious online algorithms [12], [14].

Nonetheless, modern network protocols almost always rely on feedback that experiences some (random) delay. It remains unclear whether one should employ a closed-loop scheme designed for instantaneous ACK while knowing the feedback being used is actually stale, or one should take an open-loop approach that discards the delayed feedback completely. Intuitively, even though delayed feedback is not as valuable as instantaneous ACK, it still contains some information that can assist scheduling. The question to answer, though, is how to design *schemes that extract the information from the delayed feedback and perform optimal scheduling accordingly.*

With this motivation, this work subjects the ACK traffic to a second queue so that the closed-loop decision is based on delayed ACK. See Fig. 1. The main contributions are:

(i) For any integer $K \geq 0$, we propose new ways to design an order-K achievability scheme and an order-K *genie-aided* converse result, which satisfies that the larger the K value, the smaller the performance gap between the two, and the gap is zero if $K = \infty$. Numeric evaluation shows that for $K = 1$, the gap between the achievability and converse is often $< 2\%$, and the gap reduces to $< 0.2\%$ if $K = 2$. Our achievability and converse results thus effectively determine the optimal average AoI with delayed feedback.

(ii) By characterizing the optimal AoI under delayed feedback, the results reveal a smooth transition between the closed-loop and open-loop schemes, a critical piece of information for system designers. E.g., numerical evaluation shows that if the forward and feedback queues have comparable service time, then the benefits of delayed feedback vanish almost completely, and we could use the open-loop approach to achieve near-optimal performance. On the other hand, if the feedback delay is half of the forward delay, then significant gain can still be achieved when using a closed-loop design.

The rest of the paper is organized as follows. Sec. II provides the problem formulation. Sec. III defines key quantities that will be used when describing our main results. Secs. IV and V describe the *order-1* genie-based converse bound and achievability scheme, respectively. Sec. VI describes the order-2 converse and achievability. Sec. VII presents the numerical evaluation. Sec. VIII provides the intuition and several important remarks. Sec. IX concludes this work. The proofs are relegated to the appendices of [21].

II. Problem Formulation

We assume *slotted time axis*, i.e., the injection and departure times of both queues in Fig. 1 are integers. At time 0, both queues are empty. For any packet index $i \geq 1$, source s would inject packet P_i to the forward queue at the *send time* S_i. P_i will leave the forward queue and arrive at destination d at the *delivery time* D_i. Once delivered, the ACK packet of P_i, denoted by Ack_i, is immediately injected to the backward queue (thus at time D_i). Ack_i will leave the backward queue at the *ACK time* A_i. Once it returns back to s, Ack_i will inform s the exact delivery time D_i of P_i.

For each packet P_i (and its corresponding Ack_i) we denote the i.i.d. service times of the forward and backward queues by $Y_i \sim \mathbb{P}_Y$ and $Z_i \sim \mathbb{P}_Z$, respectively. $\mathbb{P}_Y$ and $\mathbb{P}_Z$ can be arbitrary distributions with bounded supports $[1, y_{\max}]$ and $[0, z_{\max}]$, respectively. The assumption of $Y_i \geq 1$ is to avoid the complication of *instantaneous forward delivery*. We still allow for $Z_i = 0$ so that we can choose $\mathbb{P}_Z$ to include "instantaneous ACK" [10] as a special case. We initialize $S_i = D_i = A_i = 0$ for all $i \leq 0$. Under the basic FIFO-queue model, the relationships between S_i, D_i, A_i, Y_i and Z_i for all $i \geq 1$ are iteratively defined by

$$D_i = \max(S_i, D_{i-1}) + Y_i; \tag{2}$$
$$A_i = \max(D_i, A_{i-1}) + Z_i. \tag{3}$$

E.g., packet P_i will be processed at time $\max(S_i, D_{i-1})$. Then it takes Y_i additional time for P_i to be delivered to d. We also define the projection operator: $(\cdot)^+ \triangleq \max(\cdot, 0)$.

For any $i \geq 1$, define a random process $\mathsf{ack.del}_i(t) \triangleq D_i \cdot 1_{\{A_i \leq t\}}$, which jumps from 0 to D_i at ACK time A_i and stays at D_i afterward, i.e., $\mathsf{ack.del}_i(t)$ is the *acknowledged-delivery-time until time* t. Define $\mathbb{F}^{(i)} \triangleq \{\mathcal{F}_t^{(i)} : t \in [1, \infty)\}$ as the filtration generated by random processes $\{\mathsf{ack.del}_j(t) : j \in [1, i-1]\}$. I.e., σ-algebra $\mathcal{F}_t^{(i)}$ contains all the information available to s when making the S_i decision at time t.

This work studies the following AoI minimization problem:

$$\mathsf{avg.aoi}^* \triangleq \inf_{\{S_i : i \geq 1\}} \lim_{T \to \infty} \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}\{\Delta(t)\} \tag{4}$$
$$\text{subject to} \quad \forall i \in [1, \infty),\ S_{i-1} < S_i \text{ and} \tag{5}$$
$$S_i \text{ is a } \textit{stopping time} \text{ w.r.t. } \mathbb{F}^{(i)} \tag{6}$$

where $\Delta(t)$ is defined in (1); and (5) ensures that P_{i-1} is, by definition, sent at an earlier time than P_i.

Our model is general. For example, we can choose $\mathbb{P}_Z$ to be instantaneous ACK $\mathbb{P}(Z_i = 0) = 1$ [10], to be deterministic but non-zero, to be (truncated) log-normal distribution, or to be $\mathbb{P}(Z_i = z_{\max}) = 1$ for a large $z_{\max}$ that mimics the open-loop setting in which feedback never arrives.

A. Four Existing Upper and Lower Bounds of $\mathsf{avg.aoi}^*$

We first describe four existing bounds of $\mathsf{avg.aoi}^*$:

Zero-Wait-After-ACK (ZWAA) is a scheme for which s sends P_i immediately after receiving Ack_{i-1}, i.e., $S_i = A_{i-1}$. We denote the corresponding average AoI by zwaa. By definition, $\mathsf{zwaa} \geq \mathsf{avg.aoi}^*$ and simple computation shows

$$\mathsf{zwaa} = \mathbb{E}(Y) + 0.5 + \frac{\mathbb{E}(Y^2) + 2\mathbb{E}(Y)\mathbb{E}(Z) + \mathbb{E}(Z^2)}{2 \cdot (\mathbb{E}(Y) + \mathbb{E}(Z))}. \tag{7}$$

Best-After-ACK (BAA) [11], [12], [13], [15], [22] adds a constraint $S_i \geq A_{i-1}$ to (4)–(6) and solves the optimal value of the restricted problem, i.e., new packet P_i can be sent only after receiving Ack_{i-1}. By definition, the AoI achieved by this scheme, denoted by baa, satisfies $\mathsf{avg.aoi}^* \leq \mathsf{baa} \leq \mathsf{zwaa}$.

Optimal periodic (Opt.Per) is an open-loop scheme which schedules $S_i = \lfloor (i-1) \cdot c \rfloor$ where $c > 0$ is a real-valued period being used. Namely, source s sends out a new packet roughly every c time slots, while completely ignoring any feedback information. We can run Monte-Carlo simulation for each different c and then choose the (numerically found) optimal c^* that leads to the smallest $\mathsf{avg.aoi}$. The result is an upper bound of $\mathsf{avg.aoi}^*$, which we denote by $\mathsf{opt.per}$.

Instantaneous ACK (Inst.ACK) hardwires $\mathbb{P}(Z_i = 0) = 1$ and uses [10] to compute the optimal AoI value. Since the new instantaneous feedback setting dominates the delayed feedback setting in a path-wise sense, the result is a lower bound of $\mathsf{avg.aoi}^*$, which we denote by $\mathsf{inst.ack}$.

As shown in Sec. VII, none of zwaa, baa, $\mathsf{opt.per}$, and $\mathsf{inst.ack}$ is tight (i.e., close to $\mathsf{avg.aoi}^*$) in general. Further comparison to existing results will be provided in Sec. VIII-A.

B. An Alternative Way of Counting The Average AoI

For any integer $T \geq 0$, define $i(T) \triangleq \max\{i : D_i < T\}$ and for any non-negative integer (δ, y) pair, define

$$\gamma(\delta, y) \triangleq \left(\sum_{k=1}^{\delta+y} k\right) - \left(\sum_{k=1}^{y} k\right) = \frac{\delta^2}{2} + \delta \cdot (y + 0.5). \tag{8}$$

We now introduce a useful lemma.

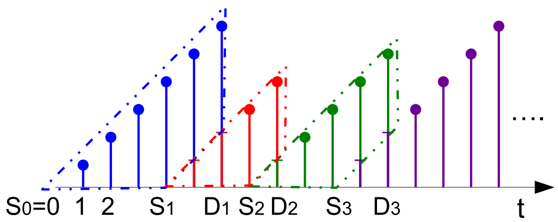


Fig. 2. An alternative way of computing $\sum_{t=1}^{T} \Delta(t)$.

Lemma 1: For any $T > 0$, we have

$$\sum_{i=1}^{i(T)} \gamma(S_i - S_{i-1}, D_i - S_i) \leq \sum_{t=1}^{T} \Delta(t) \leq \sum_{i=1}^{i(T)+1} \gamma(S_i - S_{i-1}, D_i - S_i). \quad (9)$$

Proof: We observe that we can sum the AoI in a different way as illustrated in Fig. 2, where the expression $\gamma(\delta, y)$ computes the area of each trapezium. For example, the green trapezoidal area is computed by $\gamma(S_3 - S_2, D_3 - S_3)$. Note that this alternative counting method is a well-known technique in the literature [10]. □

The intuition of $\gamma(\delta, y)$ is as follows: δ is the spacing between two consecutive *send times* $S_i - S_{i-1}$; y is how much time it takes for P_i to arrive at the destination. Jointly, $\gamma(\delta, y)$ describes the *additional* AoI cost of sending P_i (without double counting the AoI cost of sending the previous P_{i-1}.)

By Lemma 1, we can rewrite the objective function (4) by

$$\mathsf{avg.aoi}^* \triangleq \inf_{\{S_i : i \geq 1\}} \lim_{I \to \infty} \mathbb{E}\left\{ \frac{\sum_{i=1}^{I} \gamma(S_i - S_{i-1}, D_i - S_i)}{\sum_{i=1}^{I} S_i - S_{i-1}} \right\}. \quad (10)$$

The new problem (10), (5), and (6) now closely resembles the classical average cost per stage (ACPS) problem.

III. Definitions of Several Important Quantities

Since it is convenient to put all similar definitions in a central location, we now introduce several important definitions used extensively in the rest of this work. Readers may opt for skipping this section and only come back when encountering these definitions in the subsequent sections.

A. Term a_1 & The Relative Time Index With Respect To S_{i-1}

Fix any deterministic i value. We define an integer $\mathsf{a}_1 \in \mathbb{N}^+$:

$$\mathsf{a}_1 \triangleq A_{i-1} - S_{i-1} \geq 0 \quad (11)$$

where $\mathbb{N}^+$ is the set of all non-negative integers. The intuition of a_1 is as follows. The decision of the send time S_i of the current packet P_i can only be made *after* P_{i-1} has been sent, see (5). Therefore, *we can view the time index S_{i-1} as a new time origin when making the decision*. I.e., source s only needs to decide the *relative send time* with respect to the new time origin S_{i-1}. As will be seen, all our definitions are based on the *relative* time indices with respect to S_{i-1}. Herein, a_1 represents the relative time of A_{i-1} with respect to S_{i-1}.

B. Terms f_1, a_2, $\tilde{\mathsf{A}}_{i-1}$, and $m^+_{Y,1}(x)$

Similarly, we define

$$\mathsf{f}_1 \triangleq \max(D_{i-2}, S_{i-1}) - S_{i-1} = (D_{i-2} - S_{i-1})^+; \quad (12)$$

$$\mathsf{a}_2 \triangleq \max(A_{i-2} - S_{i-1}, \mathsf{f}_1) = (A_{i-2} - S_{i-1})^+. \quad (13)$$

By (12)–(13), we always have $0 \leq \mathsf{f}_1 \leq \mathsf{a}_2$. The physical meanings of $(\mathsf{f}_1, \mathsf{a}_2)$ are as follows. The term $\max(D_{i-2}, S_{i-1})$ in (12) is the instant when packet P_{i-1} starts to be processed by the forward queue. Minus the S_{i-1} value converts it to the *relative* time index versus S_{i-1}, similar to the definition of a_1.

The term $(A_{i-2} - S_{i-1})$ in (13) is the relative time index when the backward queue has finished servicing Ack_{i-2}. Since a backward queue can start processing the next packet Ack_{i-1} only if Ack_{i-2} has left the queue *and* only after the forward packet P_{i-1} has started to be processed by the forward queue, the max operator in (13) depicts the *relative time index when the backward queue can possibly start processing* Ack_{i-1}.

Perhaps the best way to illustrate $(\mathsf{f}_1, \mathsf{a}_2)$ is to introduce a related definition. For any given $(\mathsf{f}_1, \mathsf{a}_2)$ values, we define

$$\tilde{\mathsf{A}}_{i-1} \triangleq \max(\mathsf{f}_1 + Y_{i-1}, \mathsf{a}_2) + Z_{i-1}. \quad (14)$$

Following the intuition of (12)–(13), the $\mathsf{f}_1 + Y_{i-1}$ term in (14) represents when P_{i-1} will be received by d; the term $\max(\mathsf{f}_1 + Y_{i-1}, \mathsf{a}_2)$ then represents when the feedback Ack_{i-1} will start to be processed by the backward queue; and $\tilde{\mathsf{A}}_{i-1}$ thus represents when Ack_{i-1} will return back to s, but described in a relative time scale versus S_{i-1}.

We now introduce another definition that will be used extensively later. However, because its meaning is mostly related to the actual scheme construction, we will provide its intuition in Sec. IV instead. For any given deterministic values of $(\mathsf{f}_1, \mathsf{a}_2)$, define a function of $x \in \mathbb{N}^+$ as follows.

$$m^+_{Y,1}(x) \triangleq \mathbb{E}\{Y_i\} + \mathbb{E}\left\{ ((\mathsf{f}_1 + Y_{i-1}) - (\mathsf{a}_2 + x))^+ \,\middle|\, \tilde{\mathsf{A}}_{i-1} > \mathsf{a}_2 + x \right\}. \quad (15)$$

Note that when evaluating (15), the only randomness is the three independent random variables Y_i, Y_{i-1}, and Z_{i-1} since $(\mathsf{f}_1, \mathsf{a}_2)$ are assumed to be deterministic parameters and $\tilde{\mathsf{A}}_{i-1}$ in (14) involves only (Y_{i-1}, Z_{i-1}). Clearly, the distribution of $\tilde{\mathsf{A}}_{i-1}$ depends on the deterministic parameters $(\mathsf{f}_1, \mathsf{a}_2)$, and so does the function $m^+_{Y,1}(x)$. For notational simplicity, we opt for not putting $(\mathsf{f}_1, \mathsf{a}_2)$ in either the subscript or the superscript.

For example, if we have $\mathsf{f}_1 = \mathsf{a}_2 = 7$, then the following two events are equivalent:

$$\left\{ \tilde{\mathsf{A}}_{i-1} > \mathsf{a}_2 + x \right\} = \{ Y_{i-1} + Z_{i-1} > x \} \quad (16)$$

and (15) becomes

$$m^+_{Y,1}(x) = \mathbb{E}\{Y_i\} + \mathbb{E}\left\{ (Y_{i-1} - x)^+ \,\middle|\, Y_{i-1} + Z_{i-1} > x \right\} \quad (17)$$

which can be easily evaluated using the probability distributions $\mathbb{P}_Y$ and $\mathbb{P}_Z$. Note that the function $m^+_{Y,1}(x)$ is always of the expression of (17) whenever $\mathsf{f}_1 = \mathsf{a}_2$. However, for general $0 \leq \mathsf{f}_1 < \mathsf{a}_2$, the function $m^+_{Y,1}(x)$ will assume a different expression that needs to be re-derived from its original definition in (15).

C. Terms f_2, a_3, $\tilde{\mathsf{A}}_{i-2}$, and $m^+_{Y,2}(x)$

Define

$$\mathsf{f}_2 \triangleq \max(D_{i-3}, S_{i-2}) - S_{i-1}; \tag{18}$$

$$\mathsf{a}_3 \triangleq \max(A_{i-3} - S_{i-1}, \mathsf{f}_2) \tag{19}$$

$$= \max(A_{i-3}, S_{i-2}) - S_{i-1} \tag{20}$$

where (20) is by substituting the f_2 term in (19) by its definition in (18), and by noting $D_{i-3} \leq A_{i-3}$. By (19), we always have $\mathsf{f}_2 \leq \mathsf{a}_3$. The physical meanings of $(\mathsf{f}_2, \mathsf{a}_3)$ are as follows. The term $\max(D_{i-3}, S_{i-2})$ in (18) is the instant when packet P_{i-2} starts to be processed by the forward queue. Minus S_{i-1} converts it to the *relative* time index versus S_{i-1}.

The term $(A_{i-3} - S_{i-1})$ in (19) is *the relative time index when the backward queue has finished servicing* Ack_{i-3}. Since a backward queue can start processing Ack_{i-2} only if Ack_{i-3} has left the queue *and* only after P_{i-2} has started to be processed by the forward queue, the maximum operator in (19) depicts the *relative time index when the backward queue can possibly start processing the feedback packet* Ack_{i-2}.

For any given deterministic $(\mathsf{f}_2, \mathsf{a}_3)$ values, we define

$$\tilde{\mathsf{A}}_{i-2} \triangleq \max(\mathsf{f}_2 + Y_{i-2}, \mathsf{a}_3) + Z_{i-2}. \tag{21}$$

Following the same reasoning as in the discussion of $\tilde{\mathsf{A}}_{i-1}$ in (14), $\tilde{\mathsf{A}}_{i-2}$ represents when Ack_{i-2} will return back to s, under a *relative* time scale versus the new time origin S_{i-1}.

We now provide the last definition while relegating the discussion of its intuition to Sec. VI. For any given deterministic values of $(\mathsf{f}_2, \mathsf{a}_3)$, define a function of $x \in \mathbb{N}^+$ as follows.

$$m^+_{Y,2}(x) \triangleq \mathbb{E}\{Y_i\} + \mathbb{E}\left\{ \left((\mathsf{f}_2 + Y_{i-2})^+ + Y_{i-1} - (\mathsf{a}_3^+ + x)\right)^+ \middle| \tilde{\mathsf{A}}_{i-2} > \mathsf{a}_3^+ + x \right\}. \tag{22}$$

Note that when evaluating (22), the only randomness is the four independent random variables Y_i, Y_{i-1}, Y_{i-2} and Z_{i-2} since $(\mathsf{f}_2, \mathsf{a}_3)$ are assumed to be deterministic parameters and $\tilde{\mathsf{A}}_{i-2}$ in (21) involves only (Y_{i-2}, Z_{i-2}). Clearly, the distribution of $\tilde{\mathsf{A}}_{i-2}$ depends on $(\mathsf{f}_2, \mathsf{a}_3)$, and so does the function $m^+_{Y,2}(x)$. For notational simplicity, we opt for not putting $(\mathsf{f}_2, \mathsf{a}_3)$ in either the subscript or the superscript.

IV. Main Result #1: A New Class of Lower Bounds

For any $K \geq 0$, we derive an order-K converse (lower bound) by analyzing the following genie-aided scheme. Specifically, for any packet index $i \geq 1$, at time $\max(S_{i-1}, D_{i-K-1})$, a genie will temporarily take over the backward queue and deliver all packets in the following set

$$\{\mathsf{Ack}_j : j \leq i - K - 1\} \tag{23}$$

to source s *instantaneously*. Those Ack packets will be immediately removed from the backward queue and will no longer "block" the service of any newer Ack packets.

A few remarks are in order. Firstly, in our model, both the forward and backward FIFO queues are beyond the control of the source, the same setting as in [3], [4], [5], [10], [11], [12], and [15]. However, when deriving an *impossibility result*, we utilize a genie who is not bound by this constraint and can directly manipulate the backward queue (but not the forward queue).

Secondly, when $K = 0$ we have $\max(S_{i-1}, D_{i-K-1}) = D_{i-1}$. Therefore, the $K = 0$ genie will take over the backward queue whenever P_{i-1} was delivered. Since Ack_{i-1} is injected to the backward queue at time D_{i-1}, the genie will immediately deliver Ack_{i-1} back to s at time D_{i-1}, see (23). The order-0 genie essentially eliminates the backward queueing delay, and the order-0 converse bound is thus equivalent to the inst.ack bound in Sec. II-A.

Thirdly, suppose $K \geq 1$. Note that Ack_{i-K-1} was injected to the backward queue at time D_{i-K-1}. If $S_{i-1} \gg D_{i-K-1}$, then when the genie takes over at time $\max(S_{i-1}, D_{i-K-1}) = S_{i-1}$, the packet Ack_{i-K-1} could have been delivered back to s by the backward queue already. In this case, the acknowledgment packet set in (23) is empty. There is thus nothing for the genie to "deliver" in this scenario.

We now propose the following scheduling rule:

Rule G1: During time $[S_{i-1}, \max(S_{i-1}, D_{i-K-1}))$, source s waits and must not generate/send the current packet P_i. I.e., this rule imposes $S_i \geq \max(S_{i-1}, D_{i-K-1})$.

We would like to emphasize that even though s is aware of the existence of an order-K genie, it does not have access to genie's information. All s knows is that sometimes there is a batch of Ack packets being delivered instantaneously, likely by a genie but could also be delivered by the backward queue.[1] Therefore, we need to show that s is capable of carrying out Rule G1, given the knowledge available to s.

Lemma 2: With the presence of an order-K genie, source s is capable of carrying out Rule G1.

The proof is relegated to Appendix A-A of [21].

We now prove the following lemma:

Lemma 3: We can assume the optimal order-K genie-aided scheme follows Rule G1 without loss of generality.

Proof: The delivery time of P_{i-1} always satisfies $D_{i-1} \geq \max(S_{i-1}, D_{i-K-1})$. Therefore, any deviation from Rule G1 means that the P_i sent by s would get stuck behind P_{i-1}, which is strictly suboptimal for AoI minimization. □

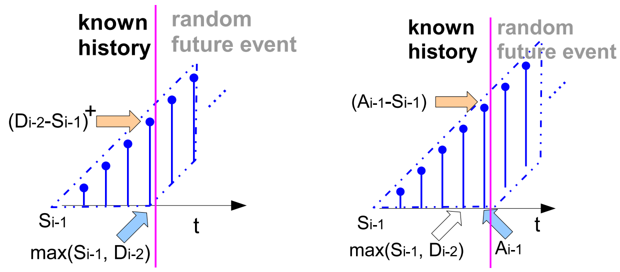
The above lemma shows that Rule G1 is optimal for general order-K genie-aided schemes. The optimal policies for $K = 1$ and 2 are described in Secs. IV-A and VI-A, respectively.

A. The Order-1 Converse

Consider two arbitrarily given *waiting time functions* $\phi^{[1]}_{\mathsf{ini}} : \mathbb{N}^+ \mapsto \mathbb{N}^+$ and $\phi^{[1]}_{\mathsf{a}} : \mathbb{N}^+ \mapsto \mathbb{N}^+$.

Rule G2: At time $t = \max(S_{i-1}, D_{i-2})$, source s computes the values of f_1 in (12) and $x^*_{\mathsf{ini}} \triangleq \phi^{[1]}_{\mathsf{ini}}(\mathsf{f}_1)$. If Ack_{i-1} has not returned by time $\max(S_{i-1}, D_{i-2}) + x^*_{\mathsf{ini}}$, then s will send P_i at that time. Namely, x^*_{ini} is the additional waiting

[1] A genie only facilitates the delivery of the Ack packets. It does not "label" the delivered Ack packets in any way.



(a) $\phi^{[1]}_{\mathsf{ini}}(\cdot)$ (b) $\phi^{[1]}_{\mathsf{a}}(\cdot)$

Fig. 3. Illustrations of $\phi^{[1]}_{\mathsf{ini}}(\cdot)$ and $\phi^{[1]}_{\mathsf{a}}(\cdot)$ of Rules G2 and G3.

time after $\max(S_{i-1}, D_{i-2})$ if Ack_{i-1} has not returned by then. The subscript "ini" stands for "initial decision".

Rule G3: If Ack_{i-1} has returned at an earlier time than $\max(S_{i-1}, D_{i-2})+x^*_{\mathsf{ini}}$, i.e., $A_{i-1} \leq \max(S_{i-1}, D_{i-2})+x^*_{\mathsf{ini}}$, then at time $t = A_{i-1}$, source s computes the values of a_1 in (11) and $x^*_{\mathsf{a}} \triangleq \phi^{[1]}_{\mathsf{a}}(\mathsf{a}_1)$. Source s will send P_i at time $A_{i-1} + x^*_{\mathsf{a}}$. Namely, x^*_{a} is the additional waiting time after A_{i-1} has returned. The subscript "a" stands for "acknowledged".

In sum, we use Rule G2 initially, but would opportunistically switch to Rule G3 at time A_{i-1} if the *precomputed send time* $\max(S_{i-1}, D_{i-2})+x^*_{\mathsf{ini}}$ has not been "committed" by the time Ack_{i-1} returns. We now have the following lemma.

Lemma 4: With the existence of an order-1 genie, we can assume the optimal order-1 genie-aided scheme follows Rules G2 and G3 without loss of generality.

Proof: At time $\max(S_{i-1}, D_{i-2})$, source s is still waiting for Ack_{i-1} because even the forward packet P_{i-1} has not been processed by the forward queue yet, see (2). If Ack_{i-1} had not returned for some time interval, then no additional "variable" is revealed to s during that interval. Therefore, *s can anticipate the situation and pre-compute the decision S_i at time as early as $t = \max(S_{i-1}, D_{i-2})$, assuming Ack_{i-1} returns later than that decision.* See Rule G2.

We now consider the "state" faced by s at time $\max(S_{i-1}, D_{i-2})$. In general, the state of a Markov decision process (MDP) must fully capture (a) the distribution of the randomness it faces, and (b) the cost it faces if certain decision is made in that particular state.

We first consider (a) the distribution of the randomness faced by s. At time $\max(S_{i-1}, D_{i-2})$, s knows with 100% certainty (i) P_{i-1} has just started to be processed and (ii) there is no other packet in either the forward or the backward queue because the order-1 genie has delivered Ack_{i-2}. Therefore, the distribution of the randomness faced by s is always the same at time $\max(S_{i-1}, D_{i-2})$. There is no variation of the distribution that needs to be included in the "state".

We now consider (b) the cost function faced by s. At time $\max(S_{i-1}, D_{i-2})$, the AoI cost in Fig. 3a has grown to $\max(S_{i-1}, D_{i-2}) - S_{i-1} = \mathsf{f}_1$, see the definition in (12). Since that value will affect the AoI cost of the subsequent MDP decisions, we must include f_1 as part of the state. From the above discussion, we impose the waiting time function to be of the form $\phi^{[1]}_{\mathsf{ini}}(\mathsf{f}_1)$ in Rule G2.

We now argue for Rule G3. Suppose that Ack_{i-1} has returned back to s before the tentative decision $\max(S_{i-1}, D_{i-2})+x^*_{\mathsf{ini}}$. Then at that time instant A_{i-1}, *source s knows with 100% certainty that both the forward and backward queues are empty,* a new piece of information that is "revealed" to s at that moment. As a result, s switches to a new waiting time decision x^*_{a}. Since the randomness faced by s at time A_{i-1} is always the same, the system state at time A_{i-1} is how much the AoI has grown, which is $A_{i-1} - S_{i-1} = \mathsf{a}_1$. See Fig. 3b. To capture the state faced by s at time A_{i-1}, we impose the waiting time to be of the form $\phi^{[1]}_{\mathsf{a}}(\mathsf{a}_1)$. □

Using Rules G1–G3, our problem (10), (5), and (6) becomes an ACPS problem of semi-MDP [23]. We can then use any ACPS solver to numerically compute the best AoI value among all order-1 genie-aided schemes, which serves as a lower bound of $\mathsf{avg.aoi}^*$ for all (non-genie-aided) solutions.

Specifically, the *value functions* $f^{[1]}_{\mathsf{a}}(\mathsf{a}_1)$ and $f^{[1]}_{\mathsf{ini}}(\mathsf{f}_1)$ and Bellman equations for Rules G3 and G2 are as follows.

$$\forall \mathsf{a}_1 \in [1, 2y_{\max} + z_{\max}], \text{ we define}$$

$$f^{[1]}_{\mathsf{a}}(\mathsf{a}_1) = \min_{x \in \mathbb{N}^+} \gamma(\mathsf{a}_1 + x, \mathbb{E}\{Y_i\}) - \mathsf{v} \cdot (\mathsf{a}_1 + x) + f^{[1]}_{\mathsf{ini}}(0); \tag{24}$$

$$\forall \mathsf{f}_1 \in [0, y_{\max}], \text{ we hardwire the value } \mathsf{a}_2 = \mathsf{f}_1 \text{ and define}$$

$$f^{[1]}_{\mathsf{ini}}(\mathsf{f}_1) = \min_{x \in \mathbb{N}^+} \left\{ \sum_{k=1}^{x} \mathbb{P}(\tilde{\mathsf{A}}_{i-1} = \mathsf{f}_1 + k) \cdot f^{[1]}_{\mathsf{a}}(\mathsf{f}_1 + k) \right. \tag{25}$$

$$+ \mathbb{P}(\tilde{\mathsf{A}}_{i-1} > \mathsf{f}_1 + x) \cdot \Bigg(\gamma(\mathsf{f}_1 + x, m^+_{Y,1}(x)) - \mathsf{v} \cdot (\mathsf{f}_1 + x) \tag{26}$$

$$\left. + \sum_{y=1}^{y_{\max}} \mathbb{P}(Y_{i-1} = y | \tilde{\mathsf{A}}_{i-1} > \mathsf{f}_1 + x) \cdot f^{[1]}_{\mathsf{ini}}\left((y - x)^+\right) \Bigg) \right\} \tag{27}$$

where the probabilities involving $\tilde{\mathsf{A}}_{i-1}$ can be computed by (16); the functions $\gamma(\cdot, \cdot)$ and $m^+_{Y,1}(\cdot)$ are defined in (8) and (17), respectively; and v is a scalar variable that represents the *average cost*. For example, if $\mathsf{f}_1 = 7$, then the Bellman equation (25)–(27) (after hardwiring $\mathsf{a}_2 = \mathsf{f}_1 = 7$) becomes

$$f^{[1]}_{\mathsf{ini}}(7) = \min_{x \in \mathbb{N}^+} \left\{ \sum_{k=1}^{x} \mathbb{P}(Y_{i-1} + Z_{i-1} = k) \cdot f^{[1]}_{\mathsf{a}}(7 + k) \right.$$

$$+ \mathbb{P}(Y_{i-1} + Z_{i-1} > x) \cdot \Bigg(\gamma(7 + x, m^+_{Y,1}(x)) - \mathsf{v} \cdot (7 + x)$$

$$\left. + \sum_{y=1}^{y_{\max}} \mathbb{P}(Y_{i-1} = y | Y_{i-1} + Z_{i-1} > x) \cdot f^{[1]}_{\mathsf{ini}}\left((y - x)^+\right) \Bigg) \right\}$$

which can be easily evaluated using the given distributions $\mathbb{P}_Y$ and $\mathbb{P}_Z$, and the expression of $m^+_{Y,1}(x)$ in (17).

The reason that we hardwire $\mathsf{a}_2 = \mathsf{f}_1$ is two-fold. Firstly, the distribution of $\tilde{\mathsf{A}}_{i-1}$ in (14) and the function $m^+_{Y,1}(x)$ in (15) are defined only after the deterministic values $(\mathsf{f}_1, \mathsf{a}_2)$ are given. Therefore, we need to explicitly specify the a_2 value being used when stating the Bellman equation (25)–(27). Secondly, we have the following lemma.

Lemma 5: With the presence of an order-1 genie, the $(\mathsf{f}_1, \mathsf{a}_2)$ computed by (12) and (13) always satisfy $\mathsf{f}_1 = \mathsf{a}_2$. A short proof is relegated to Appendix A-B of [21].

We now describe how we derive the Bellman equations. Recall that $\gamma(\delta, y)$ in (8) is the AoI cost of the *total waiting time* being $S_i - S_{i-1} = \delta$ and the end-to-end delay between sending and receiving P_i being $y = D_i - S_i$. Also see Fig. 2. Therefore, the cost of a decision S_i is simply

$$\gamma(S_i - S_{i-1}, D_i - S_i). \tag{28}$$

Also see (9) in Lemma 1 and the discussion therein.

Recall that $f_{\mathsf{a}}^{[1]}(\mathsf{a}_1)$ in (24) is the value function after receiving Ack_{i-1} at time $t = A_{i-1}$. In this case, any additional waiting time x will result in the total waiting time being

$$S_i - S_{i-1} = (A_{i-1} + x) - S_{i-1} = \mathsf{a}_1 + x \tag{29}$$

and the end-to-end delay of packet P_i being $D_i - S_i = Y_i$ since the forward queue is empty at time A_{i-1}. Let $\mathsf{evnt}_{\mathrm{G3}}$ denote the event that $S_i = A_{i-1} + x$, i.e., Rule G3 decides to wait x additional time slots. Under $\mathsf{evnt}_{\mathrm{G3}}$, we thus have

$$\gamma(S_i - S_{i-1}, D_i - S_i) = \gamma(\mathsf{a}_1 + x, Y_i). \tag{30}$$

Note that the actual value of Y_i is still unknown at time S_i. Therefore, we further take the conditional expectation under $\mathsf{evnt}_{\mathrm{G3}}$. By (30) and because $\gamma(\delta, y)$ in (8) is linear with respect to y, the expected cost becomes

$$\mathbb{E}\left\{\gamma(S_i - S_{i-1}, D_i - S_i)|\mathsf{evnt}_{\mathrm{G3}}\right\} = \gamma(\mathsf{a}_1 + x, \mathbb{E}\{Y_i\})$$

This leads to the first half of the expression of $f_{\mathsf{a}}^{[1]}(\mathsf{a}_1)$ in (24).

The term "$-\mathsf{v} \cdot (\mathsf{a}_1 + x)$" in (24) is a generalization of the average-cost adjustment term of ACPS-MDP to its counterpart for *ACPS-semi-MDP*. Namely, being a semi-MDP, the cost-per-stage is now linearly proportional to the total waiting time $S_i - S_{i-1} = \mathsf{a}_1 + x$. Therefore, we multiply the average-cost variable v with the duration of the semi-MDP decision $\mathsf{a}_1 + x$.

Finally, after sending P_i, source s will move on to the next packet index $i_{\mathrm{nx}} = i + 1$ and decide the next send time $S_{i_{\mathrm{nx}}}$ at time $\max(S_{i_{\mathrm{nx}}-1}, D_{i_{\mathrm{nx}}-2})$. Since $S_i = A_{i-1} + x \geq D_{i-1}$, at time $\max(S_{i_{\mathrm{nx}}-1}, D_{i_{\mathrm{nx}}-2}) = S_i$, source s will face a new $\mathsf{f}_1^{[\mathrm{new}]} = (D_{i_{\mathrm{nx}}-2} - S_{i_{\mathrm{nx}}-1})^+ = (D_{i-1} - S_i)^+ = 0$ according to (12). That is why in (24) the *next state value* is always $f_{\mathsf{ini}}^{[1]}(0)$. Overall, the argmin x^* value in (24) is the optimal waiting time function $\phi_{\mathsf{a}}^{[1]}(\mathsf{a}_1)$ for Rule G3.

Now consider Rule G2 and its value function $f_{\mathsf{ini}}^{[1]}(\cdot)$. Suppose at time $\max(S_{i-1}, D_{i-2})$, source s decides to wait for x extra time slots and sends P_i at time $\max(S_{i-1}, D_{i-2}) + x = S_{i-1} + \mathsf{f}_1 + x$, see the definition of f_1 in (12). We first consider the possibility that Ack_{i-1} returns back to s before Rule G2 "commits" its decision of sending P_i at time $S_{i-1} + \mathsf{f}_1 + x$. Recall that $\tilde{\mathsf{A}}_{i-1}$ in (14) is the time when Ack_{i-1} returns back to s (under a relative time scale with respect to S_{i-1}). As a result, if $\{\tilde{\mathsf{A}}_{i-1} = \mathsf{f}_1 + k\}$ for some $k \leq x$, then s will move on to Rule G3 without committing to the Rule-G2 decision.

Therefore, with probability $\mathbb{P}(\tilde{\mathsf{A}}_{i-1} = \mathsf{f}_1 + k)$, the state will transition to Rule G3 with $\mathsf{a}_1 = A_{i-1} - S_{i-1} = \mathsf{f}_1 + k$ without committing to the decision of Rule G2. This gives us the first term of $f_{\mathsf{ini}}^{[1]}(\cdot)$ as described in (25), which represents the probabilistic transition to its next state $f_{\mathsf{a}}^{[1]}(\mathsf{f}_1 + k)$ without incurring any direct cost.

Eqs. (26) and (27) consider the random event that s sends P_i at time $S_{i-1} + \mathsf{f}_1 + x$ prior to the return of Ack_{i-1}. That is why we multiply $\mathbb{P}(\tilde{\mathsf{A}}_{i-1} > \mathsf{f}_1 + x)$ in both (26) and (27).

Specifically, (26) quantifies the direct cost incurred by committing this decision. For further explanation, we define

$$\mathsf{evnt}_{\mathrm{G2}.x} \triangleq \{A_{i-1} > S_i = \max(S_{i-1}, D_{i-2}) + x\} \tag{31}$$

as the event that the Rule-G2 decision is committed prior to the return of Ack_{i-1}. We now consider the conditional expectation of the cost (28) given $\mathsf{evnt}_{\mathrm{G2}.x}$. The total waiting time is

$$S_i - S_{i-1} = (\max(S_{i-1}, D_{i-2}) + x) - S_{i-1} = \mathsf{f}_1 + x. \tag{32}$$

The end-to-end delay under $\mathsf{evnt}_{\mathrm{G2}.x}$ is

$$\begin{aligned} D_i - S_i &= (\max(S_i, D_{i-1}) + Y_i) - S_i && (33)\\ &= (D_{i-1} - S_i)^+ + Y_i && (34)\\ &= \big((\max(S_{i-1}, D_{i-2}) + Y_{i-1}) \\ &\quad - (\max(S_{i-1}, D_{i-2}) + x)\big)^+ + Y_i && (35)\\ &= (Y_{i-1} - x)^+ + Y_i && (36)\end{aligned}$$

where (33) is by substituting D_i by (2); (34) is by basic simplification; (35) is by substituting D_{i-1} by (2) and because Rule G2 chooses $S_i = \max(S_{i-1}, D_{i-2}) + x$; and (36) is by basic simplification. The total expected cost thus becomes

$$\begin{aligned} &\mathbb{E}\{\gamma(S_i - S_{i-1}, D_i - S_i)|\mathsf{evnt}_{\mathrm{G2}.x}\} \\ &= \gamma(\mathsf{f}_1 + x, \mathbb{E}\{D_i - S_i|\mathsf{evnt}_{\mathrm{G2}.x}\}) && (37)\\ &= \gamma(\mathsf{f}_1 + x, m_{Y,1}^+(x)) && (38)\end{aligned}$$

where (37) is by (32) and the linearity of $\gamma(\delta, y)$ w.r.t. y; (38) is by (36) and the definitions in (16)–(17) since we hardwire $\mathsf{a}_2 = \mathsf{f}_1$ when stating (25)–(27).

Comparing (38) to the cost term in (24), the difference is that *before the return of* Ack_{i-1}*, when sending* P_i *at time* $\max(S_{i-1}, D_{i-2}) + x$*, source* s *cannot be 100% certain that the forward queue is empty. There is a chance that* P_{i-1} *may "block"* P_i *and the expected delay of* P_i *is thus enlarged from* $\mathbb{E}\{Y_i\}$ *to* $m_{Y,1}^+(x)$ *defined in (17).* Therefore we use $m_{Y,1}^+(x)$ inside the AoI cost term $\gamma(\cdot)$ of (38).

The term "$-\mathsf{v} \cdot (\mathsf{f}_1 + x)$" in (26) is again the average-cost adjustment term for *ACPS-semi-MDP*.

Finally, (27) computes the next state values. Specifically, when s moves on to the next index $i_{\mathrm{nx}} = i + 1$ and decides the send time $S_{i_{\mathrm{nx}}}$ at time $\max(S_{i_{\mathrm{nx}}-1}, D_{i_{\mathrm{nx}}-2})$, the new state value $\mathsf{f}_1^{[\mathrm{new}]}$, defined in (12), at that time becomes

$$\mathsf{f}_1^{[\mathrm{new}]} = (D_{i_{\mathrm{nx}}-2} - S_{i_{\mathrm{nx}}-1})^+ = (Y_{i-1} - x)^+ \tag{39}$$

where (39) follows from the identical steps of deriving the equality in (34)–(36). In the end, (39) shows that the next state value is a function of Y_{i-1}. Multiplying $f_{\mathsf{ini}}^{[1]}((y - x)^+)$ by its probability $\mathbb{P}(Y_{i-1} = y|\tilde{\mathsf{A}}_{i-1} > \mathsf{f}_1 + x)$ gives us the term in (27). Once again, the argmin x^* value in (25)–(27) gives us the optimal waiting time function $\phi_{\mathsf{ini}}^{[1]}(\mathsf{f}_1)$ of Rule G2.

We use *value iteration* to find a scalar v and functions $f^{[1]}_{\mathsf{a}}(\mathsf{a}_1)$ and $f^{[1]}_{\mathsf{ini}}(\mathsf{f}_1)$ that satisfy (24)–(27) with the ground state value being $f^{[1]}_{\mathsf{ini}}(0) = 0$. The final v value is the optimal AoI of the genie-aided scheme, thus a new lower bound $\mathsf{lb}^{[1]}_{\text{new}}$.

B. Remark on The Computation

When solving the Bellman equations, it is critical to ensure the problem is finite. To that end, we note that $\mathsf{a}_1 = A_{i-1} - S_{i-1}$ is upper bounded by $2y_{\max} + z_{\max}$, since it takes at most $y_{\max}$ slots for P_{i-2} be delivered, another $y_{\max}$ slots for P_{i-1} to be delivered, and another $z_{\max}$ slots for Ack_{i-1} to return to s. That is why[2] in (24) the range of a_1 is $[1, 2y_{\max} + z_{\max}]$.

We now argue that the range of f_1 is $[0, y_{\max}]$. Specifically, we have $S_{i-1} \geq \max(S_{i-2}, D_{i-3})$ by Rule G1. Therefore, at time S_{i-1}, packet P_{i-3} has been delivered to d, and it takes at most $y_{\max}$ additional slots to deliver P_{i-2}. This implies $\mathsf{f}_1 \triangleq (D_{i-2} - S_{i-1})^+ \leq y_{\max}$.

Additionally, $\min_{x\in\mathbb{N}^+}$ in (24) can be solved analytically without trying all x values. The reason is that given any v value (and after hardwiring $f^{[1]}_{\mathsf{ini}}(0) = 0$), Eq. (24) is a second-order polynomial with a positive leading term $0.5x^2$. Therefore, the minimizing x can be found analytically.

The minimization in (25) over $x \in \mathbb{N}^+$ can be simplified as well. Specifically, we observe that if we keep increasing the x value in (25)–(27), the probability terms eventually "stabilize" and do not change anymore once $x > y_{\max} + z_{\max}$. Therefore, the minimization only needs to be over $x \in [0, y_{\max} + z_{\max}]$.

In sum, the Bellman equations (24)–(27) are finite and its solution $\mathsf{lb}^{[1]}_{\text{new}}$ can be numerically found.

V. Main Result #2: A New Class of Upper Bounds

For any $K \geq 0$, we derive an order-K achievability scheme (i.e., an upper bound) by imposing the following constraint:

$$S_i \geq A_{i-K-1}, \quad \forall i \geq 1. \tag{40}$$

in addition to (10), (5), and (6). I.e., s is prohibited to transmit P_i before the return of Ack_{i-K-1}. Such a constraint is represented by the following policy rule:

Rule A1: During time $t \in [S_{i-1}, \max(S_{i-1}, A_{i-K-1}))$, source s waits and must not send the current packet P_i.

If $K = 0$, the new constraint becomes $S_i \geq A_{i-1}$, which is exactly the BAA scheme in Sec. II-A. On the other hand, if $K = 1$, our scheme can send P_i *before* A_{i-1} if desired, but must be *after* A_{i-2}. Secs. V-A and VI-C describe the optimal order-1 and order-2 achievability schemes, respectively.

A. The Order-1 Achievability Scheme

Consider two waiting time functions $\theta^{[1]}_{\mathsf{ini}} : (\mathbb{N}^+)^2 \mapsto \mathbb{N}^+$ and $\theta^{[1]}_{\mathsf{a}} : \mathbb{N}^+ \mapsto \mathbb{N}^+$.

Rule A2: At time $t = \max(S_{i-1}, A_{i-2})$, source s computes $(\mathsf{f}_1, \mathsf{a}_2)$ in (12) and (13), respectively, and computes $x^*_{\mathsf{ini}} \triangleq \theta^{[1]}_{\mathsf{ini}}(\mathsf{f}_1, \mathsf{a}_2)$. If Ack_{i-1} has not returned by time $\max(S_{i-1}, A_{i-2}) + x^*_{\mathsf{ini}}$, then s will send P_i at that time, i.e., x^*_{ini} is the additional waiting time after $\max(S_{i-1}, A_{i-2})$ if Ack_{i-1} has not returned by then.

Rule A3: If Ack_{i-1} has arrived at an earlier time than $\max(S_{i-1}, A_{i-2}) + x^*_{\mathsf{ini}}$, then at time $t = A_{i-1}$, s computes a_1 in (11) and $x^*_{\mathsf{a}} \triangleq \theta^{[1]}_{\mathsf{a}}(\mathsf{a}_1)$, and will send P_i at time $A_{i-1} + x^*_{\mathsf{a}}$.

Rules A1 to A3 have the same structure as the genie-aided scheme (Rules G1 to G3) in Sec. IV. The main difference lies in Rules A2 vs G2, for which the state value now consists of a pair $(\mathsf{f}_1, \mathsf{a}_2)$ instead of a scalar f_1. To explain the difference, we note that in Rule G2, the state value is "how much the AoI has grown at the decision time". Since the decision of Rule A2 is made at time $\max(S_{i-1}, A_{i-2})$, we include $\mathsf{a}_2 = \max(S_{i-1}, A_{i-2}) - S_{i-1}$ as part of the state, which serves a similar role as the f_1 in Rule G2.

We now explain why we still need to include f_1 as a state value of Rule A2 when we already have a_2 as part of the state. At time $t = \max(S_{i-1}, A_{i-2})$, using Ack_{i-2}, source s knows with 100% certainty the value of D_{i-2}, the time when P_{i-2} left the forward queue. Therefore, the past a_2 slots (counted from the injection of P_{i-1} to the current time $\max(S_{i-1}, A_{i-2})$) can be divided into two segments: Segment 1: The first $\mathsf{f}_1 = (D_{i-2} - S_{i-1})^+$ slots during which the forward queue was still busy processing P_{i-2} and thus *cannot* process P_{i-1}; and Segment 2: The remaining $\mathsf{a}_2 - \mathsf{f}_1$ slots, during which the forward queue started to process P_{i-1}.

As a result, *the shorter the Segment 1 is (the longer the Segment 2), the more time the forward queue has devoted to serving P_{i-1}, the more likely that P_{i-1} has been delivered to d (though we cannot be 100% sure since there is no return of Ack_{i-1} yet), and the more likely that new packet P_i will face an empty queue and thus a shorter delay.* The value f_1 is thus another critical information when deciding the send time S_i. That is why we include both $(\mathsf{f}_1, \mathsf{a}_2)$ as the state in Rule A2.

Using Rules A1 to A3, we can numerically find the best AoI among all order-1 achievability schemes by solving the corresponding ACPS problem. The computed AoI value then becomes an upper bound of $\mathsf{avg.aoi}^*$. Specifically, the Bellman equations can be written as follows. $\forall \mathsf{a}_1 \in \mathbb{N}^+$ we have

$$g^{[1]}_{\mathsf{a}}(\mathsf{a}_1) = \min_{x\in\mathbb{N}^+} \gamma(\mathsf{a}_1 + x, \mathbb{E}\{Y_i\}) - \mathsf{v}\cdot(\mathsf{a}_1 + x) + g^{[1]}_{\mathsf{ini}}(0, 0) \tag{41}$$

and $\forall(\mathsf{f}_1, \mathsf{a}_2) \in (\mathbb{N}^+)^2$ we have

$$\begin{aligned} & g^{[1]}_{\mathsf{ini}}(\mathsf{f}_1, \mathsf{a}_2) \\ & = \min_{x\in\mathbb{N}^+} \\ & \quad \Bigg\{ \sum_{k=1}^{x} \mathbb{P}\Big(\tilde{\mathsf{A}}_{i-1} = \mathsf{a}_2 + k \Big| \tilde{\mathsf{A}}_{i-1} > \mathsf{a}_2 \Big) \cdot g^{[1]}_{\mathsf{a}}(\mathsf{a}_2 + k) && (42) \\ & \quad + \mathbb{P}\left(\tilde{\mathsf{A}}_{i-1} > \mathsf{a}_2 + x \Big| \tilde{\mathsf{A}}_{i-1} > \mathsf{a}_2 \right) \cdot \\ & \quad \left(\gamma\left(\mathsf{a}_2 + x, m^+_{Y,1}(x)\right) - \mathsf{v}\cdot(\mathsf{a}_2 + x) \right. && (43) \\ & \quad + \sum_{y,z} \mathbb{P}\left(Y_{i-1} = y, Z_{i-1} = z \middle| \tilde{\mathsf{A}}_{i-1} > \mathsf{a}_2 + x \right) \cdot \\ & \quad \left. \left. \bar{g}^{[1]}_{\mathsf{ini}}\left(\mathsf{f}^{[\text{new}]}_1, \mathsf{a}^{[\text{new}]}_2\right) \right) \right\} && (44) \end{aligned}$$

[2]We do not need to worry about the time for Ack_{i-2} to be delivered since that packet will be delivered instantaneously by the order-1 genie.

where $\tilde{\mathsf{A}}_{i-1}$ is defined in (14) and

$$\overline{g}^{[1]}_{\mathsf{ini}}(\mathsf{f},\mathsf{a}) \triangleq \mathbb{P}(\max(\mathsf{f}+Y_i,\mathsf{a})+Z_i=\mathsf{a})\cdot g^{[1]}_{\mathsf{a}}(\mathsf{a}) + \mathbb{P}(\max(\mathsf{f}+Y_i,\mathsf{a})+Z_i>\mathsf{a})\cdot g^{[1]}_{\mathsf{ini}}(\mathsf{f},\mathsf{a}) \quad (45)$$

$$\mathsf{f}_1^{[\text{new}]} \triangleq (\mathsf{f}_1+y-\mathsf{a}_2-x)^+ \quad (46)$$

$$\mathsf{a}_2^{[\text{new}]} \triangleq \left((\mathsf{f}_1+y-\mathsf{a}_2)^+ +z-x\right)^+. \quad (47)$$

We now explain how we derive the Bellman equations. Specifically, (41) describes the Bellman equation under Rule A3, which is almost identical to (24) and consists of the AoI cost term $\gamma(\mathsf{a}_1+x,\mathbb{E}\{Y_i\})$, the ACPS adjustment term $-\mathsf{v}(\mathsf{a}_1+x)$, and the next state value term $g^{[1]}_{\mathsf{ini}}(0,0)$.

The reasoning of the first two terms is verbatim to the discussion of (24). We thus focus our discussion on the last term. In Rule A3, we send P_i *after* A_{i-1}. Therefore, for the next packet index $i_{\text{nx}}=i+1$, the decision time must be $\max(S_{i_{\text{nx}}-1},A_{i_{\text{nx}}-2})=S_i$ since $A_{i-1}\le S_i$. On the other hand, because the forward delay $Y_i\ge 1$ with probability one, we also have $A_{i_{\text{nx}}-1}\ge D_{i_{\text{nx}}-1}>S_{i_{\text{nx}}-1}$. Jointly, it means that when deciding the send time of packet $P_{i_{\text{nx}}}$ at time $S_{i_{\text{nx}}-1}$, the feedback $\mathsf{Ack}_{i_{\text{nx}}-2}$ has returned to s but $\mathsf{Ack}_{i_{\text{nx}}-1}$ has not. Therefore, the scheme must apply Rule A2, which corresponds to the value function $g^{[1]}_{\mathsf{ini}}(\cdot,\cdot)$ term in the end of (41).

Furthermore, the new state values become

$$\mathsf{f}_1^{[\text{new}]} = (D_{i_{\text{nx}}-2}-S_{i_{\text{nx}}-1})^+ = (D_{i-1}-S_i)^+=0 \quad (48)$$

$$\mathsf{a}_2^{[\text{new}]} = (A_{i_{\text{nx}}-2}-S_{i_{\text{nx}}-1})^+ = (A_{i-1}-S_i)^+=0 \quad (49)$$

where (48) and (49) follow from $D_{i-1}\le A_{i-1}\le S_i$ since Bellman equation (41) corresponds to applying Rule A3.

Eqs. (42)–(47) correspond to Rule A2, and they follow the same structure as in (25)–(27). Recall that $\tilde{\mathsf{A}}_{i-1}$ in (14) is a random variable representing when s will receive Ack_{i-1} given the deterministic $(\mathsf{f}_1,\mathsf{a}_2)$ value. Since (42) represents the scenario that we apply Rule A2 at time $\max(S_{i-1},A_{i-2})=S_{i-1}+\mathsf{a}_2$, it implies, though implicitly, that $\tilde{\mathsf{A}}_{i-1}>\mathsf{a}_2$. Otherwise, the scheme would skip Rule A2 and move on to Rule A3 instead. Because of this subtlety of implicitly assuming $\tilde{\mathsf{A}}_{i-1}>\mathsf{a}_2$, the main probability terms in (42) and (43) are both conditional probabilities given $\{\tilde{\mathsf{A}}_{i-1}>\mathsf{a}_2\}$.

After noting that we need to condition on $\{\tilde{\mathsf{A}}_{i-1}>\mathsf{a}_2\}$, the derivation of (42)–(47) is similar[3] to (25)–(27). Specifically, (42) depicts the event that Ack_{i-1} returns back to s before the scheduled send time decision a_2+x. Once it happens, source s will skip Rule A2 and move on to Rule A3, which is represented by the value function $g^{[1]}_{\mathsf{a}}(\mathsf{a}_2+k)$, where a_2+k is the (relative) time index when Ack_{i-1} returns back to s.

Eqs. (43)–(44) describe the event that Ack_{i-1} returns back to s after time a_2+x. In particular, the AoI cost term in (43) uses $m^+_{Y,1}(x)$ function first defined in (15). The reason is similar to the discussion of (26), i.e., P_i is facing *elongated* expected delay (when passing through the forward queue) due to the possibility of being blocked by the earlier packet P_{i-1}. The term "$-\mathsf{v}\cdot(\mathsf{a}_2+x)$" is the adjustment term for the ACPS-semi-MDP problem.

Eq. (44) analyzes the next state when transmitting the next packet $P_{i_{\text{nx}}}$ with $i_{\text{nx}}=i+1$. Specifically, suppose we have $Y_{i-1}=y$ and $Z_{i-1}=z$ under the event $\{\tilde{\mathsf{A}}_{i-1}>\mathsf{a}_2+x\}$. The $\mathsf{f}_1^{[\text{new}]}$ of packet $P_{i_{\text{nx}}}$ is

$$\mathsf{f}_1^{[\text{new}]} = (D_{i_{\text{nx}}-2}-S_{i_{\text{nx}}-1})^+ = ((\mathsf{f}_1+y)-(\mathsf{a}_2+x))^+ \quad (50)$$

where f_1 is the time $P_{i_{\text{nx}}-2}=P_{i-1}$ being processed and $Y_{i-1}=y$ is the random delay for P_{i-1} to reach d; and $(\mathsf{a}_2+x)=S_i$ is the send time of packet $P_{i_{\text{nx}}-1}$. Similarly, the $\mathsf{a}_2^{[\text{new}]}$ of packet $P_{i_{\text{nx}}}$ is

$$\begin{aligned}\mathsf{a}_2^{[\text{new}]} &= (A_{i_{\text{nx}}-2}-S_{i_{\text{nx}}-1})^+ \\ &= ((\max(\mathsf{f}_1+y,\mathsf{a}_2)+z)-(\mathsf{a}_2+x))^+ \end{aligned} \quad (51)$$

where $\max(\mathsf{f}_1+y,\mathsf{a}_2)$ is when $\mathsf{Ack}_{i_{\text{nx}}-2}=\mathsf{Ack}_{i-1}$ is being processed by the backward queue. Adding the backward delay $Z_{i-1}=z$ gives us the $A_{i_{\text{nx}}-2}$ value. The above two equations give us the next state value defined in (46) and (47).

Finally, we describe why we introduce the $\overline{g}^{[1]}_{\mathsf{ini}}(\mathsf{f},\mathsf{a})$ term in (44) and (45). Recall that the next-state-value term in (44) is under the event that $A_{i-1}>S_i=S_{i-1}+\mathsf{a}_2+x$. Therefore, source s will decide the send time of $P_{i_{\text{nx}}}=P_{i+1}$ at time $\max(S_{i_{\text{nx}}-1},A_{i_{\text{nx}}-2})=A_{i_{\text{nx}}-2}$. Namely, the decision of $P_{i_{\text{nx}}}$ is made at time $A_{i_{\text{nx}}-2}$. *However, it is possible that the feedback* $\mathsf{Ack}_{i_{\text{nx}}-1}$ *may return back to* s *at the same time as* $\mathsf{Ack}_{i_{\text{nx}}-2}$ *since the feedback queue could sometimes be instantaneous, i.e.,* $\mathbb{P}(Z_{i_{\text{nx}}-1}=0)>0$. If that happens, the scheduling policy for $P_{i_{\text{nx}}}$ will skip Rule A2 and move on to Rule A3 instead. Therefore, we introduce the $\overline{g}^{[1]}_{\mathsf{ini}}(\mathsf{f},\mathsf{a})$ term in (44) and (45) to properly take into account the probabilistic weights of each event regarding the timing of $\mathsf{Ack}_{i_{\text{nx}}-1}$.

Specifically, the first half of (45) describes the event $\{A_{i_{\text{nx}}-1}=A_{i_{\text{nx}}-2}\}$ or, equivalently, the event

$$\left\{\left(S_{i_{\text{nx}}-1}+\max(\mathsf{f}+Y_i,\mathsf{a})+Z_i\right)=\left(S_{i_{\text{nx}}-1}+\mathsf{a}\right)\right\}$$

where we use (f,a) as shorthand for $(\mathsf{f}_1^{[\text{new}]},\mathsf{a}_2^{[\text{new}]})$. Under this event, s will skip Rule A2 and move on to Rule A3. That is why the first half of (45) is coupled with the value function $g^{[1]}_{\mathsf{a}}(\mathsf{a}_2^{[\text{new}]})$. Similarly, the second half of (45) describes the event $\{A_{i_{\text{nx}}-1}>A_{i_{\text{nx}}-2}\}$, which means that s will use Rule A2 for packet $P_{i_{\text{nx}}}$. That is why the second half of (45) is coupled with the value function $g^{[1]}_{\mathsf{ini}}(\mathsf{f}_1^{[\text{new}]},\mathsf{a}_2^{[\text{new}]})$.

We use value iteration to find a scalar v and functions $g^{[1]}_{\mathsf{a}}(\mathsf{a}_1)$ and $g^{[1]}_{\mathsf{ini}}(\mathsf{f}_1,\mathsf{a}_2)$ that satisfy (41)–(47) and $g^{[1]}_{\mathsf{ini}}(0,0)=0$. The final v value is the AoI cost of the optimal order-1 achievability scheme, which we denote by $\mathsf{ub}^{[1]}_{\mathsf{new}}$. The argmin x^* values in (41)–(44) give the optimal waiting time functions $\theta^{[1]}_{\mathsf{a}}(\mathsf{a}_1)$ and $\theta^{[1]}_{\mathsf{ini}}(\mathsf{f}_1,\mathsf{a}_2)$, respectively. Once the entire functions $\theta^{[1]}_{\mathsf{a}}(\mathsf{a}_1)$ and $\theta^{[1]}_{\mathsf{ini}}(\mathsf{f}_1,\mathsf{a}_2)$ are computed, the scheme can be easily implemented following Rules A1 to A3.

[3] The reason that (25)–(27) does not have to condition on $\{\tilde{\mathsf{A}}_{i-1}>\mathsf{a}_2\}$ is because in (25)–(27) we hardwire $\mathsf{a}_2=\mathsf{f}_1$. Then by (14) we will have $\mathbb{P}(\tilde{\mathsf{A}}_{i-1}>\mathsf{a}_2)=1$ since $\mathbb{P}(Y_{i-1}\ge 1)=1$, and the conditioning event automatically disappears in (25)–(27). However, in the order-1 achievability scheme, we sometimes have $0\le\mathsf{f}_1<\mathsf{a}_2$ and thus $\mathbb{P}(\tilde{\mathsf{A}}_{i-1}>\mathsf{a}_2)<1$. This necessitates the use of the conditional probabilities as in (42)–(47).

B. Remark on The Computation

Thus far, we have described (41)–(44) in their *unbounded form*, i.e., both the input arguments and their minimization ranges $\min_{x\in\mathbb{N}^+}$ are unbounded. To ensure computability/solvability, we further convert it to its bounded form.

We first discuss the search range of $\min_{x\in\mathbb{N}^+}$. Specifically, we note that the $\max_{x\in\mathbb{N}^+}$ in (41) can be solved analytically without trying all $x \in \mathbb{N}^+$ values since (41) is a quadratic polynomial of x with a positive second order term $0.5x^2$.

We then notice that when $x \to \infty$ in (42), the conditional probability eventually becomes $\mathbb{P}(\tilde{\mathsf{A}}_{i-1} > \mathsf{a}_2 + x|\tilde{\mathsf{A}}_{i-1} > \mathsf{a}_2) = 0$, and the value of (42)–(44) will no longer change once $x > (\mathsf{f}_1 + y_{\max} - \mathsf{a}_2)^+ + z_{\max}$. Without loss of generality, we can thus limit the search range of x in (42) to be

$$\mathcal{X}_{\mathsf{ini}}(\mathsf{f}_1, \mathsf{a}_2) = [0, (\mathsf{f}_1 + y_{\max} - \mathsf{a}_2)^+ + z_{\max}]. \tag{52}$$

Finally, we discuss the ranges of the input parameters a_1 in (41) and $(\mathsf{f}_1, \mathsf{a}_2)$ in (42), respectively. Because (41) can be solved analytically, the value of $g^{[1]}_{\mathsf{a}}(\mathsf{a}_1)$ can be computed on the fly for any given a_1 and there is no need to worry about the range of a_1 in numerical computation.

To decide the range of $(\mathsf{f}_1, \mathsf{a}_2)$, we note that we only need to consider $(\mathsf{f}_1, \mathsf{a}_2)$ satisfying $0 \le \mathsf{f}_1 \le \mathsf{a}_2$ because of their definitions in (12) and (13). For any finite or infinite subset

$$\Omega \subseteq \{(\mathsf{f}_1, \mathsf{a}_2) : 0 \le \mathsf{f}_1 \le \mathsf{a}_2 < \infty\} \tag{53}$$

we introduce the following definition.

Definition 1: The set Ω is *self-contained* with respect to the Bellman equations (42)–(47) if it satisfies (i) the ground state $(\mathsf{f}_1, \mathsf{a}_2) = (0, 0) \in \Omega$; and (ii) whenever the $(\mathsf{f}_1, \mathsf{a}_2)$ in the left-hand side of (42) belongs to Ω, then any $g^{[1]}_{\mathsf{ini}}(\mathsf{f}_1^{[\text{new}]}, \mathsf{a}_2^{[\text{new}]})$ involved in the computation of the right-hand side of (42)–(47) must also satisfy $(\mathsf{f}_1^{[\text{new}]}, \mathsf{a}_2^{[\text{new}]}) \in \Omega$.

It is straightforward to see that when solving the Bellman equations, we only need to consider a self-contained Ω since any state $(\mathsf{f}_1, \mathsf{a}_2) \notin \Omega$ is not *reachable* under an optimal policy. I.e., the evolution of the state value is strictly within the self-contained Ω. Using this observation, we can reduce the range of $(\mathsf{f}_1, \mathsf{a}_2)$ to any arbitrarily given finite self-contained Ω:

Lemma 6: The finite set Ω_{sc} that contains all $(\mathsf{f}_1, \mathsf{a}_2)$ satisfying

$$\mathsf{f}_1 \in [0, y_{\max}] \text{ and } (\mathsf{a}_2 - \mathsf{f}_1) \in [0, z_{\max}] \tag{54}$$

is self-contained w.r.t. the Bellman equations (42)–(47).

The proof of Lemma 6 is straightforward by verifying that both conditions (i) and (ii) in Definition 1 hold for Ω_{sc}. A detailed argument is relegated to Appendix A-C of [21].

The above discussion shows that the ACPS-semi-MDP problem (41)–(47) can be made finite without loss of generality. The application of value iteration is thus straightforward.

VI. The Order-2 Converse & Achievability Results

Secs. IV and V discuss the order-1 results. This section focuses on order-2 genie and achievability schemes. Since the derivation is based on similar ideas (being more complicated due to the more involved dynamics of order-2 schemes), we provide complete descriptions and high-level intuitions, and leave detailed discussion in the appendices of [21].

Subsequent sections describe two ACPS problems. Using their corresponding Bellman equations, a user can numerically solve the best AoI value of order-2 genie-aided schemes, which becomes a lower bound of $\mathsf{avg.aoi}^*$; Or a user can numerically solve the best AoI value of order-2 achievability schemes, which becomes an upper bound of $\mathsf{avg.aoi}^*$.

A. The Order-2 Converse

Consider $K = 2$. Lemma 3 says that an optimal scheme must follow Rule G1, i.e., s waits until time $\max(S_{i-1}, D_{i-3})$ and then decides "when to transmit the current packet P_i". In the sequel, we strengthen Rule G1 with new Rules G4 to G6.

Consider three arbitrarily given "waiting time functions" $\phi^{[2]}_{\mathsf{ini}}(\mathsf{f}, \mathsf{a})$, $\phi^{[2]}_{\mathsf{a}}(\mathsf{f}, \mathsf{a})$, and $\phi^{[2]}_{\mathsf{aa}}(\mathsf{a})$, where the input parameters f and a are integers, which can sometimes be negative.

Rule G4: At time $t = \max(S_{i-1}, D_{i-3})$, source s computes $(\mathsf{f}_2, \mathsf{a}_3)$ by (18) and (20), respectively, and computes $x^*_{\mathsf{ini}} \triangleq \phi^{[2]}_{\mathsf{ini}}(\mathsf{f}_2, \mathsf{a}_3)$. If Ack_{i-2} has not returned back to s by time $\max(S_{i-1}, D_{i-3}) + x^*_{\mathsf{ini}}$, then s will send P_i at that time. The subscript "ini" signifies that it is the initial decision at the decision time $\max(S_{i-1}, D_{i-3})$.

Remark 1: It is possible that $A_{i-2} \le \max(S_{i-1}, D_{i-3})$, i.e., at time $t = \max(S_{i-1}, D_{i-3})$, feedback Ack_{i-2} has already returned back to s. In this case, s will automatically skip Rule G4 and move on to the following Rule G5.

Rule G5: This rule is for the scenario that Ack_{i-2} returns back to s before s can "commit" the waiting time decision of Rule G4. To be precise, we will "activate" Rule G5 at time $t = \max(A_{i-2}, \max(S_{i-1}, D_{i-3})) = \max(A_{i-2}, S_{i-1})$ if Rule G4 has not been "committed" at that time yet. Specifically, at time $t = \max(A_{i-2}, S_{i-1})$, source s computes $(\mathsf{f}_1, \mathsf{a}_2)$ by (12) and (13), respectively, and computes $x^*_{\mathsf{a}} \triangleq \phi^{[2]}_{\mathsf{a}}(\mathsf{f}_1, \mathsf{a}_2)$. If Ack_{i-1} has not returned back to s by time $\max(A_{i-2}, S_{i-1}) + x^*_{\mathsf{a}}$, then s will send P_i at that time. The subscript "a" signifies that it is the decision under the scenario that after time $\max(S_{i-1}, D_{i-3})$ we have received *exactly one* more acknowledgement packet Ack_{i-2}.

Remark 2: Because the backward queue could have instantaneous delivery, i.e., $\mathbb{P}(Z_{i-1} = 0) > 0$, it is possible that the next feedback packet Ack_{i-1} returns back to s at the same time as the activation time of Rule G5. In this case, s will skip Rule G5 and move on to Rule G6 immediately.

Rule G6: This rule is for the scenario that the second acknowledgement packet Ack_{i-1} returns back to s before s can "commit" the waiting time decision of Rule G5. To be precise, we will "activate" Rule G6 at time $t = \max(A_{i-1}, \max(A_{i-2}, S_{i-1})) = A_{i-1}$. Specifically, at time $t = A_{i-1}$, source s computes a_1 in (11) and $x^*_{\mathsf{aa}} \triangleq \phi^{[2]}_{\mathsf{aa}}(\mathsf{a}_1)$, and will send P_i at time $A_{i-1} + x^*_{\mathsf{aa}}$. The subscript "aa" signifies that it is the decision under the scenario that after time $\max(S_{i-1}, D_{i-3})$ we have received *both* acknowledgement packets Ack_{i-2} and Ack_{i-1}.

Lemma 7: With the presence of an order-2 genie ($K = 2$), we can assume the optimal genie-aided scheme follows Rules G1, G4, G5, and G6 without loss of generality.

The proof is relegated to Appendix B of [21].

The Bellman equations corresponding to Rules G4, G5, and G6 fall into three different types. The type-1 Bellman equations are for Rule G6 and they are

$$\forall \mathsf{a}_1 \in \mathbb{N}^+,\ \text{we have}$$
$$f^{[2]}_{\mathsf{aa}}(\mathsf{a}_1) = \min_{x\in\mathbb{N}^+} \gamma(\mathsf{a}_1 + x, \mathbb{E}\{Y_i\}) - \mathsf{v}\cdot(\mathsf{a}_1 + x) + f^{[2]}_{\mathsf{a}}(0,0) \tag{55}$$

where $\gamma(\cdot,\cdot)$ was defined in (8); and $f^{[2]}_{\mathsf{a}}(\cdot,\cdot)$ is the type-2 Bellman equation to be described next.

The intuition of type-1 Bellman equations is the simplest. Rule G6 makes its decision at time $A_{i-1} = S_{i-1} + \mathsf{a}_1$ and we use x to denote the additional waiting time. The term $\gamma(\mathsf{a}_1 + x, \mathbb{E}\{Y_i\})$ quantifies the AoI cost of the decision. The term "$-\mathsf{v}\cdot(\mathsf{a}_1 + x)$" is the average-cost adjustment term of ACPS-semi-MDP. The term $f^{[2]}_{\mathsf{a}}(0,0)$ represents the next state value, the derivation of which is relegated to Appendix C of [21].

Recall the definitions of f_1, a_2, $\tilde{\mathsf{A}}_{i-1}$, and $m^+_{Y,1}(x)$ in (12), (13), (14), and (15), respectively. The type-2 Bellman equations are for Rule G5 and they are described as follows.

$$\forall 0 \le \mathsf{f}_1 \le \mathsf{a}_2,\ \text{we have}$$
$$f^{[2]}_{\mathsf{a}}(\mathsf{f}_1, \mathsf{a}_2)$$
$$= \min_{x\in\mathbb{N}^+}$$
$$\left\{ \sum_{k=1}^{x} \mathbb{P}(\tilde{\mathsf{A}}_{i-1} = \mathsf{a}_2 + k | \tilde{\mathsf{A}}_{i-1} > \mathsf{a}_2)\cdot f^{[2]}_{\mathsf{aa}}(\mathsf{a}_2 + k) \right. \tag{56}$$
$$+ \mathbb{P}(\tilde{\mathsf{A}}_{i-1} > \mathsf{a}_2 + x | \tilde{\mathsf{A}}_{i-1} > \mathsf{a}_2)\cdot \Big(\gamma\big(\mathsf{a}_2 + x, m^+_{Y,1}(x)\big)$$
$$\left. - \mathsf{v}\cdot(\mathsf{a}_2 + x) + f^{[2]}_{\mathsf{ini}}(\mathsf{f}_1 - (\mathsf{a}_2 + x), -x)\Big)\right\} \tag{57}$$

where $f^{[2]}_{\mathsf{ini}}(\cdot,\cdot)$ is the type-3 Bellman equation to be described later. The intuition of type-2 Bellman equations is as follows. We first note that $\tilde{\mathsf{A}}_{i-1}$ in (14) represents when s will receive the feedback Ack_{i-1} at a relative time scale versus S_{i-1}. Recall that the activation time of Rule G5 is $\max(A_{i-2}, S_{i-1}) = S_{i-1} + \mathsf{a}_2$. Since we would skip Rule G5 and activate Rule G6 instead if Ack_{i-1} has returned back to s before the (relative) decision time a_2, whenever we are making a decision for Rule G5, we are implicitly assuming Ack_{i-1} returns *after* (the relative) time a_2, i.e., we are under the event $\{\tilde{\mathsf{A}}_{i-1} > \mathsf{a}_2\}$. That is why both the *state transition probabilities* in (56) and (57) are conditioned on $\{\tilde{\mathsf{A}}_{i-1} > \mathsf{a}_2\}$.

The term in (56) represents the events that we will skip Rule G5 and switch to Rule G6, i.e., the scenario in which Ack_{i-1} returns at time $S_{i-1} + \mathsf{a}_2 + k$, no later than the tentative decision $S_{i-1} + \mathsf{a}_2 + x$. The derivation of the next state value $f^{[2]}_{\mathsf{aa}}(\mathsf{a}_2 + k)$ is relegated to Appendix A-D of [21].

The term $\gamma\big(\mathsf{a}_2 + x, m^+_{Y,1}(x)\big)$ in (57) quantifies the AoI cost of the decision. The first input argument $\mathsf{a}_2 + x = S_i - S_{i-1}$ is the time difference between sending P_i and P_{i-1}. The input argument $m^+_{Y,1}(x)$ is the average delay experienced by P_i, which is different from the simple delay $\mathbb{E}\{Y_i\}$ in (55) because P_i could potentially be blocked by P_{i-1}. Note that $m^+_{Y,1}(x)$ also appears in (26) and (43). See the discussion therein.

The term $-\mathsf{v}\cdot(\mathsf{a}_2 + x)$ is once again the average-cost adjustment term of ACPS-semi-MDP. The last term $f^{[2]}_{\mathsf{ini}}(\mathsf{f}_1 - (\mathsf{a}_2 + x), -x)$ represents the next state value, the derivation of which is relegated to Appendix D-B of [21].

The type-3 Bellman equations (for Rule G4) are described as follows. We first define a function of x_1 and x_2:

$$f^{[\mathrm{cmb}]}_{\mathsf{a}}(x_1, x_2)$$
$$= \mathbb{P}(\max(x_1 + Y_{i-1}, x_2) + Z_{i-1} = x_2)\cdot f^{[2]}_{\mathsf{aa}}(x_2)$$
$$+ \mathbb{P}(\max(x_1 + Y_{i-1}, x_2) + Z_{i-1} > x_2)\cdot f^{[2]}_{\mathsf{a}}(x_1, x_2) \tag{58}$$

which combines the type-1 and type-2 Bellman equations described previously. We also define

$$f^{[\mathrm{cmb}]}_{\mathsf{ini}}(x)$$
$$= \mathbb{P}(\max(x + Y_{i-1}, x^+) + Z_{i-1} = x^+)\cdot f^{[2]}_{\mathsf{a}}(x^+, x^+)$$
$$+ \mathbb{P}(\max(x + Y_{i-1}, x^+) + Z_{i-1} > x^+)\cdot f^{[2]}_{\mathsf{ini}}(x, x^+) \tag{59}$$

which combines $f^{[2]}_{\mathsf{a}}(\cdot,\cdot)$ and $f^{[2]}_{\mathsf{ini}}(\cdot,\cdot)$, where the type-3 Bellman equations $f^{[2]}_{\mathsf{ini}}(\cdot,\cdot)$ will be described shortly.

Recall the definitions of f_2, a_3, $\tilde{\mathsf{A}}_{i-2}$, and $m^+_{Y,2}(x)$ in (18), (20), (21), and (22), respectively. The type-3 Bellman equations become

$$\forall (\mathsf{f}_2, \mathsf{a}_3)\ \text{satisfying}\ \mathsf{f}_2 \le \mathsf{a}_3 \le \mathsf{f}_2^+,\ \text{we have}$$
$$f^{[2]}_{\mathsf{ini}}(\mathsf{f}_2, \mathsf{a}_3)$$
$$= \min_{x\in\mathbb{N}^+}$$
$$\left\{ \sum_{k=1}^{x}\sum_{y=1}^{y_{\max}} \mathbb{P}(Y_{i-2} = y, \tilde{\mathsf{A}}_{i-2} = \mathsf{a}_3^+ + k | \tilde{\mathsf{A}}_{i-2} > \mathsf{a}_3^+) \right.$$
$$\cdot f^{[\mathrm{cmb}]}_{\mathsf{a}}\left((\mathsf{f}_2 + y)^+, \mathsf{a}_3^+ + k\right) \tag{60}$$
$$+ \mathbb{P}(\tilde{\mathsf{A}}_{i-2} > \mathsf{a}_3^+ + x | \tilde{\mathsf{A}}_{i-2} > \mathsf{a}_3^+)\cdot \Big(\gamma\big(\mathsf{a}_3^+ + x, m^+_{Y,2}(x)\big)$$
$$- \mathsf{v}\cdot(\mathsf{a}_3^+ + x)\Big) \tag{61}$$
$$+ \sum_{y=1}^{y_{\max}} \mathbb{P}\left(Y_{i-2} = y, \tilde{\mathsf{A}}_{i-2} > \mathsf{a}_3^+ + x \,\middle|\, \tilde{\mathsf{A}}_{i-2} > \mathsf{a}_3^+\right)$$
$$\left. \cdot f^{[\mathrm{cmb}]}_{\mathsf{ini}}\left((\mathsf{f}_2 + y)^+ - (\mathsf{a}_3^+ + x)\right)\right\} \tag{62}$$

To explain the intuition of (60)–(62), we need the following lemma, the proof of which is relegated to Appendix E of [21].

Lemma 8: With the presence of an order-2 genie, we have

$$\max(S_{i-1}, D_{i-3}) = S_{i-1} + (\mathsf{a}_3)^+ \tag{63}$$
$$\text{and}\quad \mathsf{f}_2 \le \mathsf{a}_3 \le (\mathsf{f}_2)^+. \tag{64}$$

We now provide the intuition of type-3 Bellman equations. Given a fixed pair of $(\mathsf{f}_2, \mathsf{a}_3)$ values, $\tilde{\mathsf{A}}_{i-2}$, defined in (21), represents when s will receive the feedback Ack_{i-2}. By (63), source s applies Rule G4 at time a_3^+ at a relative time scale of S_{i-1}. Since we would immediately skip Rule G4 and activate Rule G5 instead if Ack_{i-2} has returned back to s before the decision time a_3^+, whenever we are making a decision for Rule

G4, we are implicitly assuming Ack_{i-2} returns *after* time a_3^+. That is why all the *state transition probabilities* in (60), (61), and (62) are conditioned on $\{\tilde{\mathsf{A}}_{i-2} > \mathsf{a}_3^+\}$.

The term in (60) represents the events that we will skip Rule G4 and switch to Rule G5 instead, i.e., Ack_{i-2} returns back to s before the tentative decision $S_{i-1} + \mathsf{a}_3^+ + x$. The derivation of the next state value $f_{\mathsf{a}}^{[\mathrm{cmb}]}\left((\mathsf{f}_2 + y)^+, \mathsf{a}_3^+ + k\right)$ is relegated to Appendix F-A of [21].

Nonetheless, unlike the type-2 Bellman equations, there is some subtlety for the type-3 Bellman equations. That is, even if we skip Rule G4 and move on to Rule G5 (because of the early return of Ack_{i-2}), there is a chance that Ack_{i-1} will return to s at the same time as Ack_{i-2} since the backward delay Z_{i-1} could be zero with some positive probability. If that happens, we will immediately skip Rule G5 again and move on to Rule G6 instead. As a result, we introduce the combined value function $f_{\mathsf{a}}^{[\mathrm{cmb}]}(\cdot,\cdot)$ in (58), which further allows for the switching to Rule G6 depending on the arrival time of Ack_{i-1}. Namely, (58) carefully quantifies the probabilities of staying in Rule G5 versus switching to Rule G6, by discussing the corresponding events in terms of Y_{i-1} and Z_{i-1}. Also see the detailed analysis in Appendix F-A of [21].

The term $\gamma\left(\mathsf{a}_3^+ + x, m_{Y,2}^+(x)\right)$ in (61) quantifies the AoI cost of the decision. Herein, the average delay experienced by P_i is once again lengthened due to the fact that P_i could potentially be blocked by P_{i-1} and P_{i-2} since when under Rule G4, source s only knows that P_{i-3} has been delivered (because Ack_{i-3} has returned) but has no knowledge about the delivery times of P_{i-1} and P_{i-2}. The expected time needed to deliver P_i is characterized by the $m_{Y,2}^+(x)$ term defined in (22). See Appendix F.B of [21] for detailed discussion. Plugging $m_{Y,2}^+(x)$ into $\gamma(\cdot,\cdot)$ gives us the AoI cost of the decision.

The term $-\mathsf{v}\cdot(\mathsf{a}_3^+ + x)$ is the average-cost adjustment term of ACPS-semi-MDP.

The last term $f_{\mathsf{ini}}^{[\mathrm{cmb}]}\left((\mathsf{f}_2 + y)^+ - (\mathsf{a}_3^+ + x)\right)$ represents the next state value function when sending $P_{i_{\mathrm{nx}}}$ with $i_{\mathrm{nx}} = i+1$, the derivation of which is relegated to Appendix F-C of [21].

Even though we have figured out the next state values

$$\mathsf{f}_2^{[\mathrm{new}]} = (\mathsf{f}_2 + y)^+ - (\mathsf{a}_3^+ + x) \tag{65}$$

$$\mathsf{a}_3^{[\mathrm{new}]} = \left(\mathsf{f}_2^{[\mathrm{new}]}\right)^+ \tag{66}$$

in Appendix F-C of [21], there is some subtlety when considering the next state value function, i.e., even though one may expect that we would apply Rule G4 again for packet $P_{i_{\mathrm{nx}}}$ under state values $(\mathsf{f}_2^{[\mathrm{new}]}, \mathsf{a}_3^{[\mathrm{new}]})$, we may skip Rule G4 and move on to Rule G5 instead if $\mathsf{Ack}_{i_{\mathrm{nx}}-2} = \mathsf{Ack}_{i-1}$ returns back to s before we make the decision at time $\max(S_{i_{\mathrm{nx}}-1}, D_{i_{\mathrm{nx}}-3})$. Also see our discussion of the $f_{\mathsf{a}}^{[\mathrm{cmb}]}(\cdot,\cdot)$ term in (60).

As a result, we introduce the combined value function $f_{\mathsf{ini}}^{[\mathrm{cmb}]}(\cdot)$ in (59), which further allows for the switching to Rule G5 depending on the arrival time of $\mathsf{Ack}_{i_{\mathrm{nx}}-2}$. Specifically, (59) carefully quantifies the probabilities of staying in Rule G4 versus switching to Rule G5 for packet $P_{i_{\mathrm{nx}}}$, by discussing the corresponding events in terms of $Y_{i_{\mathrm{nx}}-2} = Y_{i-1}$ and $Z_{i_{\mathrm{nx}}-2} = Z_{i-1}$. By (65), (66), and (59), we set the next-state value function in (62) to $f_{\mathsf{ini}}^{[\mathrm{cmb}]}\left((\mathsf{f}_2 + y)^+ - (\mathsf{a}_3^+ + x)\right)$. Also see the detailed analysis in Appendix F-A of [21].

In the end, the Bellman equations for the optimal order-2 genie scheme consist of (55), (57), and (62).

B. Remark on The Computation

We use *value iteration* to find a scalar v and functions $f_{\mathsf{aa}}^{[2]}(\mathsf{a}_1)$, $f_{\mathsf{a}}^{[2]}(\mathsf{f}_1, \mathsf{a}_2)$, and $f_{\mathsf{ini}}^{[2]}(\mathsf{f}_2, \mathsf{a}_3)$ that satisfy (55), (57), and (62) with the ground state value hardwired to $f_{\mathsf{a}}^{[2]}(0,0) = 0$. The final v value is the optimal AoI of the order-2 genie-aided scheme, thus a new lower bound $\mathsf{lb}_{\mathrm{new}}^{[2]}$.

To ensure computability, we prove that the "unbounded" version of (55), (57), and (62) can be replaced by their "bounded" counterparts without loss of generality using the following three steps based on almost identical arguments as in Secs. IV-B and V-B. Step 1: We argue that there is no need to change the unbounded version (55) since (55) can be solved analytically. See the discussion in Secs. IV-B and V-B.

Step 2: We argue that we can limit the search range of the minimizing x of (57) to be

$$\mathcal{X}_{\mathsf{a}}(\mathsf{f}_1, \mathsf{a}_2) = [0, (\mathsf{f}_1 + y_{\max} - \mathsf{a}_2)^+ + z_{\max}] \tag{67}$$

and limit the search range of the minimizing x of (62) to be

$$\mathcal{X}_{\mathsf{ini}}(\mathsf{f}_2, \mathsf{a}_3) = [0, \left(\max(\mathsf{f}_2 + y_{\max}, \mathsf{a}_3) + z_{\max} - \mathsf{a}_3^+\right)^+] \tag{68}$$

without loss of generality. The reason is that all the probability terms, e.g., $\mathbb{P}(\tilde{\mathsf{A}}_{i-1} > \mathsf{a}_2 + x | \tilde{\mathsf{A}}_{i-1} > \mathsf{a}_2)$ in (57), remain unchanged if x is larger than the specified ranges in (67) and (68). See the discussion in Secs. IV-B and V-B.

Step 3: We again use the concept of a *self-contained input parameter set* in Definition 1. Namely, we only need to solve the Bellman equations for a bounded input parameter set Ω_{sc} such that the evaluation of both the left-hand and right-hand sides of (57) and (62) are "fully covered" within the set Ω_{sc}.

Lemma 9: The Ω_{sc} that contains all $(\mathsf{f}_1, \mathsf{a}_2)$ satisfying

$$\mathsf{f}_1 \in [0, 2y_{\max}], \quad (\mathsf{a}_2 - \mathsf{f}_1) \in [0, z_{\max}]; \tag{69}$$

and all $(\mathsf{f}_2, \mathsf{a}_3)$ satisfying

$$\text{either } \mathsf{f}_2 = \mathsf{a}_3 \in [0, y_{\max}], \tag{70}$$

$$\text{or } \begin{cases} -\max(y_{\max}, z_{\max}) - z_{\max} \le \mathsf{f}_2 \le \mathsf{a}_3 \le 0, \\ -y_{\max} - z_{\max} \le \mathsf{a}_3 \le \mathsf{f}_2 + z_{\max} \end{cases} \tag{71}$$

is self-contained w.r.t. the Bellman equations (57) and (62), provided we use the search ranges of x defined in (67)–(68).

The proof is relegated to Appendix F-D of [21].

After applying this 3-step process, we have a finite set of Bellman equations that can be numerically solved.

C. The Order-2 Achievability Scheme

The structure of the order-2 achievability scheme is similar to the order-2 genie-aided scheme described earlier. Therefore, we focus on providing a complete description and leave most of the derivations to Appendix G of [21].

Consider three arbitrarily given "waiting time functions" $\theta_{\mathsf{ini}}^{[2]}(\mathsf{f}, \mathsf{a})$, $\theta_{\mathsf{a}}^{[2]}(\mathsf{f}, \mathsf{a})$, and $\theta_{\mathsf{aa}}^{[2]}(\mathsf{a})$, where the input parameters f and a are integers, which can sometimes be negative.

Rule A4: At time $t = \max(S_{i-1}, A_{i-3})$, source s computes the $(\mathsf{f}_2, \mathsf{a}_3)$ values by (18) and (20), respectively. It then uses them to compute $x^*_{\mathsf{ini}} \triangleq \theta^{[2]}_{\mathsf{ini}}(\mathsf{f}_2, \mathsf{a}_3)$. If Ack_{i-2} has not returned back to s by time $\max(S_{i-1}, A_{i-3}) + x^*_{\mathsf{ini}}$, then s will send P_i at that time. The subscript "ini" signifies that it is the initial decision at the decision time $\max(S_{i-1}, A_{i-3})$.

Rule A5: Suppose Ack_{i-2} returns back to s before s can "commit" the waiting time decision of Rule A4. At time $t = \max(A_{i-2}, \max(S_{i-1}, A_{i-3})) = \max(A_{i-2}, S_{i-1})$, source s computes $(\mathsf{f}_1, \mathsf{a}_2)$ by (12) and (13), respectively. It then computes $x^*_{\mathsf{a}} \triangleq \theta^{[2]}_{\mathsf{a}}(\mathsf{f}_1, \mathsf{a}_2)$. If Ack_{i-1} has not returned back to s by time $\max(A_{i-2}, S_{i-1}) + x^*_{\mathsf{a}}$, then s will send P_i at that time. The subscript "a" signifies that it is the decision under the scenario that after time $\max(S_{i-1}, A_{i-3})$ we have received exactly one more acknowledgement packet Ack_{i-2}.

Rule A6: Suppose Ack_{i-1} returns back to s before s can "commit" the waiting time decision of Rule A5. At time $\max(A_{i-1}, \max(A_{i-2}, S_{i-1})) = A_{i-1}$, source s computes a_1 by (11), computes $x^*_{\mathsf{aa}} \triangleq \theta^{[2]}_{\mathsf{aa}}(\mathsf{a}_1)$, and will send P_i at time $A_{i-1} + x^*_{\mathsf{aa}}$. The subscript "aa" signifies that it is the decision under the scenario that after time $\max(S_{i-1}, A_{i-3})$ we have received *both* acknowledgement packets Ack_{i-2} and Ack_{i-1}.

The optimal choices of the waiting time functions can be found by solving the Bellman equations described below, and we leave the detailed derivation to Appendix G of [21].

The following type-1 Bellman equations are for Rule A6:

$$\forall \mathsf{a}_1 \in \mathbb{N}^+, \text{ we have}$$
$$g^{[2]}_{\mathsf{aa}}(\mathsf{a}_1) = \min_{x \in \mathbb{N}^+} \gamma(\mathsf{a}_1 + x, \mathbb{E}\{Y_i\}) - \mathsf{v} \cdot (\mathsf{a}_1 + x) + g^{[2]}_{\mathsf{a}}(0, 0) \tag{72}$$

where $\gamma(\cdot, \cdot)$ was defined in (8); and $g^{[2]}_{\mathsf{a}}(\cdot, \cdot)$ is the type-2 Bellman equation to be described next.

The type-2 Bellman equations are for Rule A5. Recall the definitions of f_1, a_2, $\tilde{\mathsf{A}}_{i-1}$, and $m^+_{Y,1}(x)$ in (12), (13), (14), and (15), respectively. The type-2 Bellman equations then become

$$\begin{aligned}
&\forall 0 \le \mathsf{f}_1 \le \mathsf{a}_2, \text{ we have}\\
&g^{[2]}_{\mathsf{a}}(\mathsf{f}_1, \mathsf{a}_2)\\
&= \min_{x \in \mathbb{N}^+}\\
&\Bigg\{ \sum_{k=1}^{x} \mathbb{P}(\tilde{\mathsf{A}}_{i-1} = \mathsf{a}_2 + k | \tilde{\mathsf{A}}_{i-1} > \mathsf{a}_2) \cdot g^{[2]}_{\mathsf{aa}}(\mathsf{a}_2 + k) && (73)\\
&+ \mathbb{P}(\tilde{\mathsf{A}}_{i-1} > \mathsf{a}_2 + x | \tilde{\mathsf{A}}_{i-1} > \mathsf{a}_2) \cdot \Big(\gamma\big(\mathsf{a}_2 + x, m^+_{Y,1}(x)\big)\\
&- \mathsf{v} \cdot (\mathsf{a}_2 + x) + g^{[2]}_{\mathsf{ini}}\left(\mathsf{f}_1 - (\mathsf{a}_2 + x), -x\right) \Big) \Bigg\} && (74)
\end{aligned}$$

where $g^{[2]}_{\mathsf{ini}}(\cdot, \cdot)$ will be described next.

The type-3 Bellman equations $g^{[2]}_{\mathsf{ini}}(\cdot, \cdot)$ are described as follows. We first define the following functions of x_1 and x_2:

$$\begin{aligned}
&g^{[\text{cmb}]}_{\mathsf{a}}(x_1, x_2)\\
&= \mathbb{P}(\max(x_1 + Y_{i-1}, x_2) + Z_{i-1} = x_2) g^{[2]}_{\mathsf{aa}}(x_2)\\
&\quad + \mathbb{P}(\max(x_1 + Y_{i-1}, x_2) + Z_{i-1} > x_2) g^{[2]}_{\mathsf{a}}(x_1, x_2)
\end{aligned} \tag{75}$$

which combines the type-1 and type-2 Bellman equations described previously. We also define

$$\begin{aligned}
&g^{[\text{cmb}]}_{\mathsf{ini}}(x_1, x_2)\\
&= \sum_{\tilde{y}=1}^{y_{\max}} \mathbb{P}\Big(Y_{i-1} = \tilde{y}, \max(x_1 + \tilde{y}, x_2) + Z_{i-1} = x_2\Big)\\
&\quad \cdot g^{[\text{cmb}]}_{\mathsf{a}}((x_1 + \tilde{y})^+, x_2)\\
&\quad + \mathbb{P}(\max(x_1 + Y_{i-1}, x_2) + Z_{i-1} > x_2) \cdot g^{[2]}_{\mathsf{ini}}(x_1, x_2)
\end{aligned} \tag{76}$$

which combines the new function in (75) with the type-3 Bellman equations, the latter of which will be described shortly.

Recall the definitions of f_2, a_3, $\tilde{\mathsf{A}}_{i-2}$, and $m^+_{Y,2}(x)$ in (18), (20), (21), and (22), respectively. The type-3 Bellman equations then become

$$\begin{aligned}
&\forall (\mathsf{f}_2, \mathsf{a}_3) \text{ satisfying } \mathsf{f}_2 \le \mathsf{a}_3, \text{ we have}\\
&g^{[2]}_{\mathsf{ini}}(\mathsf{f}_2, \mathsf{a}_3)\\
&= \min_{x \in \mathbb{N}^+}\\
&\Bigg\{ \sum_{k=1}^{x} \sum_{y=1}^{y_{\max}} \mathbb{P}(Y_{i-2} = y, \tilde{\mathsf{A}}_{i-2} = \mathsf{a}_3^+ + k | \tilde{\mathsf{A}}_{i-2} > \mathsf{a}_3^+)\\
&\qquad \cdot g^{[\text{cmb}]}_{\mathsf{a}}\left((\mathsf{f}_2 + y)^+, \mathsf{a}_3^+ + k\right) && (77)\\
&+ \mathbb{P}(\tilde{\mathsf{A}}_{i-2} > \mathsf{a}_3^+ + x | \tilde{\mathsf{A}}_{i-2} > \mathsf{a}_3^+) \cdot \Big(\gamma\big(\mathsf{a}_3^+ + x, m^+_{Y,2}(x)\big)\\
&\qquad - \mathsf{v} \cdot (\mathsf{a}_3^+ + x) \Big) && (78)\\
&+ \sum_{y,z} \mathbb{P}\left(Y_{i-2} = y, Z_{i-2}, = z, \tilde{\mathsf{A}}_{i-2} > \mathsf{a}_3^+ + x \,\middle|\, \tilde{\mathsf{A}}_{i-2} > \mathsf{a}_3^+ \right)\\
&\cdot g^{[\text{cmb}]}_{\mathsf{ini}}\left((\mathsf{f}_2 + y)^+ - (\mathsf{a}_3^+ + x), \right.\\
&\qquad \left. \max(\mathsf{f}_2 + y, \mathsf{a}_3) + z - (\mathsf{a}_3^+ + x) \right) \Bigg\} && (79)
\end{aligned}$$

We can further convert the above unbounded versions of Bellman equations in (72), (74), and (79) to their equivalent bounded versions, in ways almost identical to the discussion in Sec. VI-B. That is, Step 1: There is no need to change the unbounded version (72) since (72) can be solved analytically.

Step 2: We can limit the search range of the minimizing x of (74) to be the $\mathcal{X}_{\mathsf{a}}$ defined in (67), and limit the search range of the minimizing x of (79) to be the $\mathcal{X}_{\mathsf{ini}}$ defined in (68) without loss of generality.

Step 3: We once again use the concept of a *self-contained input parameter set*, and only consider the $(\mathsf{f}_1, \mathsf{a}_2)$ and $(\mathsf{f}_2, \mathsf{a}_3)$ values satisfying

$$0 \le \mathsf{f}_1 \le \mathsf{a}_2 \le y_{\max} + z_{\max} + \max(y_{\max}, z_{\max}); \tag{80}$$
$$\begin{cases} -\max(y_{\max}, z_{\max}) - y_{\max} - 2z_{\max} \le \mathsf{f}_2 \le y_{\max} \\ \mathsf{f}_2 \le \mathsf{a}_3 \le y_{\max} + z_{\max} \end{cases} \tag{81}$$

The description of the Bellman equations for the optimal order-2 achievability scheme is now complete.

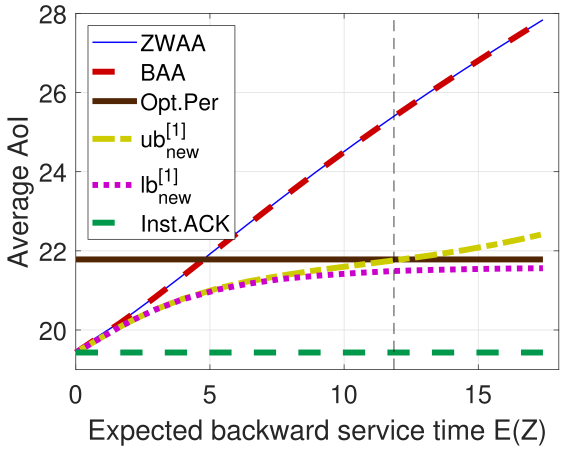


Fig. 4. $(\mathsf{lb}^{[1]}_{\text{new}}, \mathsf{ub}^{[1]}_{\text{new}})$ versus existing results — log-normal $\mathbb{P}_Y$.

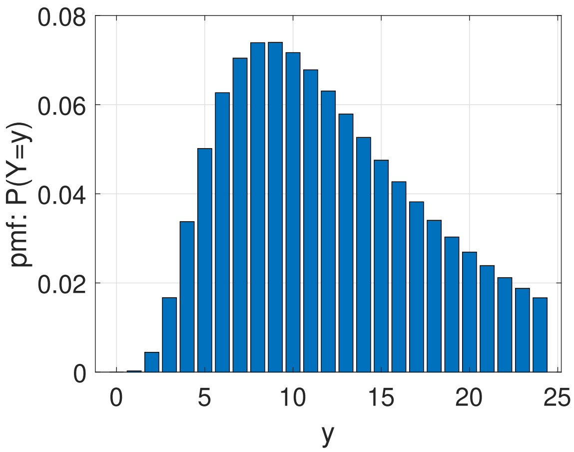


Fig. 5. The pmf of the quantized log-normal distribution of the forward service time Y.

VII. Numerical Evaluation

For any given $[M_L, M_U]$, μ, and σ^2 values, we say a random variable Q is integer-quantized, $[M_L, M_U]$-truncated, log-normal with parameters (μ, σ^2) if $\forall q \in [M_L, M_U]$,

$$\mathbb{P}(Q = q) \propto \mathbb{P}\big(W \in (q - 0.5, q + 0.5]\big)$$

where W is log-normal with parameters (μ, σ^2). That is, we first truncate the values outside $[M_L, M_U]$ so the total probability becomes strictly less than one, and then we *proportionally scale it* so that the total probability is back to one.

Fig. 4 plots the order-1 converse and achievability bounds $(\mathsf{lb}^{[1]}_{\text{new}}, \mathsf{ub}^{[1]}_{\text{new}})$ in Secs. IV and V, versus existing bounds zwaa, baa, opt.per, and inst.ack, for which we assume Y_i (resp. Z_i) is integer-quantized, $[1, 24]$-truncated (resp. $[0, 24]$-truncated), log-normal with parameters (μ_Y, σ_Y^2) (resp. (μ_Z, σ_Z^2)). The truncation intervals are slightly different since we assume $Y_i \geq 1$ and $Z_i \geq 0$ in our setting, see Sec. II. We fix $(\mu_Y, \sigma_Y^2) = (2.5, 0.6^2)$ and set $\sigma_Z^2 = 0.6^2$ while varying the value of μ_Z to change the expected backward delay. The pmf of the forward service time Y is illustrated in Fig. 5, a unimodal curve with a unique peak at $Y = 9$. A thin vertical line $\mathbb{E}(Y) = 11.86$ is drawn in Fig. 4 to indicate when the *expected backward service time* $\mathbb{E}(Z)$ is equal to the average forward delay 11.86.

As can be seen, when $\mathbb{E}(Z) = 0$, the upper bound baa and the lower bound inst.ack coincide since baa is indeed the optimal scheme in the *instantaneous feedback* setting [10]. However, for the general cases of $\mathbb{E}(Z) > 0$, none of the

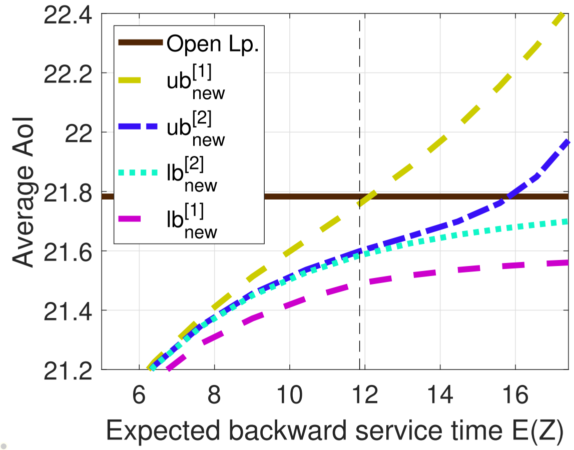


Fig. 6. $(\mathsf{lb}^{[K]}_{\text{new}}, \mathsf{ub}^{[K]}_{\text{new}})$ for $K = 1$ and 2, respectively, — log-normal $\mathbb{P}_Y$.

existing bounds zwaa, baa, opt.per and inst.ack is tight. In Fig. 4, we plot the first-order ($K = 1$) converse lower bound $\mathsf{lb}^{[1]}_{\text{new}}$ and achievability upper bound $\mathsf{ub}^{[1]}_{\text{new}}$, respectively. As can be seen, $\mathsf{lb}^{[1]}_{\text{new}}$ and $\mathsf{ub}^{[1]}_{\text{new}}$ closely follow each other for a wide range of $\mathbb{E}(Z)$ values.

In fact, the smaller the $\mathbb{E}(Z)$, the smaller the gap ratio $\frac{\mathsf{ub}^{[1]}_{\text{new}} - \mathsf{lb}^{[1]}_{\text{new}}}{\mathsf{lb}^{[1]}_{\text{new}}}$. Specifically, it is less than 0.28% when $\mathbb{E}(Z) \leq 6.40$ and it grows to 1.24% when $\mathbb{E}(Z) = \mathbb{E}(Y) = 11.86$. The bounds do diverge for $\mathbb{E}(Z) \geq \mathbb{E}(Y)$, also see our subsequent discussion in Sec. VIII. We can also sharpen the upper bound by $\overline{\mathsf{ub}}^{[1]}_{\text{new}} \triangleq \min(\mathsf{ub}^{[1]}_{\text{new}}, \mathsf{opt.per})$. The gap ratio $\frac{\overline{\mathsf{ub}}^{[1]}_{\text{new}} - \mathsf{lb}^{[1]}_{\text{new}}}{\mathsf{lb}^{[1]}_{\text{new}}}$ is less than 1.24% for all $\mathbb{E}(Z)$. The pair $(\mathsf{lb}^{[1]}_{\text{new}}, \overline{\mathsf{ub}}^{[1]}_{\text{new}})$ thus tightly brackets the true $\mathsf{avg.aoi}^*$, the optimum value of the ACPS-MDP problem (10), (5), and (6), for all our choices of different μ_Z values.

While $\mathsf{lb}^{[1]}_{\text{new}}$ and $\mathsf{ub}^{[1]}_{\text{new}}$ have already bracketed $\mathsf{avg.aoi}^*$ tightly, we are interested in learning whether it is $\mathsf{lb}^{[1]}_{\text{new}}$ or $\mathsf{ub}^{[1]}_{\text{new}}$ that is farther away from the optimum $\mathsf{avg.aoi}^*$. To that end, we evaluate the order $K = 2$ converse lower bound $\mathsf{lb}^{[2]}_{\text{new}}$ and achievability upper bound $\mathsf{ub}^{[2]}_{\text{new}}$, respectively, and plot them in Fig. 6. The gap ratio between the $K = 2$ pair $(\mathsf{lb}^{[2]}_{\text{new}}, \mathsf{ub}^{[2]}_{\text{new}})$, computed by $\frac{\mathsf{ub}^{[2]}_{\text{new}} - \mathsf{lb}^{[2]}_{\text{new}}}{\mathsf{lb}^{[2]}_{\text{new}}}$, is much smaller than the $K = 1$ pair $(\mathsf{lb}^{[1]}_{\text{new}}, \mathsf{ub}^{[1]}_{\text{new}})$. Specifically, the gap ratio of $(\mathsf{lb}^{[2]}_{\text{new}}, \mathsf{ub}^{[2]}_{\text{new}})$ is less than 0.065% for all the data points satisfying $\mathbb{E}(Z) \leq \mathbb{E}(Y) = 11.86$.

The comparison among the three achievability schemes opt.per, $\mathsf{ub}^{[1]}_{\text{new}}$, and $\mathsf{ub}^{[2]}_{\text{new}}$ also gives us practical guidelines when to switch between different classes of schemes. For example, if $\mathbb{E}(Z) \leq 6$, then $\mathsf{ub}^{[1]}_{\text{new}} \approx \mathsf{ub}^{[2]}_{\text{new}} < \mathsf{opt.per}$. There is thus no need to use the more complicated order-2 achievability scheme and we can simply use the order-1 achievability scheme to harvest significant AoI savings over the open-loop opt.per solutions. If $6 < \mathbb{E}(Z) < 15.5$, then the order-2 scheme starts to outperform the order-1 scheme and should be the choice for a performance conscientious user. Finally, if $15.5 < \mathbb{E}(Z)$, then the open-loop scheme opt.per starts to dominate and one can simply discard all feedback information while comfortably knowing that the gap ratio between $\mathsf{lb}^{[2]}_{\text{new}}$ and opt.per is $\leq 0.44\%$, a near-optimality guarantee for opt.per.

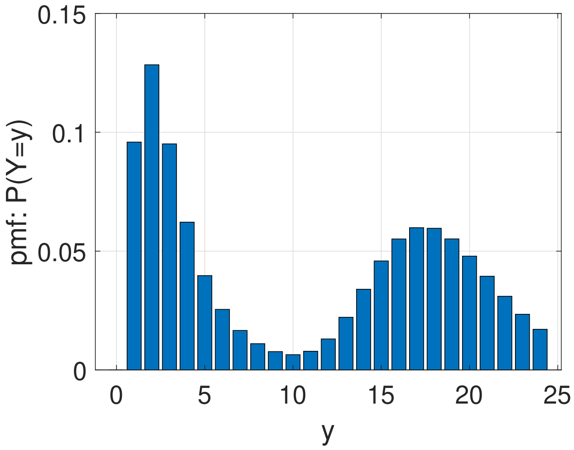


Fig. 7. The pmf of the quantized bimodal composite log-normal distribution of the forward service time Y.

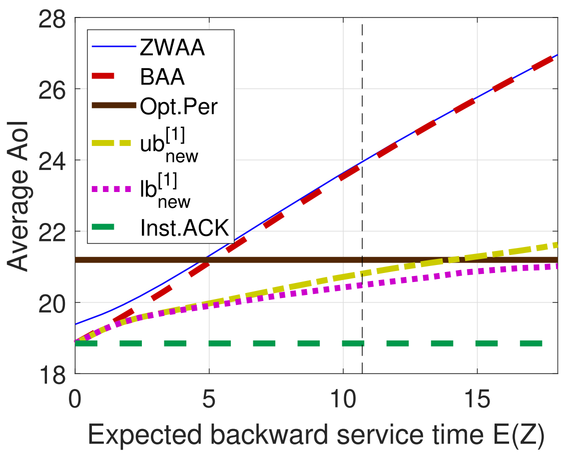


Fig. 8. $(\mathsf{lb}^{[1]}_{\text{new}}, \mathsf{ub}^{[1]}_{\text{new}})$ versus existing results — composite log-normal $\mathbb{P}_Y$.

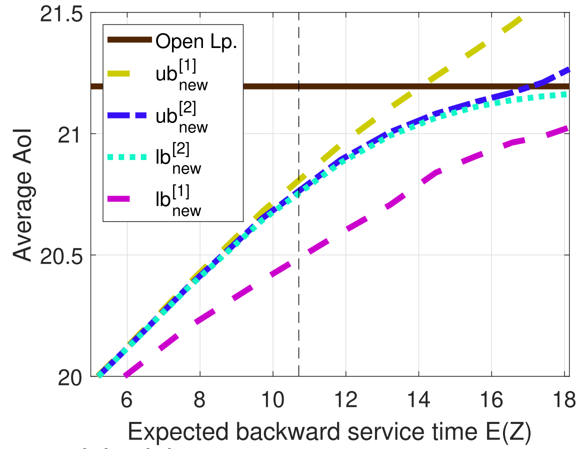


Fig. 9. $(\mathsf{lb}^{[K]}_{\text{new}}, \mathsf{ub}^{[K]}_{\text{new}})$ for $K = 1$ and 2, respectively, — composite log-normal $\mathbb{P}_Y$.

We repeat the same numerical evaluations but this time we examine a bimodal distribution of the forward service time Y. Specifically, we let $\mathbb{P}_Y$ be a $(0.5, 0.5)$ mixture of two integer-quantized $[1, 24]$-truncated log-normals with parameters $(\mu_{Y_1}, \sigma^2_{Y_1}) = (2.9, 0.2^2)$ and $(\mu_{Y_2}, \sigma^2_{Y_2}) = (1.0, 0.7^2)$, respectively. That is, $\mathbb{P}_Y$ is bimodal composite-log-normal as illustrated in Fig. 5. We reuse the same feedback service distribution of $\mathbb{P}_Z$. That is, Z is a simple $[0, 24]$-truncated log-normal with parameters $\sigma^2_Z = 0.6^2$ and we vary the value of μ_Z to change the expected backward delay.

Fig. 8 repeats the same experiment of Fig. 4 using the new bimodal $\mathbb{P}_Y$ in Fig. 7. The thin vertical line indicates the new $\mathbb{E}(Y) = 10.71$. The gap ratio between $(\mathsf{lb}^{[1]}_{\text{new}}, \mathsf{ub}^{[1]}_{\text{new}})$ is less than 0.54% when $\mathbb{E}(Z) \leq 5.83$ and grows to 1.6% when $\mathbb{E}(Z) = \mathbb{E}(Y) = 10.71$. If we define the improved upper bound $\overline{\mathsf{ub}}^{[1]}_{\text{new}} \triangleq \min(\mathsf{ub}^{[1]}_{\text{new}}, \mathsf{opt.per})$, then the largest gap ratio between $\mathsf{lb}^{[1]}_{\text{new}}$ and $\overline{\mathsf{ub}}^{[1]}_{\text{new}}$ is 1.9% for all $\mathbb{E}(Z)$.

Under the instantaneous ACK setting, the gap between zwaa and baa is larger if $\mathbb{P}(Y)$ happens to be bimodal, see the diverging gap between zwaa and baa in Fig. 8 when $\mathbb{E}(Z) = 0$. This is why we are interested in bimodal $\mathbb{P}_Y$ of Fig. 8 in the first place. In both Figs. 4 and 8, the gap between zwaa and ub_{new} continues to widen when $\mathbb{E}(Z)$ grows. Namely, the AoI improvement of our new achievability schemes over the naive zero-wait policy gets bigger since our schemes utilize the delayed feedback in a near-optimal way. It also shows that the performance of Best-After-ACK is quite bad when $\mathbb{E}(Z) > 0$ and one really should not take a pessimistic stance that sends P_i only after receiving Ack_{i-1}.

Similar to Fig. 6, we compare $(\mathsf{lb}^{[K]}_{\text{new}}, \mathsf{ub}^{[K]}_{\text{new}})$ for $K = 1, 2$ in Fig. 9. As can be seen, the order-2 lower and upper bounds are significantly tighter than their order-1 counterparts. The gap ratio between $(\mathsf{lb}^{[2]}_{\text{new}}, \mathsf{ub}^{[2]}_{\text{new}})$ is less than 0.044% for all $\mathbb{E}(Z) < \mathbb{E}(Y)$. From a practical perspective, the results have characterized the $\mathsf{avg.aoi}^*$ in this numerical example.

Figs. 4 to 9 show that our bounds are numerically tight for two very distinct distributions, e.g., unimodal versus bimodal. In other not-reported experiments, the tightness persists for both uniform and geometric delay distributions as well.

VIII. Further Discussion

A. Contribution

While our results do not innovate any MDP methodology to solve the problem (just like most AoI minimization results can be viewed as a specialization of a general MDP problem), new observations are made to facilitate tractable analysis and computation. We summarize these observations as below.

Observation 1: The state space of AoI minimization under 2-way queues is exceedingly large. The reason is that under the 2-way delay setting, there is a temporal dependence across multiple waiting time decisions. This is in contrast with the instantaneous feedback setting, for which each instantaneous feedback severs the temporal dependence across multiple decisions and greatly simplifies the state space [10]. To overcome this challenge, the first innovation of ours is to propose new ways of reducing the state space.

In the order-K achievability schemes, we judicially impose the condition $S_i \geq A_{i-K-1}$ to reduce the state space. On the converse side, we derive rigorous AoI lower bounds by introducing the order-K genie-aided schemes, the first of its kind in the AoI literature. Our deceptively simple order-K genie definition in (23) was obtained after numerous unsuccessful attempts during the development stage. Furthermore, by providing companying converse bounds $\mathsf{lb}^{[K]}_{\text{new}}$, we can numerically compute the performance loss of our design (imposing $S_i \geq$

A_{i-K-1}) when compared to $\mathsf{avg.aoi}^*$ and show that our schemes are near-optimal from a practical perspective.

Observation 2: The second innovation is to notice that even though the *uncertainty* faced by source s is reduced during each passing time slot, the major *change* of the situation happens only when we receive a new feedback Ack_j for some $j < i$. Therefore, we can let the decision maker s "simulate" the decision process during each individual time slot and only make meaningful new decisions at each major event (when Ack_j returns to s). This important observation converts the scheduling problem to a sequential opportunistic policy of (i) first propose a *waiting time*; (ii) wait and see whether there is any major event (when Ack_j returns to s) before the proposed waiting time; (iii) If so, abandon the proposed waiting time and propose a new waiting time instead. If not, commit to the proposed waiting time. By rigorously formulating the above sequential policy, we convert the traditional MDP problem into a semi-MDP problem with special structures, which is much more tractable for numerical computation.

Observation 3: Even with the structure of the semi-MDP formulation, the derivation of the Bellman equations is highly non-trivial, which is evidenced by the involved expressions and various subtle considerations in our Bellman equations.

The above three observations have addressed the critical challenges when analyzing the 2-way queue systems. The results answer an important problem that have been open for several years despite the early works that completely solved the instantaneous ACK setting [10] and many follow-up results since then. Our approach also provides a clear road map about how to evaluate the AoI benefits under a delayed feedback setting for the first time in the literature.

In terms of the converse, our AoI lower bounds are the first and only results that govern $\mathsf{avg.aoi}^*$ in a 2-way delay setting. In terms of the acheivability, several schemes have been proposed based on the Best-After-ACK (BAA) designs [11], [12], [13], [15], [22], which, as discussed in Secs. II-A and VII, are far from optimum. The only existing non-BAA design is [24], a parallel work to this paper. Similar to our results, [24] shows significant AoI improvement over all BAA schemes.

This work and [24] have the following differences: (i) [24] focuses exclusively on *geometric service times* while this work allows for arbitrary[4] service time distributions; (ii) Under the sampler-controller framework introduced in [11] and [13], the authors of [24] study a controller-centric setting with an obedient sampler, while this work studies sampler-centric setting with an obedient controller. As a result, the settings are very different and incompatible to each other; (iii) [24] does not study any converse bound that governs *all* achievable schemes; (iv) [24] solves MDP problems under simplified/augmented state spaces, which, as discussed in Remark 1 of [24] "may not always be practical". In contrast, the state spaces in our achievability results capture *exactly* the available information at the sampler/source. The resulting schemes are guaranteed to be feasible; (v) [24] derives closed-form expressions of the average AoI for three simple suboptimal policies called Zero-Wait-1, Zero-Wait-2, and Wait-1, respectively.

B. Complexity

For $K = 1$, the performance bounds $(\mathsf{lb}^{[1]}_{\text{new}}, \mathsf{ub}^{[1]}_{\text{new}})$ are easily computable and their performance is reasonably close to optimality, see Sec. VII. Unfortunately, the complexity of computing $(\mathsf{lb}^{[2]}_{\text{new}}, \mathsf{ub}^{[2]}_{\text{new}})$ is high when $(y_{\max}, z_{\max})$ are large. For example, the reason why we set $y_{\max} = z_{\max} = 24$ in our numerical evaluations is that larger $(y_{\max}, z_{\max})$ would slow down the computation $(\mathsf{lb}^{[2]}_{\text{new}}, \mathsf{ub}^{[2]}_{\text{new}})$ substantially.[5]

It is worth noting that the complexity is on the design stage. For implementation, the complexity is low as one only needs to memorize the waiting time tables found when solving the ACPS-semi-MDP, e.g., for $K = 2$, the waiting time functions $\theta^{[2]}_{\mathsf{a}}(\mathsf{a}_1)$, $\theta^{[2]}_{\mathsf{aa}}(\mathsf{f}_1, \mathsf{a}_2)$, and $\theta^{[2]}_{\mathsf{ini}}(\mathsf{f}_2, \mathsf{a}_3)$ are simple 1D and 2D tables. If desired, one can also use the lower-complexity $K = 1$ scheme, at the cost of slightly larger AoI.

Note that smaller $y_{\max}$ and $z_{\max}$ do not mean that the algorithm can only handle short delays. Instead, it simply says that the algorithm is applicable when the *quantization level is coarse.* For example, say Y_i is exponentially distributed with average service time 25ms. Because only 1% of the delay would be larger than 115.1ms, we can quantize the continuous range of $[0, 115.1\text{ms}]$ by an integer interval $\{0, 1, \cdots, 24\}$ with each integer $j \in \{0, \cdots, 24\}$ represents the delay $Y \approx j \cdot 4.80\text{ms}$. We can then solve our integer-based AoI minimization problem. If we have access to a more powerful computer capable of solving the order-2 achievability semi-MDP problem for $(y_{\max}, z_{\max}) = (50, 50)$, then we quantize $[0, 115.1\text{ms}]$ by an integer interval $j \in \{0, \cdots, 50\}$ satisfying $Y \approx j \cdot 2.30\text{ms}$. The benefits of using larger $(y_{\max}, z_{\max})$ lie in the finer granularity of the network scheduler, not the actual range of the delay it can handle. As in most quantization-based schemes, any AoI suboptimality caused by coarser quantization levels generally diminishes quickly to zero after we increase the levels of quantization.

C. Intuition

The intuition why the performance of our achievability schemes $\mathsf{ub}^{[1]}_{\text{new}}$ and $\mathsf{ub}^{[2]}_{\text{new}}$ is near-optimal is that when backward delay $\mathbb{E}(Z)$ is zero, then obviously an optimal scheduler should wait until Ack_{i-1} has returned, which indicates that no backlog in the forward queue. When the backward delay $\mathbb{E}(Z)$ is small (but non-zero), then waiting for Ack_{i-2} to return before we start to send P_i is reasonable since if we have not received Ack_{i-2} yet, it likely means that P_{i-1} has not been delivered yet since even the previous packet P_{i-2} has not been officially acknowledged. That is why even for the simple $K = 1$ achievability scheme, the best scheme under the condition $S_i \geq A_{i-2}$ has already achieved excellent performance for small $\mathbb{E}(Z)$. The same logic applies that if when deciding to send P_i, we have not even received the acknowledgement A_{i-3}

[4]Our results can be greatly simplified if assuming geometric $\mathbb{P}_Y$ and $\mathbb{P}_Z$. However, we deliberately focus on arbitrary $\mathbb{P}_Y$ and $\mathbb{P}_Z$ so that we can characterize $\mathsf{avg.aoi}^*$ under a general 2-way-queue setting.

[5]We can use larger $(y_{\max}, z_{\max})$ when computing $(\mathsf{lb}^{[1]}_{\text{new}}, \mathsf{ub}^{[1]}_{\text{new}})$ since the computation of the order-1 bounds is fast. However, to ensure fair comparison of $(\mathsf{lb}^{[K]}_{\text{new}}, \mathsf{ub}^{[K]}_{\text{new}})$ between $K = 1$ versus $K = 2$, we limit the range to be $y_{\max} = z_{\max} = 24$.

(two packets before the P_{i-1}), then with probability close-to-one the packet P_{i-1} has not been delivered yet and is still clogging the forward queue. Therefore, we should wait for Ack_{i-3} and only send the current packet P_i after time A_{i-3} *almost always*. That is why the $K = 2$ achievability scheme is almost indistinguishable from the converse lower bound in all cases satisfying $\mathbb{E}(Z) \leq \mathbb{E}(Y)$.

The above intuition also explains why the upper and lower bounds start to diverge significantly *only if* $E(Z) > E(Y)$. In this scenario, the backward queue has a strictly smaller *sustainable throughput* and if we (almost) saturate the forward queue, then the backward queue length will explode, and no acknowledgement packets can return back to s within a reasonable amount of time. The performance of the achievability scheme $\mathsf{ub}_{\text{new}}^{[K]}$ thus suffers greatly. This observation also explains why when $\mathbb{E}(Z) > \mathbb{E}(Y)$, the performance of feedback-based schemes (with a heavily clogged backward queue) is not much better than the best open-loop scheme that periodically sends out P_i while completely ignoring any feedback information $\{\mathsf{Ack}_j : j < i\}$, because all Ack_j packets now experience exceedingly long delay.

D. Future Extension

Because of the analytical and practical importance of characterizing the optimal AoI for the 2-way-queue information update systems and because of the complicated nature of this problem, this work focuses exclusively on the most canonical setting of *linear AoI penalty* with no average energy/cost constraints. At the same time, the proposed framework can be readily extended to include *arbitrary AoI penalty* functions [12] by simply changing the cost function/structure of the proposed semi-MDP problems. Built upon a general ACPS framework, the proposed approaches are likely to be applicable to other age-based metrics, including Age-of-Version (AoV), Age-of-Synchronization (AoS), etc. Our solution can also be used to solve the problem with energy/cost constraints by the well known techniques of Lagrange multipliers [15], [25]. See the references therein.

IX. Conclusion

This work has studied the AoI minimization problem with 2-way queues. Near-optimal schedulers have been devised, which smoothly transition from the instantaneous-ACK schemes to the open-loop schemes depending on how long the feedback delay is. The results have provided a useful road map for other AoI minimization problems with delayed feedback, and can serve as important guidelines when implementing an update-through-queues system in practice.

References

[1] C.-C. Wang, "How useful is delayed feedback in AoI minimization—A study on systems with queues in both forward and backward directions," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Finland, Jun. 2022, pp. 3192–3197.

[2] J. Zhang and C.-C. Wang, "On the rate-cost of Gaussian linear control systems with random communication delays," in *Proc. IEEE Int. Symp. Inform. Theory*, Jun. 2018, pp. 2441–2445.

[3] Y. Sun, Y. Polyanskiy, and E. Uysal, "Sampling of the Wiener process for remote estimation over a channel with random delay," *IEEE Trans. Inf. Theory*, vol. 66, no. 2, pp. 1118–1135, Feb. 2020.

[4] T. Z. Ornee and Y. Sun, "Sampling and remote estimation for the Ornstein–Uhlenbeck process through queues: Age of information and beyond," *IEEE/ACM Trans. Netw.*, vol. 29, no. 5, pp. 1962–1975, Oct. 2021.

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

[6] Y. Zou, K. T. Kim, X. Lin, and M. Chiang, "Minimizing age-of-information in heterogeneous multi-channel systems: A new partial-index approach," in *Proc. 22nd Int. Symp. Theory, Algorithmic Found., Protocol Design Mobile Netw. Mobile Comput.*, Jul. 2021, pp. 11–20, doi: 10.1145/3466772.3467030.

[7] X. Chen, K. Gatsis, H. Hassani, and S. S. Bidokhti, "Age of information in random access channels," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Los Angeles, CA, USA, Jun. 2020, pp. 1770–1775, doi: 10.1109/ISIT44484.2020.9174254.

[8] L. Huang and E. Modiano, "Optimizing age-of-information in a multi-class queueing system," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Hong Kong, Jun. 2015, pp. 1681–1685.

[9] V. Tripathi, R. Talak, and E. Modiano, "Age of information for discrete time queues," 2019, *arXiv:1901.10463*.

[10] 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 Trans. Inf. Theory*, vol. 63, no. 11, pp. 7492–7508, Nov. 2017.

[11] C.-H. Tsai and C.-C. Wang, "Unifying AoI minimization and remote estimation—Optimal sensor/controller coordination with random two-way delay," in *Proc. IEEE INFOCOM Conf. Comput. Commun.*, Jul. 2020, pp. 466–475.

[12] C.-H. Tsai and C.-C. Wang, "Age-of-information revisited: Two-way delay and distribution-oblivious online algorithm," in *Proc. IEEE Int. Symp. Inform. Theory*, Los Angeles, CA, USA, Jun. 2020, pp. 1782–1787.

[13] C.-H. Tsai and C.-C. Wang, "Unifying AoI minimization and remote estimation—Optimal sensor/controller coordination with random two-way delay," *IEEE/ACM Trans. Netw.*, vol. 30, no. 1, pp. 229–242, Feb. 2022, doi: 10.1109/TNET.2021.3111495.

[14] C.-H. Tsai and C.-C. Wang, "Distribution-oblivious online algorithms for age-of-information penalty minimization," *IEEE/ACM Trans. Netw.*, vol. 31, no. 4, pp. 1779–1794, Aug. 2023, doi: 10.1109/TNET.2022.3230009.

[15] C.-H. Tsai and C.-C. Wang, "Jointly minimizing AoI penalty and network cost among coexisting source-destination pairs," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jul. 2021, pp. 3255–3260.

[16] R. D. Yates and S. K. Kaul, "The age of information: Real-time status updating by multiple sources," *IEEE Trans. Inf. Theory*, vol. 65, no. 3, pp. 1807–1827, Mar. 2019.

[17] S. K. Kaul, R. D. Yates, and M. Gruteser, "Status updates through queues," in *Proc. 46th Annu. Conf. Inf. Sci. Syst. (CISS)*, Princeton, NJ, USA, Mar. 2012, pp. 1–6.

[18] C. Kam, S. Kompella, and A. Ephremides, "Learning to sample a signal through an unknown system for minimum AoI," in *Proc. IEEE INFOCOM Conf. Comput. Commun. Workshops*, Apr. 2019, pp. 177–182.

[19] A. M. Bedewy, Y. Sun, and N. B. Shroff, "Minimizing the age of information through queues," *IEEE Trans. Inf. Theory*, vol. 65, no. 8, pp. 5215–5232, Aug. 2019.

[20] A. M. Bedewy, Y. Sun, S. Kompella, and N. B. Shroff, "Optimal sampling and scheduling for timely status updates in multi-source networks," *IEEE Trans. Inf. Theory*, vol. 67, no. 6, pp. 4019–4034, Jun. 2021.

[21] C.-C. Wang, "Technical report of this TNET submission," Tech. Rep., pp. 1–27, doi: 10.1109/TNET.2024.3379895.

[22] J. Pan, A. M. Bedewy, Y. Sun, and N. B. Shroff, "Optimal sampling for data freshness: Unreliable transmissions with random two-way delay," *IEEE/ACM Trans. Netw.*, vol. 31, no. 1, pp. 408–420, Feb. 2023.

[23] D. Bertsekas, *Dynamic Programming and Optimal Control*, 4th ed. Nashua, New Hampshire: Athena Scientific, 2017.

[24] M. Moltafet, M. Leinonen, M. Codreanu, and R. D. Yates, "Status update control and analysis under two-way delay," *IEEE/ACM Trans. Netw.*, vol. 31, no. 6, pp. 2918–2933, Dec. 2023, doi: 10.1109/TNET.2023.3270388.

[25] G. Yao, C.-C. Wang, and N. B. Shroff, "Age minimization with energy and distortion constraints," in *Proc. 24th Int. Symp. Theory, Algorithmic Found., Protocol Design Mobile Netw. Mobile Comput.*, Oct. 2023, pp. 101–110.