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

Chih-Chun Wang, *Fellow, IEEE, Purdue ECE, chihw@purdue.edu*

***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]–[5]. AoI minimization has since attracted significant research attentions on subjects like broadcast channels [6], random access channels [7], etc.

One earliest canonical example of AoI minimization is the *update-through-queues* systems [1], [8]–[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

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

Part of this work was presented in ISIT2022 [1]. This work was supported in parts by NSF Grants CCF-1618475, CCF-1816013, CCF-2008527, CNS-2107363, CCF-2309887, and also by National Spectrum Consortium (NSC) under grant W15QKN-15-9-1004.

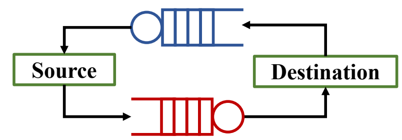


Fig. 1: Information Update System with 2-Way Queues.

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]–[13], [15], [18], [20] s has instantaneous ACK of the departure time D_{i-1}. Optimal $\{S_i : \forall i\}$ 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

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

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:

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

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

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

Best-After-ACK (BAA) [11]–[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

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

We now introduce a useful lemma.

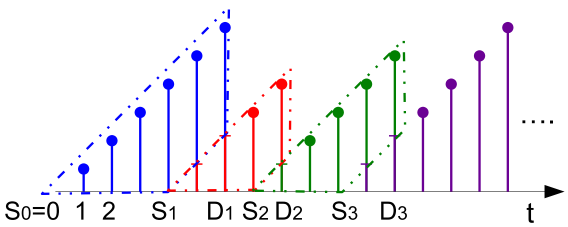


Fig. 2: An alternative way of computing $\sum_{t=1}^{T} \Delta(t)$.

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

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

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

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

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 $\mathsf{a}_1 \in \mathbb{N}^+$:

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

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 (13), we always have $0 \le \mathsf{f}_1 \le \mathsf{a}_2$. 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 $0 \le \mathsf{f}_1 < \mathsf{a}_2$, 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]–[5], [10]–[13], [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 $\mathsf{inst.ack}$ 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.

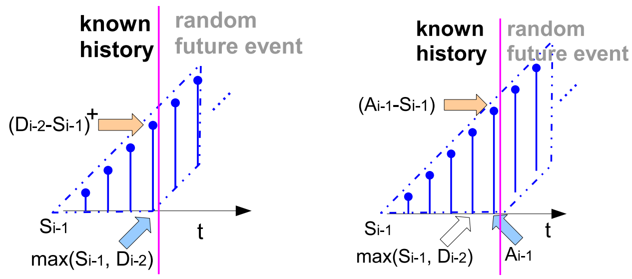
A. The Order-1 Converse

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 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".

[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.

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.

$$\forall \mathsf{a}_1 \in [1, 2y_{\max} + z_{\max}], \text{ we define}$$
$$f^{[1]}_{\mathsf{a}}(\mathsf{a}_1) = \min_{x \in \mathbb{N}^+} \gamma(\mathsf{a}_1 + x, \mathbb{E}\{Y_i\}) - \mathsf{v} \cdot (\mathsf{a}_1 + x) + f^{[1]}_{\mathsf{ini}}(0); \tag{24}$$

$$\forall \mathsf{f}_1 \in [0, y_{\max}], \text{ we hardwire the value } \mathsf{a}_2 = \mathsf{f}_1 \text{ and define}$$
$$f^{[1]}_{\mathsf{ini}}(\mathsf{f}_1) = \min_{x \in \mathbb{N}^+} \Bigg\{ \sum_{k=1}^{x} \mathbb{P}(\tilde{\mathsf{A}}_{i-1} = \mathsf{f}_1 + k) \cdot f^{[1]}_{\mathsf{a}}(\mathsf{f}_1 + k) \tag{25}$$
$$+ \mathbb{P}(\tilde{\mathsf{A}}_{i-1} > \mathsf{f}_1 + x) \cdot \Bigg(\gamma\big(\mathsf{f}_1 + x, m^+_{Y,1}(x)\big) - \mathsf{v} \cdot (\mathsf{f}_1 + x) \tag{26}$$
$$+ \sum_{y=1}^{y_{\max}} \mathbb{P}(Y_{i-1} = y | \tilde{\mathsf{A}}_{i-1} > \mathsf{f}_1 + x) \cdot f^{[1]}_{\mathsf{ini}}\left((y - x)^+\right) \Bigg) \Bigg\} \tag{27}$$

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

$$f^{[1]}_{\mathsf{ini}}(7) = \min_{x \in \mathbb{N}^+} \Bigg\{ \sum_{k=1}^{x} \mathbb{P}(Y_{i-1} + Z_{i-1} = k) \cdot f^{[1]}_{\mathsf{a}}(7 + k)$$
$$+ \mathbb{P}(Y_{i-1} + Z_{i-1} > x) \cdot \Bigg(\gamma\big(7 + x, m^+_{Y,1}(x)\big) - \mathsf{v} \cdot (7 + x)$$
$$+ \sum_{y=1}^{y_{\max}} \mathbb{P}(Y_{i-1} = y | Y_{i-1} + Z_{i-1} > x) \cdot f^{[1]}_{\mathsf{ini}}\left((y - x)^+\right) \Bigg) \Bigg\}$$

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

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

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

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 $\mathsf{lb}^{[1]}_{\mathsf{new}}$ 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

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

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

$$\begin{aligned} \bar{g}^{[1]}_{\mathsf{ini}}(\mathsf{f}, \mathsf{a}) &\triangleq \mathbb{P}(\max(\mathsf{f} + Y_i, \mathsf{a}) + Z_i = \mathsf{a}) \cdot g^{[1]}_{\mathsf{a}}(\mathsf{a}) \\ &+ \mathbb{P}(\max(\mathsf{f} + Y_i, \mathsf{a}) + Z_i > \mathsf{a}) \cdot g^{[1]}_{\mathsf{ini}}(\mathsf{f}, \mathsf{a}) \end{aligned} \tag{45}$$

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

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

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

[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.

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 $A_{i-1} \leq S_i$. On the other hand, because the forward delay $Y_i \geq 1$ with probability one, we also have $A_{i_{\text{nx}}-1} \geq D_{i_{\text{nx}}-1} > S_{i_{\text{nx}}-1}$. 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 $D_{i-1} \leq A_{i-1} \leq S_i$ 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 $\mathsf{a}_2 + x$. 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 $\mathsf{a}_2 + k$ 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 $\mathsf{a}_2 + x$. 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.

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

[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 $\mathbb{P}(Y_{i-1} \geq 1) = 1$, and the conditioning event automatically disappears in (25)–(27). However, in the order-1 achievability scheme, we sometimes have $0 \leq \mathsf{f}_1 < \mathsf{a}_2$ 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).

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 $0 \leq \mathsf{f}_1 \leq \mathsf{a}_2$ because of their definitions in (12) and (13). For any finite or infinite subset

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

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 $g^{[1]}_{\mathsf{ini}}(\mathsf{f}^{[\text{new}]}_1, \mathsf{a}^{[\text{new}]}_2)$ involved in the computation of the right-hand side of (42)–(47) must also satisfy $(\mathsf{f}^{[\text{new}]}_1, \mathsf{a}^{[\text{new}]}_2) \in \Omega$.

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 $A_{i-2} \leq \max(S_{i-1}, D_{i-3})$, 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 $f_{\mathsf{a}}^{[2]}(\cdot,\cdot)$ 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 $f_{\mathsf{a}}^{[2]}(0,0)$ 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.

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

where $f_{\mathsf{ini}}^{[2]}(\cdot,\cdot)$ 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 $f_{\mathsf{aa}}^{[2]}(\mathsf{a}_2 + k)$ is relegated to Appendix D-A 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 $f_{\mathsf{ini}}^{[2]}\left(\mathsf{f}_1 - (\mathsf{a}_2 + x), -x\right)$ 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:

$$\begin{aligned}
&f_{\mathsf{a}}^{[\mathrm{cmb}]}(x_1,x_2) =\\
&\mathbb{P}(\max(x_1 + Y_{i-1}, x_2) + Z_{i-1} = x_2)\cdot f_{\mathsf{aa}}^{[2]}(x_2)\\
&+ \mathbb{P}(\max(x_1 + Y_{i-1}, x_2) + Z_{i-1} > x_2)\cdot f_{\mathsf{a}}^{[2]}(x_1,x_2) \qquad (58)
\end{aligned}$$

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

$$\begin{aligned}
&f_{\mathsf{ini}}^{[\mathrm{cmb}]}(x) =\\
&\quad\mathbb{P}(\max(x + Y_{i-1}, x^{+}) + Z_{i-1} = x^{+})\cdot f_{\mathsf{a}}^{[2]}(x^{+},x^{+})\\
&\quad + \mathbb{P}(\max(x + Y_{i-1}, x^{+}) + Z_{i-1} > x^{+})\cdot f_{\mathsf{ini}}^{[2]}(x,x^{+}) \qquad (59)
\end{aligned}$$

which combines $f_{\mathsf{a}}^{[2]}(\cdot,\cdot)$ and $f_{\mathsf{ini}}^{[2]}(\cdot,\cdot)$, where the type-3 Bellman equations $f_{\mathsf{ini}}^{[2]}(\cdot,\cdot)$ 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

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

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

$$\max(S_{i-1}, D_{i-3}) = S_{i-1} + (\mathsf{a}_3)^{+} \qquad (63)$$

$$\text{and} \quad \mathsf{f}_2 \le \mathsf{a}_3 \le (\mathsf{f}_2)^{+} \qquad (64)$$

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 $f_{\mathsf{a}}^{[\text{cmb}]}(\cdot,\cdot)$ 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 $\gamma\big(\mathsf{a}_3^+ + x, m_{Y,2}^+(x)\big)$ 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 $f_{\mathsf{ini}}^{[\text{cmb}]}\left((\mathsf{f}_2+y)^+ - (\mathsf{a}_3^+ + x)\right)$ represents the next state value function when sending $P_{i_{\text{nx}}}$ with $i_{\text{nx}} = i+1$, the derivation of which is relegated to Appendix F-C of [21].

Even though we have figured out the next state values

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

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

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 $P_{i_{\text{nx}}}$ under state values $(\mathsf{f}_2^{[\text{new}]}, \mathsf{a}_3^{[\text{new}]})$, we may skip Rule G4 and move on to Rule G5 instead if $\mathsf{Ack}_{i_{\text{nx}}-2} = \mathsf{Ack}_{i-1}$ returns back to s before we make the decision at time $\max(S_{i_{\text{nx}}-1}, D_{i_{\text{nx}}-3})$. Also see our discussion of the $f_{\mathsf{a}}^{[\text{cmb}]}(\cdot,\cdot)$ term in (60).

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

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

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

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

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

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

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

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

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

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

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 $x_{\mathsf{ini}}^* \triangleq \theta_{\mathsf{ini}}^{[2]}(\mathsf{f}_2,\mathsf{a}_3)$. 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 $x_{\mathsf{a}}^* \triangleq \theta_{\mathsf{a}}^{[2]}(\mathsf{f}_1,\mathsf{a}_2)$. 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

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

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:

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

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

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

which combines 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

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

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.

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 ($\mathsf{lb}^{[1]}_{\mathrm{new}}$, $\mathsf{ub}^{[1]}_{\mathrm{new}}$) 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^2_Y) (resp. (μ_Z, σ^2_Z)). The truncation intervals are slightly different since we assume $Y_i \ge 1$ and $Z_i \ge 0$ in our setting, see Sec. II. We fix $(\mu_Y, \sigma^2_Y) = (2.5, 0.6^2)$ and set $\sigma^2_Z = 0.6^2$ while varying the value of μ_Z to

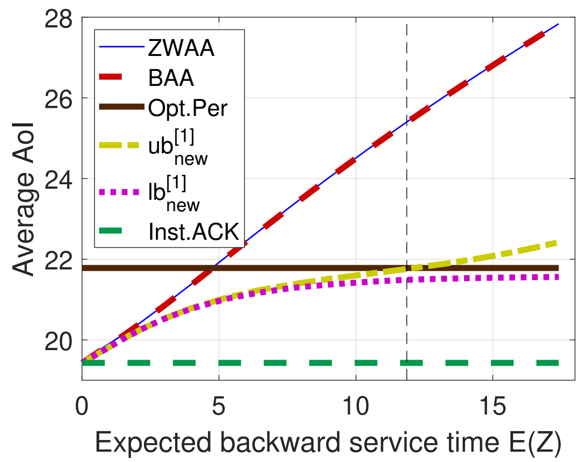


Fig. 4: $(\mathsf{lb}^{[1]}_{\text{new}}, \mathsf{ub}^{[1]}_{\text{new}})$ versus existing results — log-normal $\mathbb{P}_Y$

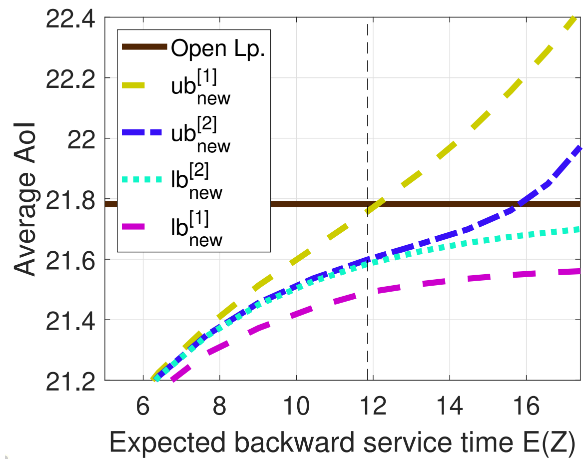


Fig. 6: $(\mathsf{lb}^{[K]}_{\text{new}}, \mathsf{ub}^{[K]}_{\text{new}})$ for $K = 1$ and 2, respectively, — log-normal $\mathbb{P}_Y$

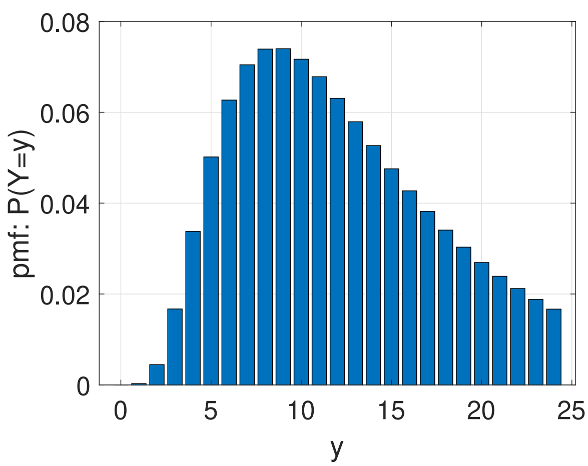


Fig. 5: The pmf of the quantized log-normal distribution of the forward service time Y.

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 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 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 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 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.

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

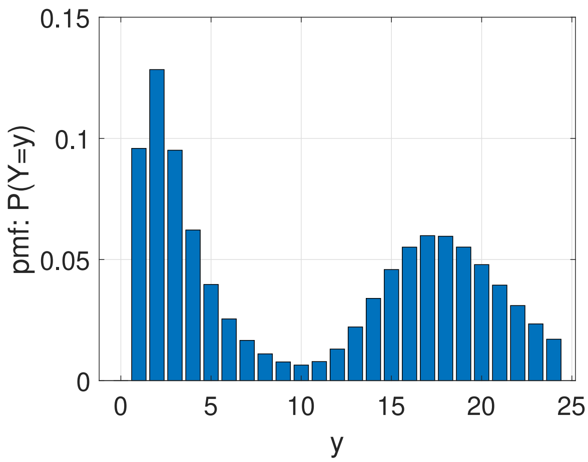


Fig. 7: The pmf of the quantized bimodal composite log-normal distribution of the forward service time Y.

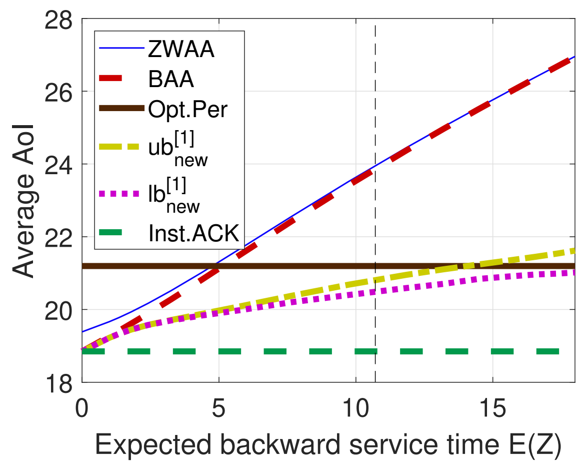


Fig. 8: $(\mathsf{lb}^{[1]}_{\text{new}}, \mathsf{ub}^{[1]}_{\text{new}})$ versus existing results — composite log-normal $\mathbb{P}_Y$.

distribution of $\mathbb{P}_Z$. That is, Z is a simple $[0, 24]$-truncated log-normal with parameters $\sigma_Z^2 = 0.6^2$ and we vary the values 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}.

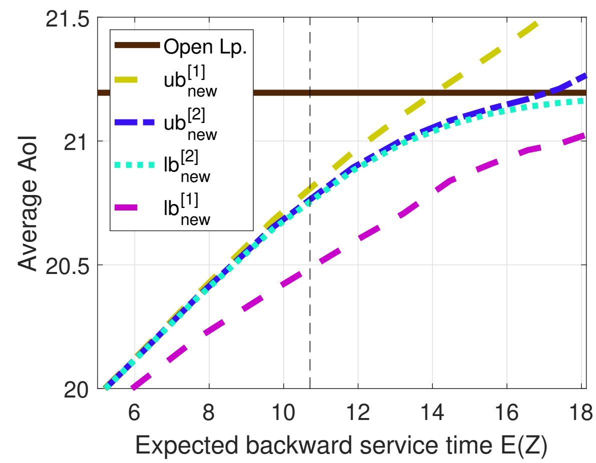


Fig. 9: $(\mathsf{lb}^{[K]}_{\text{new}}, \mathsf{ub}^{[K]}_{\text{new}})$ for $K = 1$ and 2, respectively, — composite log-normal $\mathbb{P}_Y$

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 decision 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]–[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], [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.[4] 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

[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 $y_{\max} = z_{\max} = 24$.

send P_i, we have not even received the acknowledgement A_{i-3} (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’l Symp. Inform. Theory*. Espoo, Finland, June 2022, 6 pages.

[2] J. Zhang and C.-C. Wang, “On the rate-cost of Gaussian linear control systems with random communication delays,” in *Proc. IEEE Int’l Symp. Inform. Theory*. Vail, USA, June 2018.

[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, February 2019.

[4] T. 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, October 2021.

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

[6] Y. Zou, K. Kim, X. Lin, and M. Chiang, “Minimizing age-of-information in heterogeneous multi-channel systems: A new partial-index approach,” in *Proc. ACM Int’l Symp. on Theory, Algorithmic Foundation, Protocol Design for Mobile Networks and Mobile Computing (MobiHoc)*, July 2021, https://doi.org/10.1145/3466772.3467030.

[7] X. Chen, K. Gatsis, and H. H. amd S.S. Bidokhti, “Age of information in random access channels,” in *Proc. IEEE Int’l Symp. Inform. Theory*. Los Angeles, California, June 2020, https://doi.org/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’l Symp. Inform. Theory*. Hong Kong, China, June 2015.

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

[10] Y. Sun, E. Uysal-Biyikoglu, R. Yates, C. Koksal, and N. Shroff, “Update or wait: How to keep your data fresh,” *IEEE Trans. Inf. Theory*, vol. 63, no. 11, pp. 7492–7508, December 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. 39th IEEE Conference on Computer Communications (INFOCOM)*. Toronto, Canada, July 2020, 10 pages.

[12] ——, “Age-of-information revisited: Two-way delay and distribution-oblivious online algorithm,” in *Proc. IEEE Int’l Symp. Inform. Theory*. Los Angeles, USA, June 2020, 6 pages.

[13] ——, “Unifying AoI minimization and remote estimation – optimal sensor/controller coordination with random two-way delay,” *IEEE/ACM Trans. Netw.*, vol. 30, no. 1, February 2022, https://doi.org/10.1109/TNET.2021.3111495.

[14] ——, “Distribution-oblivious online algorithms for age-of-information penalty minimization,” *IEEE/ACM Trans. Netw.*, 2023, https://doi.org/10.1109/TNET.2022.3230009.

[15] ——, “Jointly minimizing AoI penalty and network cost among coexisting source-destination pairs,” in *Proc. IEEE Int’l Symp. Inform. Theory*. Melbourne, Australia, July 2021, 6 pages.

[16] R. Yates and S. Kaul, “The age of information: Real-time status updating by multiple sources,” *IEEE Trans. Inf. Theory*, vol. 65, no. 3, pp. 1807–1826, March 2019.

[17] S. Kaul, R. Yates, and M. Gruteser, “Status updates through queues,” in *Proc. 46th Conf. Inform. Sciences and Systems*. Princeton, NJ, USA, March 2012.

[18] C. Kam, S. Kompella, and A. Ephremides, “Learning to sample a signal through an unknown system for minimum AoI,” in *Proc. IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2019.

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

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

[21] C.-C. Wang, “Technical report of this TNET submission,” included as a supplementary material.

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

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

[24] M. Moltafet, M. Leinonen, M. Codreanu, and R. Yates, “Status update control and analysis under two-way delay,” *IEEE/ACM Trans. Netw.*, eary Access with DOI: 10.1109/TNET.2023.3270388.

[25] G. Yao, C.-C. Wang, and N. Shroff, “Age minimization with energy and distortion constraints,” in *Proc. ACM Int’l Symp. on Theory, Algorithmic Foundation, Protocol Design for Mobile Networks and Mobile Computing (MobiHoc)*, October 2023.

APPENDIX A
PROOFS OF LEMMAS 2 AND 5

A. Proof of Lemma 2

To show that s is capable of following Rule G1, we consider the view point of s at time S_{i-1}. There are two possible scenarios: Scenario 1: $A_{i-K-1} \leq S_{i-1}$. In this scenario, s must know the value of D_{i-K-1} at time S_{i-1} via the delivery of Ack_{i-K-1}. This implies that s is able to compute the "must-wait" period $[S_{i-1}, \max(S_{i-1}, D_{i-K-1}))$ at time S_{i-1}, and can thus follow Rule G1 if it desires.

Scenario 2: s has not received Ack_{i-K-1} by time S_{i-1}, i.e., $A_{i-K-1} > S_{i-1}$. In this scenario, s knows that we must have $D_{i-K-1} > S_{i-1}$, otherwise our special order-K genie would have already delivered Ack_{i-K-1} by time $\max(S_{i-1}, D_{i-K-1}) = S_{i-1}$. More importantly, in this scenario, *the very moment that source s receives* Ack_{i-K-1} *must also be the forward delivery time* $t = D_{i-K-1}$*, i.e., the time when the genie actually takes over.* Therefore, in this scenario, source s, if it desires, can keep waiting until it has received Ack_{i-K-1}, delivered by the order-K genie at time $\max(S_{i-1}, D_{i-K-1})$, which essentially means that s is capable of following Rule G1.

B. Proof of Lemma 5

By the definitions (12) and (13), we always have $0 \leq \mathsf{f}_1 \leq \mathsf{a}_2$. Because of the presence of the order-1 genie, we always have $A_{i-2} \leq \max(S_{i-1}, D_{i-2})$, which, when combined with (12) and (13), implies $\mathsf{a}_2 \leq \mathsf{f}_1$. The proof is complete. The intuition is that the order-1 genie will expedite the delivery of Ack_{i-2} at time $\max(S_{i-1}, D_{i-2}) = S_{i-1} + \mathsf{f}_1$, see (12). Therefore, the relative acknowledgement time a_2 cannot be larger than the relative forward queue clearance time f_1.

C. Proof of Lemma 6

It is obvious that $(\mathsf{f}_1, \mathsf{a}_2) = (0, 0)$ belongs to the Ω_{sc} defined in (54).

Suppose $(\mathsf{f}_1, \mathsf{a}_2)$ satisfies (54). By (50), we have

$$0 \leq \mathsf{f}_1^{[\text{new}]} \leq (\mathsf{f}_1 + y_{\max} - (\mathsf{a}_2 + 0))^+ \leq y_{\max} \tag{82}$$

since $\mathsf{a}_2 - \mathsf{f}_1 \geq 0$. By (51), we have

$$\mathsf{a}_2^{[\text{new}]} - \mathsf{f}_1^{[\text{new}]} \tag{83}$$
$$\leq \max\big(- \mathsf{f}_1^{[\text{new}]}, \tag{84}$$
$$\mathsf{f}_1 + y + z - (\mathsf{a}_2 + x) - \mathsf{f}_1^{[\text{new}]}, \tag{85}$$
$$\mathsf{a}_2 + z - (\mathsf{a}_2 + x) - \mathsf{f}_1^{[\text{new}]}\big) \tag{86}$$

The term in (84) is clearly no larger than 0, which is no larger than $z_{\max}$. By (50), the term in (85) is no larger than z since $x \geq 0$ and $\mathsf{f}_1^{[\text{new}]} \geq 0$, which is no larger than $z_{\max}$. The term in (86) is no larger than z, which is no larger than $z_{\max}$. As a result, we must have $\mathsf{a}_2^{[\text{new}]} - \mathsf{f}_1^{[\text{new}]} \leq z_{\max}$.

Since $(\mathsf{f}_1^{[\text{new}]}, \mathsf{a}_2^{[\text{new}]})$ are originally derived using the formulas in (12) and (13), respectively, we also have $\mathsf{a}_2^{[\text{new}]} - \mathsf{f}_1^{[\text{new}]} \geq 0$. The proof of Lemma 6 is complete.

APPENDIX B
PROOF OF LEMMA 7

At time $\max(S_{i-1}, D_{i-3})$, source s is still waiting for Ack_{i-2} and Ack_{i-1} (though it is possible that Ack_{i-2} has already returned to s at that time). Since new information arrives only when either Ack_{i-2} or Ack_{i-1} returns, s can anticipate the situation and pre-compute the decision S_i at time $t = \max(S_{i-1}, D_{i-3})$, $\max(S_{i-1}, A_{i-2})$, and A_{i-1}, depending on whether any of Ack_{i-2} and Ack_{i-1} has returned or not, which corresponds to Rules G4, G5, and G6, respectively. In the sequel, we discuss each of these three rules separately.

A. Rule G4

We first consider the "state" faced by source s at time $\max(S_{i-1}, D_{i-3}) = S_{i-1} + \mathsf{a}_3^+$, where the equality follows from (63) in Lemma 8. We then note that the state of Markov decision process must fully capture (a) the distribution of the randomness it faces, and (b) the cost it faces if a decision is made at that particular state.

We first consider (b), the cost function s faces at time $\max(S_{i-1}, D_{i-3}) = S_{i-1} + \mathsf{a}_3^+$. At that time, the AoI cost has grown to

$$\max(S_{i-1}, D_{i-3}) - S_{i-1} = \mathsf{a}_3^+, \tag{87}$$

which thus needs to be included as part of the state.

We now consider (a) the distribution of the randomness s faces at time $\max(S_{i-1}, D_{i-3})$. We first note that packet P_{i-2} has started to be processed by the forward queue at time $\max(D_{i-3}, S_{i-2}) = S_{i-1} + \mathsf{f}_2$, where the equality follows from (18). As a result, the term

$$\max(S_{i-1}, D_{i-3}) - \max(D_{i-3}, S_{i-2}) = \mathsf{a}_3^+ - \mathsf{f}_2 \tag{88}$$

represents how many time slots have passed since P_{i-2} was first processed by the forward queue until the current time $S_{i-1} + \mathsf{a}_3^+$, the time instant of making the scheduling decision. Out of these many time slots, there are

$$(A_{i-3} - \max(D_{i-3}, S_{i-2}))^+ = \mathsf{a}_3 - \mathsf{f}_2 \tag{89}$$

time slots during which the Ack_{i-3} are still in the backward queue and thus could potentially block the return of Ack_{i-2}. The equality in (89) follows from the definitions of a_3 and f_2 in (19) and (18), respectively.

Both pieces of information (i.e., (88) and (89)) will help the source accurately assess the randomness it faces. For example, the larger the value of (88), the more likely that source s can receive Ack_{i-2} soon since P_{i-2} has been processed for a while, compared to the decision time $\max(S_{i-1}, D_{i-3})$. On the other hand, the larger the value of (89), the less likely that source s can receive Ack_{i-2} soon since the backward queue has been blocked/clogged by Ack_{i-3} for a long time, which, in turns, blocks Ack_{i-2} since the backward queue was busy serving Ack_{i-3} and could not serve Ack_{i-2} during those time slots.

We now argue that (88) and (89) jointly capture all the randomness of the system. The reason is as follows. At time $\max(S_{i-1}, D_{i-3})$, the order-2 genie has already delivered Ack_{i-3} back to s. Therefore, the only randomness remaining in the system is the *delivery times* of P_{i-2} and P_{i-1} and the *return times* of Ack_{i-2} and Ack_{i-1}. The delivery time D_{i-2} only depends on when the P_{i-2} got processed at the first place, which is fully reflected by (88). The return time A_{i-2} depends both on D_{i-2} and on when the backward queue is available to process Ack_{i-2}, which is fully captured by the pair (88) and (89). Since the forward packet P_{i-1} is stuck behind P_{i-2} and the backward packet Ack_{i-1} is stuck behind Ack_{i-2}, knowing the joint distribution of (D_{i-2}, A_{i-2}) also gives us the joint distribution of D_{i-1} and A_{i-1}. Collectively, it shows that (88) and (89) collectively capture the randomness faced by s at time $\max(S_{i-1}, D_{i-3})$.

The above analysis proves that the necessary state value can be captured by the tuple of $(\mathsf{a}_3^+, \mathsf{a}_3^+ - \mathsf{f}_2, \mathsf{a}_3 - \mathsf{f}_2)$ due to the discussion in (87)–(89). We can easily simplify this 3D state-value tuple to the pair $(\mathsf{f}_2, \mathsf{a}_3)$, which is why we can choose the waiting time of Rule G4 by $x^*_{\mathsf{ini}} = \phi^{[2]}_{\mathsf{ini}}(\mathsf{f}_2, \mathsf{a}_3)$.

B. Rule G5

We now consider the "state" faced by source s at time $\max(A_{i-2}, S_{i-1}) = S_{i-1} + \mathsf{a}_2$, where the equality follows from (13). At that time, we know that either P_{i-1} is still in the forward queue or its acknowledgement Ack_{i-1} is still in the backward queue. Similar to the arguments around (88) and (89) in the previous sub-section, the randomness faced by s is determined by

$$\max(A_{i-2}, S_{i-1}) - \max(D_{i-2}, S_{i-1}) = \mathsf{a}_2 - \mathsf{f}_1 \quad (90)$$

which represents the number of time slots for which P_{i-1} has begun to be processed at the current moment $\max(A_{i-2}, S_{i-1})$. The role of (90) is similar to (88). Herein, the equality of (90) follows from (13) and (12).

Out of these many time slots, there are

$$(A_{i-2} - \max(D_{i-2}, S_{i-1}))^+ = \mathsf{a}_2 - \mathsf{f}_1 \quad (91)$$

time slots during which Ack_{i-2} is still in the backward queue and thus could potentially block the return of Ack_{i-1}. The reason is that the time instant $\max(D_{i-2}, S_{i-1})$ is the start of the interval $(\max(D_{i-2}, S_{i-1}, \max(A_{i-2}, S_{i-1})]$ in which we are currently interested. A_{i-2} is the time instant when Ack_{i-2} is out of the backward queue. The difference, when projected to a non-negative integer, is the number of time slots that Ack_{i-2} is still in the backward queue during the time interval $(\max(D_{i-2}, S_{i-1}, \max(A_{i-2}, S_{i-1})]$. The role of (91) is similar to (89). Herein, the equality of (91) follows from (13) and (12). Based on the above discussion, (90) and (91) jointly capture (a) the randomness faced by s at time $\max(S_{i-1}, A_{i-2})$.

Note that

$$\max(A_{i-2}, S_{i-1}) - S_{i-1} = \mathsf{a}_2 \quad (92)$$

represents how much AoI has grown until the current decision time $\max(A_{i-2}, S_{i-1})$. The role of (92) is similar to (87).

The above arguments in (90)–(92) show that the pair $(\mathsf{a}_2 - \mathsf{f}_1, \mathsf{a}_2)$ is sufficient for the system state. That is why we can choose the waiting time of Rule G5 by $x^*_{\mathsf{a}} = \phi^{[2]}_{\mathsf{a}}(\mathsf{f}_1, \mathsf{a}_2)$.

C. Rule G6

We now consider the "state" faced by source s at time A_{i-1}. At that time, we know that there is no packet in either the forward or the backward queue. The only "state" faced by s is how much AoI has grown until the decision time A_{i-1}, which is $A_{i-1} - S_{i-1} = \mathsf{a}_1$ as defined in (11). That is why we can choose the waiting time of Rule G6 by $x^*_{\mathsf{aa}} = \phi^{[2]}_{\mathsf{aa}}(\mathsf{a}_1)$.

APPENDIX C
THE DERIVATION OF $f^{[2]}_{\mathsf{a}}(0,0)$ IN (55)

In Rule G6, we send P_i after A_{i-1}. Therefore, for the next packet index $i_{\text{nx}} = i + 1$, the time instant of making the G6-decision must be $\max(S_{i_{\text{nx}}-1}, D_{i_{\text{nx}}-3}) = S_i = S_{i_{\text{nx}}-1}$. This also means that $A_{i_{\text{nx}}-2} \leq S_{i_{\text{nx}}-1}$ since $A_{i-1} \leq S_i$. However, because the forward delay $Y_i \geq 1$ with probability one, we also have $A_{i_{\text{nx}}-1} > S_{i_{\text{nx}}-1}$. Jointly, it means that at the time instant of deciding the send time of the next packet $P_{i_{\text{nx}}}$, $\mathsf{Ack}_{i_{\text{nx}}-2}$ has returned to s but $\mathsf{Ack}_{i_{\text{nx}}-1}$ has not. Therefore, the scheme will skip Rule G4 and apply Rule G5, and definitely without using Rule G6. Note that in this scenario, at the decision time $S_{i_{\text{nx}}-1}$, the new state values become

$$\mathsf{f}_1^{[\text{new}]} = (D_{i_{\text{nx}}-2} - S_{i_{\text{nx}}-1})^+ = (D_{i-1} - S_i)^+ = 0 \quad (93)$$

$$\mathsf{a}_2^{[\text{new}]} = (A_{i_{\text{nx}}-2} - S_{i_{\text{nx}}-1})^+ = (A_{i-1} - S_i)^+ = 0 \quad (94)$$

where (93) follows from (12); and (94) follows from (13). Therefore, it will enter the next state $f^{[2]}_{\mathsf{a}}(0,0)$, where the subscript "a" signifies that only one acknowledgement packet has returned back to s (i.e., the value function under Rule G5).

APPENDIX D
THE NEXT STATE VALUES IN (56)–(57)

A. The Derivation of $f^{[2]}_{\mathsf{aa}}(\mathsf{a}_2 + k)$ in (56)

Specifically, the event $\{\tilde{\mathsf{A}}_{i-1} = \mathsf{a}_2 + k\}$ is equivalent to the event $\{A_{i-1} = S_{i-1} + \mathsf{a}_2 + k\}$. When this event happens, we will switch to Rule G6 and the new state value becomes $\mathsf{a}_1 = A_{i-1} - S_{i-1} = \mathsf{a}_2 + k$. That is why the next state value in (56) is set to $f^{[2]}_{\mathsf{aa}}(\mathsf{a}_2 + k)$.

B. The Derivation of $f^{[2]}_{\mathsf{ini}}(\mathsf{f}_1 - (\mathsf{a}_2 + x), -x)$ in (57)

In Rule G5, we send P_i after the return of Ack_{i-2} but before the return of Ack_{i-1}. Therefore, for the next packet index $i_{\text{nx}} = i + 1$, the time instant of making the decision must be $\max(S_{i_{\text{nx}}-1}, D_{i_{\text{nx}}-3}) = S_{i_{\text{nx}}-1} = S_i \geq A_{i-2} \geq D_{i-2} = D_{i_{\text{nx}}-3}$. By the same observation, at the next decision time $S_{i_{\text{nx}}-1}$, the feedback packet $\mathsf{Ack}_{i_{\text{nx}}-2}$ has not returned to s yet since $A_{i-1} > S_i$. Jointly it means that we must be activating Rule G4 when we try to decide the send time of $P_{i_{\text{nx}}}$. That is why we use the type-3 Bellman equations $f^{[2]}_{\mathsf{ini}}(\cdot,\cdot)$ that is tailored for Rule G4.

We now analyze the next state value $(\mathsf{f}_2^{[\text{new}]}, \mathsf{a}_3^{[\text{new}]})$. Specifically, we have

$$\mathsf{f}_2^{[\text{new}]} = \max(S_{i_{\text{nx}}-2}, D_{i_{\text{nx}}-3}) - S_{i_{\text{nx}}-1} \quad (95)$$
$$= \max(S_{i-1}, D_{i-2}) - S_i \quad (96)$$
$$= (S_{i-1} + \mathsf{f}_1) - (S_{i-1} + \mathsf{a}_2 + x) \quad (97)$$
$$= \mathsf{f}_1 - (\mathsf{a}_2 + x)$$
$$\mathsf{a}_3^{[\text{new}]} = \max(A_{i_{\text{nx}}-3}, S_{i_{\text{nx}}-2}) - S_{i_{\text{nx}}-1} \quad (98)$$
$$= \max(A_{i-2}, S_{i-1}) - S_i \quad (99)$$
$$= (S_{i-1} + \mathsf{a}_2) - (S_{i-1} + \mathsf{a}_2 + x) = -x \quad (100)$$

where (95) follows from (18); (96) follows from $i_{\text{nx}} = i+1$; (97) follows from (12) and our target send time is $S_i = S_{i-1} + \mathsf{a}_2 + x$; (98) follows from (20); (99) follows from $i_{\text{nx}} = i+1$; (100) follows from (13). The above argument shows that it will enter the next state $f_{\mathsf{ini}}^{[2]}(\mathsf{f}_1 - (\mathsf{a}_2 + x), -x)$.

Appendix E
The Proof of Lemma 8

We first discuss the intuition of Lemma 8. Recall that the time instant of making a Rule-G4 decision is at time $t = \max(S_{i-1}, D_{i-3})$. The equality (63) of Lemma 8 implies that the relative time of making the decision, w.r.t. S_{i-1}, is exactly a_3^+. The inequality $\mathsf{f}_2 \leq \mathsf{a}_3$ is by the max operation in (19). When combining the last inequality (64) of Lemma 8 with the simple inequality $\mathsf{f}_2 \leq \mathsf{a}_3$, Lemma 8 essentially proves that if $\mathsf{f}_2 \geq 0$, then $\mathsf{a}_3 = \mathsf{f}_2$. The intuition is that if P_{i-2} has not started to be processed by the forward queue at time S_{i-1} (thus $\mathsf{f}_2 \geq 0$), then when it is finally being processed (the very moment in which P_{i-3} has been delivered), the order-2 genie will help deliver Ack_{i-3} back to s immediately, which leads to the equality $\mathsf{a}_3 = \mathsf{f}_2$.

In the sequel, we provide the detailed proofs. Because $\mathsf{f}_2 \leq \mathsf{a}_3$ by the definition in (19), we only need to prove (63) and the following inequality:

$$\mathsf{a}_3 \leq (\mathsf{f}_2)^+. \quad (101)$$

We consider two cases. Case 1: $D_{i-3} > S_{i-1}$. In this case, the decision time is

$$\max(S_{i-1}, D_{i-3}) = S_{i-1} + (D_{i-3} - S_{i-1}). \quad (102)$$

Because the order-2 genie will deliver Ack_{i-3} instantaneously at time $\max(D_{i-3}, S_{i-1}) = D_{i-3}$, we have $A_{i-3} = D_{i-3}$, which, by (20), implies $\mathsf{a}_3 = D_{i-3} - S_{i-1} > 0$. Eq. (102) then implies (63).

Additionally, in Case 1, by (18) we have $\mathsf{f}_2 = D_{i-3} - S_{i-1}$. Eq. (101) thus holds as well.

Case 2: $D_{i-3} \leq S_{i-1}$. In this case, we have

$$\max(S_{i-1}, D_{i-3}) = S_{i-1} + 0. \quad (103)$$

Because the order-2 genie will deliver Ack_{i-3} instantaneously at time $\max(D_{i-3}, S_{i-1}) = S_{i-1}$ if Ack_{i-3} has not returned back to s yet, we have $A_{i-3} \leq S_{i-1}$. (20) then implies $\mathsf{a}_3 \leq 0$. Eq. (103) then implies (63).

Additionally, in Case 2, we have $\mathsf{f}_2 \leq 0$ by (18). Eq. (101) thus holds as well.

The proof of Lemma 8 is complete.

Appendix F
Various Terms in (60)–(62)

A. The Derivation of $f_{\mathsf{a}}^{[\text{cmb}]}\left((\mathsf{f}_2 + y)^+, \mathsf{a}_3^+ + k\right)$ in (60)

Consider the scenario in which Ack_{i-2} returns back to s before the tentative decision $S_{i-1} + \mathsf{a}_3^+ + x$, i.e., the event $\{\tilde{\mathsf{A}}_{i-2} = \mathsf{a}_3^+ + k\}$ for some $k \leq x$. When this event happens, we will switch to Rule G5 and the new state values become

$$\mathsf{f}_1 = (D_{i-2} - S_{i-1})^+ \quad (104)$$
$$= ((\max(S_{i-2}, D_{i-3}) + Y_{i-2}) - S_{i-1})^+ \quad (105)$$
$$= (\mathsf{f}_2 + Y_{i-2})^+ \quad (106)$$
$$\mathsf{a}_2 = (A_{i-2} - S_{i-1})^+ \quad (107)$$
$$= (\mathsf{a}_3^+ + k)^+ = \mathsf{a}_3^+ + k \quad (108)$$

where (104) follows from the definition of f_1 in (12); (105) follows from the queue evolution equation (2); (106) follows from the definition of f_2 in (18); (107) follows from the definition of a_2 in (13); (108) from that this discussion is under the event $\{A_{i-2} = S_{i-1} + \mathsf{a}_3^+ + k\}$ with $k \geq 0$. Jointly, it implies that the next state value in (60) is set to $f_{\mathsf{a}}^{[\text{cmb}]}\left((\mathsf{f}_2 + y)^+, \mathsf{a}_3^+ + k\right)$.

It is worth emphasizing that by (58), the term $f_{\mathsf{a}}^{[\text{cmb}]}\left((\mathsf{f}_2 + y)^+, \mathsf{a}_3^+ + k\right)$ is actually a summation of two sub-terms

$$\mathbb{P}\left(\max\left((\mathsf{f}_2 + y)^+ + Y_{i-1}, \mathsf{a}_3^+ + k\right) + Z_{i-1} > \mathsf{a}_3^+ + k\right) \cdot f_{\mathsf{a}}^{[2]}\left((\mathsf{f}_2 + y)^+, \mathsf{a}_3^+ + k\right) \quad (109)$$

and

$$\mathbb{P}\left(\max\left((\mathsf{f}_2 + y)^+ + Y_{i-1}, \mathsf{a}_3^+ + k\right) + Z_{i-1} = \mathsf{a}_3^+ + k\right) \cdot f_{\mathsf{aa}}^{[2]}\left(\mathsf{a}_3^+ + k\right). \quad (110)$$

To explain these two sub-terms, we first notice that

$$A_{i-1} - S_{i-1} = \max\left((\mathsf{f}_2 + y)^+ + Y_{i-1}, \mathsf{a}_3^+ + k\right) + Z_{i-1} \quad (111)$$

where $(\mathsf{f}_2 + y) = D_{i-2}$ is when P_{i-2} was delivered; $(\mathsf{f}_2 + y)^+$ is when P_{i-1} started to be processed by the forward queue; $(\mathsf{f}_2 + y)^+ + Y_{i-1} = D_{i-1}$ is when P_{i-1} was delivered; and $\max\left((\mathsf{f}_2 + y)^+ + Y_{i-1}, \mathsf{a}_3^+ + k\right)$ is when the backward queue started to process Ack_{i-1}. Using (111), the probabilistic weight in (109) quantifies the probability of the event

$$\left\{A_{i-1} > A_{i-2} = S_{i-1} + \mathsf{a}_3^+ + k\right\}. \quad (112)$$

Under this event, source s will apply Rule G5 for packet P_i even though the original purpose of computing the Bellman equation $f_{\mathsf{ini}}^{[2]}(\mathsf{f}_2, \mathsf{a}_3)$ in (60)–(62) is to decide the best Rule-G4 decision. This is why we multiply the $f_{\mathsf{a}}^{[2]}(\cdot, \cdot)$ term in (109).

For (110), the probabilistic weight quantifies the probability of the event

$$\left\{A_{i-1} = A_{i-2} = S_{i-1} + \mathsf{a}_3^+ + k\right\}. \quad (113)$$

Under this event, two acknowledgement packets Ack_{i-2} and Ack_{i-1} return back to s at the same time instant. Therefore, s will skip Rule G5 and immediately switch to Rule G6 for packet P_i even though the original purpose of computing the Bellman equation $f_{\mathsf{ini}}^{[2]}(\mathsf{f}_2, \mathsf{a}_3)$ in (60)–(62) is to decide the best

Rule-G4 decision. (Essentially, we skip Rule G4 to consider Rule G5, and then immediately skip Rule G5 and apply Rule G6 instead.) This is why we multiply the $f^{[2]}_{\mathsf{aa}}(\cdot)$ term in (110). When applying Rule G6, we need to know the state value a_1, which, by definition, is $A_{i-1} - S_{i-1} = A_{i-2} - S_{i-1} = \mathsf{a}_3^+ + k$ in this event. That is why we have the $f^{[2]}_{\mathsf{aa}}\left(\mathsf{a}_3^+ + k\right)$ term in (110).

B. The Derivation of $m^+_{Y,2}(x)$ in (22)

Consider the conditional expectation expression in (22). The $(\mathsf{f}_2 + Y_{i-2})^+$ term in (22) is when P_{i-1} will be first processed by the forward queue, i.e., after S_{i-1} and after P_{i-2} has cleared the forward queue. Adding the processing time Y_{i-1} will give us the delivery time D_{i-1} of P_{i-1}. If the delivery time D_{i-1} of P_{i-1} is later than the scheduled transmission time $(\mathsf{a}_3^+ + x)$, the packet P_{i-1} will start to block the processing of P_i. Eq. (22) describes that if we eventually commit to the send time $(\mathsf{a}_3^+ + x)$, thus the conditioning event $\tilde{\mathsf{A}}_{i-2} > \mathsf{a}_3^+ + x$, how much average delay of P_i we should expect once we include the service delay Y_i *plus the queueing delay caused by* P_{i-1} *and* P_{i-2}.

C. The Derivation of $f^{[\mathrm{cmb}]}_{\mathsf{ini}}\left((\mathsf{f}_2 + y)^+ - (\mathsf{a}_3^+ + x)\right)$ in (62)

We analyze the state values $(\mathsf{f}_2^{[\mathrm{new}]}, \mathsf{a}_3^{[\mathrm{new}]})$ under Rule G4 for the next packet index $i_{\mathrm{nx}} = i + 1$. That is,

$$\mathsf{f}_2^{[\mathrm{new}]} = \max(S_{i-1}, D_{i-2}) - S_i \tag{114}$$

$$= (S_{i-1} + (D_{i-2} - S_{i-1})^+) - (S_{i-1} + \mathsf{a}_3^+ + x) \tag{115}$$

$$= (\mathsf{f}_2 + Y_{i-2})^+ - (\mathsf{a}_3^+ + x) \tag{116}$$

$$\mathsf{a}_3^{[\mathrm{new}]} = \max(A_{i-2}, S_{i-1}) - S_i \tag{117}$$

where (114) follows from (18) and $i_{\mathrm{nx}} = i + 1$; (115) follows from basic simplification and that our target send time is $S_i = S_{i-1} + \mathsf{a}_3^+ + x$; (116) follows from rewriting D_{i-2} using the queue evolution equation (2) and by the definition of f_2 in (18); (117) follows from (20) and $i_{\mathrm{nx}} = i + 1$. We now argue that $(\mathsf{f}_2^{[\mathrm{new}]}, \mathsf{a}_3^{[\mathrm{new}]})$ always satisfy

$$\mathsf{a}_3^{[\mathrm{new}]} = \left(\mathsf{f}_2^{[\mathrm{new}]}\right)^+ \tag{118}$$

To prove (118), we first prove that $\mathsf{a}_3^{[\mathrm{new}]} \geq 0$. The reason is that under Rule G4, we send the packet P_i before the return of Ack_{i-2}. Therefore, we always have $A_{i-2} \geq S_i$. If we dive deeper, technically, we should always have $A_{i-2} > S_i$, otherwise we will switch to Rule G5 before committing to the send time S_i. However, after we send P_i and move on to the next packet $P_{i_{\mathrm{nx}}}$, we could sometimes have $\max(S_{i_{\mathrm{nx}}-1}, D_{i_{\mathrm{nx}}-3}) = \max(S_i, D_{i-2}) = S_i$. If that happens, at time $S_{i_{\mathrm{nx}}-1}$, the order-2 genie will immediately deliver $\mathsf{Ack}_{i_{\mathrm{nx}}-3} = \mathsf{Ack}_{i-2}$ back to s. This will force $A_{i-2} = S_i$ (after we commit the send time S_i) even though before we commit the send time, we have not received Ack_{i-2} yet. The fact that $A_{i-2} \geq S_i$ when considering packet $P_{i_{\mathrm{nx}}}$, implies that $0 \leq \mathsf{a}_3^{[\mathrm{new}]}$ because of (117).

By Lemma 8, we also have $\mathsf{f}_2^{[\mathrm{new}]} \leq \mathsf{a}_3^{[\mathrm{new}]} \leq \left(\mathsf{f}_2^{[\mathrm{new}]}\right)^+$. If $\mathsf{f}_2^{[\mathrm{new}]} \geq 0$, then (118) holds obviously. On the other hand, if $\mathsf{f}_2^{[\mathrm{new}]} < 0$, then the arguments in the previous paragraph establishes that $0 \leq \mathsf{a}_3^{[\mathrm{new}]} \leq 0$. Therefore, (118) again holds. We have thus proven (118) for both cases. Jointly, (117) and (118) imply that the next state value in (62) can be set to $f^{[\mathrm{cmb}]}_{\mathsf{ini}}\left((\mathsf{f}_2 + y)^+ - (\mathsf{a}_3^+ + x)\right)$.

Specifically, by (59), the term $f^{[\mathrm{cmb}]}_{\mathsf{ini}}\left((\mathsf{f}_2 + y)^+ - (\mathsf{a}_3^+ + x)\right)$ is actually a summation of two sub-terms

$$\mathbb{P}\Big(\max\left(\mathsf{f}_2^{[\mathrm{new}]} + Y_{i-1}, \mathsf{a}_3^{[\mathrm{new}]}\right) + Z_{i-1} > \mathsf{a}_3^{[\mathrm{new}]}\Big) \cdot f^{[2]}_{\mathsf{ini}}\left(\mathsf{f}_2^{[\mathrm{new}]}, \mathsf{a}_3^{[\mathrm{new}]}\right) \tag{119}$$

and

$$\mathbb{P}\Big(\max\left(\mathsf{f}_2^{[\mathrm{new}]} + Y_{i-1}, \mathsf{a}_3^{[\mathrm{new}]}\right) + Z_{i-1} = \mathsf{a}_3^{[\mathrm{new}]}\Big) \cdot f^{[2]}_{\mathsf{a}}\left(\mathsf{a}_3^{[\mathrm{new}]}, \mathsf{a}_3^{[\mathrm{new}]}\right) \tag{120}$$

where the $\mathsf{a}_3^{[\mathrm{new}]}$ terms are automatically computed via the x^+ terms in (59) and due to (118).

For (119), because $\mathsf{a}_3^{[\mathrm{new}]} \geq 0$, the probabilistic weight quantifies the probability of the event

$$\left\{A_{i-1} > A_{i-2} = S_i + \mathsf{a}_3^{[\mathrm{new}]}\right\} = \left\{A_{i_{\mathrm{nx}}-2} > A_{i_{\mathrm{nx}}-3} = S_{i_{\mathrm{nx}}-1} + \mathsf{a}_3^{[\mathrm{new}]}\right\}. \tag{121}$$

By Lemma 8, source s will make the scheduling decision of $P_{i_{\mathrm{nx}}}$ at time $S_{i_{\mathrm{nx}}-1} + \mathsf{a}_3^{[\mathrm{new}]}$. The above event (121) then implies that when making the scheduling decision for $P_{i_{\mathrm{nx}}}$, the feedback $\mathsf{Ack}_{i_{\mathrm{nx}}-2}$ has not returned yet. Therefore, source s will apply Rule G4 for packet $P_{i_{\mathrm{nx}}}$, which is why we multiply the $f^{[2]}_{\mathsf{ini}}(\mathsf{f}_2^{[\mathrm{new}]}, \mathsf{a}_3^{[\mathrm{new}]})$ term in (119).

For (120), the probabilistic weight quantifies the probability of the event

$$\left\{A_{i-1} = A_{i-2} = S_i + \mathsf{a}_3^{[\mathrm{new}]}\right\} = \left\{A_{i_{\mathrm{nx}}-2} = A_{i_{\mathrm{nx}}-3} = S_{i_{\mathrm{nx}}-1} + \mathsf{a}_3^{[\mathrm{new}]}\right\}. \tag{122}$$

Under this event, acknowledgement packet $\mathsf{Ack}_{i_{\mathrm{nx}}-2}$ returns back to s at the same time when s tries to make a decision for $P_{i_{\mathrm{nx}}}$. Therefore, s will skip Rule G4 and directly switch to Rule G5 for packet $P_{i_{\mathrm{nx}}}$ (when it originally intended to apply Rule G4), which is why we multiply the $f^{[2]}_{\mathsf{a}}(\cdot,\cdot)$ term in (120).

When applying Rule G5, we need to know the state values $(\mathsf{f}_1^{[\mathrm{new}]}, \mathsf{a}_2^{[\mathrm{new}]})$. To that end, we note that for packet $P_{i_{\mathrm{nx}}} = P_{i+1}$, the new $\mathsf{a}_3^{[\mathrm{new}]}$, $\mathsf{f}_1^{[\mathrm{new}]}$, and $\mathsf{a}_2^{[\mathrm{new}]}$ values satisfy the following relationship

$$\mathsf{a}_3^{[\mathrm{new}]} = (D_{i_{\mathrm{nx}}-3} - S_{i_{\mathrm{nx}}-1})^+ \tag{123}$$

$$\leq \mathsf{f}_1^{[\mathrm{new}]} \leq \mathsf{a}_2^{[\mathrm{new}]} \tag{124}$$

$$= (A_{i_{\mathrm{nx}}-2} - S_{i_{\mathrm{nx}}-1})^+ = \mathsf{a}_3^{[\mathrm{new}]} \tag{125}$$

where (123) follows from Lemma 8, which shows that the time to make the decision is $\max(D_{i_{\mathrm{nx}}-3}, S_{i_{\mathrm{nx}}-1}) = S_{i_{\mathrm{nx}}-1} +$

$\left(\mathsf{a}_3^{[\text{new}]}\right)^+$ and because we have $\mathsf{a}_3^{[\text{new}]} \geq 0$ by (118); the first inequality of (124) follows from the definition of f_1 in (12); the second inequality of (124) follows from (13); the first equality of (125) once again follows from (13); and the second equality of (125) follows from that we are under the event described by (122). Jointly, we have proven $\mathsf{a}_3^{[\text{new}]} = \mathsf{f}_1^{[\text{new}]} = \mathsf{a}_2^{[\text{new}]}$. That is why we have the $f_{\mathsf{a}}^{[2]}(\mathsf{a}_3^{[\text{new}]}, \mathsf{a}_3^{[\text{new}]})$ term in (120). Intuitively speaking, the event in (122) is when both $\mathsf{Ack}_{i_{\text{nx}}-2}$ and $\mathsf{Ack}_{i_{\text{nx}}-3}$ return back to s at the same time, which implies $\mathsf{a}_3^{[\text{new}]} = \mathsf{a}_2^{[\text{new}]}$. Since $\mathsf{f}_1^{[\text{new}]}$ is always bracketed by $\mathsf{a}_3^{[\text{new}]}$ and $\mathsf{a}_2^{[\text{new}]}$, all three terms must be identical.

D. Proof of Lemma 9

Before introducing the detailed proof of Lemma 9, we first discuss the behind-the-scene construction of the self-contained parameter set Ω_{sc} in (69)–(71), which could shed some intuition why Lemma 9 must hold in the first place.

Specifically, we construct a series of parameter sets $\Omega^{(l)}$ iteratively for $l = 0, 1, \cdots$. Firstly, we let

$$\Omega^{(0)} \triangleq \{(\mathsf{f}_1, \mathsf{a}_2) : (\mathsf{f}_1, \mathsf{a}_2) = (0, 0)\} \cup \{(\mathsf{f}_2, \mathsf{a}_3) : (\mathsf{f}_2, \mathsf{a}_3) = (0, 0)\}. \tag{126}$$

contain two basic pairs $(\mathsf{f}_1, \mathsf{a}_2) = (0, 0)$ and $(\mathsf{f}_2, \mathsf{a}_3) = (0, 0)$. For any $l \geq 1$, we then define $\Omega^{(l)} \supseteq \Omega^{(l-1)}$ as the *smallest* superset of $\Omega^{(l-1)}$ such that knowing the values of $f_{\mathsf{ini}}^{[2]}(\cdot, \cdot)$, $f_{\mathsf{a}}^{[\text{cmb}]}(\cdot, \cdot)$, and $f_{\mathsf{ini}}^{[\text{cmb}]}(\cdot)$ in (57), (60), and (62), respectively, for those values in $\Omega^{(l)}$ *would have allowed us* to use (56)–(57) and (60)–(62) to compute the values of $f_{\mathsf{a}}^{[2]}(\mathsf{f}_1, \mathsf{a}_2)$ and $f_{\mathsf{ini}}^{[2]}(\mathsf{f}_2, \mathsf{a}_3)$ for all $(\mathsf{f}_1, \mathsf{a}_2)$ and $(\mathsf{f}_2, \mathsf{a}_3)$ in $\Omega^{(l-1)}$.

For example, with the $\Omega^{(0)}$ defined in (126), by (56)–(57) the next set $\Omega^{(1)}$ must contain

$$\Omega^{(1)} \supseteq \left\{(\mathsf{f}_2, \mathsf{a}_3) : \mathsf{f}_2 = \mathsf{a}_3 \in [-y_{\max} - z_{\max}, 0]\right\} \tag{127}$$

The reason is that to compute the value of $f_{\mathsf{a}}^{[2]}(0, 0)$ in the left-hand side of (56)–(57) one must know the values of $f_{\mathsf{ini}}^{[2]}(\mathsf{f}_2, \mathsf{a}_3)$ in (57) satisfying $\mathsf{f}_2 = \mathsf{a}_3 = -x$. Since the x value belongs to $\mathcal{X}_{\mathsf{a}}(0, 0) = [0, y_{\max} + z_{\max}]$ as defined in (67), the new parameter set $\Omega^{(1)}$ must contain the values of $(\mathsf{f}_2, \mathsf{a}_3)$ satisfying (127).

The above construction gradually enlarges the parameter set $\Omega^{(l)}$ to ensure that the computation is feasible for l iterative rounds. We manually verified that $\Omega^{(l)}$ converges to the set described in (69)–(71) when $l \to \infty$. That was how we derived our self-contained parameter set $\Omega_{\text{sc}} = \lim_{l \to \infty} \Omega^{(l)}$ in the first place. In the sequel, instead of the above empirical iterative construction, we directly prove that the converged parameter set Ω_{sc} is self-contained.

Part 1 of the proof:

It is obvious that $(\mathsf{f}_1, \mathsf{a}_2) = (0, 0)$ belongs to the Ω_{sc} defined in (69)–(71).

Part 2 of the proof:

Suppose $(\mathsf{f}_1, \mathsf{a}_2)$ satisfies (69) and we consider the evaluation of the right-hand side of (57). We now show that

$$\mathsf{f}_2^{[\text{new}]} = \mathsf{f}_1 - (\mathsf{a}_2 + x) \tag{128}$$

$$\mathsf{a}_3^{[\text{new}]} = -x \tag{129}$$

evaluated in (57) will satisfy the condition (71), provided that x belongs to the range $\mathcal{X}_{\mathsf{a}}(\mathsf{f}_1, \mathsf{a}_2)$ defined in (67).

Specifically, (i) we have $\mathsf{a}_3^{[\text{new}]} \leq 0$ since $x \geq 0$; (ii) we have $\mathsf{f}_2^{[\text{new}]} \leq \mathsf{a}_3^{[\text{new}]}$ since $\mathsf{a}_2 - \mathsf{f}_1 \geq 0$; (iii) we have

$$\mathsf{f}_2^{[\text{new}]} \geq (\mathsf{f}_1 - \mathsf{a}_2) - (\mathsf{f}_1 - \mathsf{a}_2 + y_{\max})^+ - z_{\max} \tag{130}$$

$$= \min(\mathsf{f}_1 - \mathsf{a}_2, -y_{\max}) - z_{\max} \tag{131}$$

$$= -\max(z_{\max}, y_{\max}) - z_{\max} \tag{132}$$

where (130) follows from using the largest x in the range $\mathcal{X}_{\mathsf{a}}(\mathsf{f}_1, \mathsf{a}_2)$; (131) follows from basic algebra; and (132) follows from $\mathsf{a}_2 - \mathsf{f}_1 \leq z_{\max}$.

We also have (iv) $\mathsf{a}_3^{[\text{new}]} \leq \mathsf{f}_2^{[\text{new}]} + z_{\max}$ because $\mathsf{a}_2 - \mathsf{f}_1 \leq z_{\max}$; and (v) $\mathsf{a}_3^{[\text{new}]} \geq -y_{\max} - z_{\max}$ by choosing the largest x in $\mathcal{X}_{\mathsf{a}}(\mathsf{f}_1, \mathsf{a}_2)$, while setting $\mathsf{f}_1 = \mathsf{a}_2$ to ensure the largest possible x. From (i)–(v), we have proven that $(\mathsf{f}_2^{[\text{new}]}, \mathsf{a}_3^{[\text{new}]})$ satisfy (71).

Part 3 of the proof:

Suppose $(\mathsf{f}_2, \mathsf{a}_3)$ satisfies (70) and we consider the evaluation of the right-hand side of (60)–(62).

Part 3.1: We first consider the term $f_{\mathsf{a}}^{[\text{cmb}]}\left((\mathsf{f}_2 + y)^+, \mathsf{a}_3^+ + k\right)$ in (60). Specifically, define

$$\mathsf{f}_1^{[\text{new}]} = (\mathsf{f}_2 + y)^+ \tag{133}$$

$$\mathsf{a}_2^{[\text{new}]} = \mathsf{a}_3^+ + k \tag{134}$$

where (y, k) satisfies

$$\max(\mathsf{f}_2 + y, \mathsf{a}_3) + z = \mathsf{a}_3^+ + k \tag{135}$$

and $k \leq x$ since we are conditioning on the event $\tilde{\mathsf{A}}_{i-2} = \mathsf{a}_3^+ + k$ with $k \leq x$. We now show that the above $(\mathsf{f}_1^{[\text{new}]}, \mathsf{a}_2^{[\text{new}]})$ must satisfy the condition in (69).

Specifically, (i) we have $\mathsf{f}_1^{[\text{new}]} \geq 0$ by its definition in (133); (ii) we have $\mathsf{f}_1^{[\text{new}]} \leq 2y_{\max}$ since $\mathsf{f}_2 \in [0, y_{\max}]$ by (70) and $y \leq y_{\max}$; (iii) because of $\mathsf{f}_2 = \mathsf{a}_3 \geq 0$, we have

$$\mathsf{a}_2^{[\text{new}]} - \mathsf{f}_1^{[\text{new}]} = (\mathsf{a}_3 + k) - (\mathsf{f}_2 + y) = k - y = z \tag{136}$$

where the last equality follows from (135). Eq. (136) then implies that $\mathsf{a}_2^{[\text{new}]} - \mathsf{f}_1^{[\text{new}]}$ belongs to the interval $[0, z_{\max}]$. From (i)–(iii), we have proven that $(\mathsf{f}_1^{[\text{new}]}, \mathsf{a}_2^{[\text{new}]})$ satisfy (69).

Part 3.2: We now consider the term $f_{\mathsf{ini}}^{[\text{cmb}]}\left((\mathsf{f}_2 + y)^+ - (\mathsf{a}_3^+ + x)\right)$ in (62).

Specifically, define

$$\mathsf{f}_2^{[\text{new}]} = (\mathsf{f}_2 + y)^+ - (\mathsf{a}_3^+ + x) = y - x \tag{137}$$

$$\mathsf{a}_3^{[\text{new}]} = \left(\mathsf{f}_2^{[\text{new}]}\right)^+ = (y - x)^+ \tag{138}$$

$$\mathsf{f}_1^{[\text{new}]} = \left(\mathsf{f}_2^{[\text{new}]}\right)^+ = (y - x)^+ \tag{139}$$

$$\mathsf{a}_2^{[\text{new}]} = \left(\mathsf{f}_2^{[\text{new}]}\right)^+ = (y - x)^+ \tag{140}$$

where the last equality in (137) follows from $\mathsf{f}_2 = \mathsf{a}_3 \geq 0$. Because of (59), the expression $f_{\mathsf{ini}}^{[\text{cmb}]}\left((\mathsf{f}_2 + y)^+ - (\mathsf{a}_3^+ + x)\right)$ in (62) actually contains two more terms $f_{\mathsf{a}}^{[2]}(\mathsf{f}_1^{[\text{new}]}, \mathsf{a}_2^{[\text{new}]})$ and $f_{\mathsf{ini}}^{[2]}(\mathsf{f}_2^{[\text{new}]}, \mathsf{a}_3^{[\text{new}]})$. That is why we expand the single term $\mathsf{f}_2^{[\text{new}]}$ in (137) to also consider the $\mathsf{a}_3^{[\text{new}]}$, $\mathsf{f}_1^{[\text{new}]}$, and $\mathsf{a}_2^{[\text{new}]}$ terms in (138)–(140).

It is straightforward to see that the $(\mathsf{f}_1^{[\text{new}]}, \mathsf{a}_2^{[\text{new}]})$ in (139) and (140) satisfy (69) since $y \leq y_{\max}$ and $0 \leq x$.

We now show that the $(\mathsf{f}_2^{[\text{new}]}, \mathsf{a}_3^{[\text{new}]})$ in (137) and (138) must satisfy either (70) or (71).

Specifically, if $\mathsf{f}_2^{[\text{new}]} \geq 0$, then we must have $\mathsf{a}_3^{[\text{new}]} = \mathsf{f}_2^{[\text{new}]} = y - x \in [0, y_{\max}]$, which satisfies (70).

We now consider the case of $\mathsf{f}_2^{[\text{new}]} < 0$, which immediately implies $\mathsf{a}_3^{[\text{new}]} = 0$. Before we prove the desired conditions, we argue that for any (x, y) value satisfying

$$\mathsf{f}_2^{[\text{new}]} = y - x < -z_{\max} \tag{141}$$

the following probability must be zero

$$\mathbb{P}\left(Y_{i-2} = y, \tilde{\mathsf{A}}_{i-2} > \mathsf{a}_3^+ + x \middle| \tilde{\mathsf{A}}_{i-2} > \mathsf{a}_3^+ \right) = 0. \tag{142}$$

The reason is that by the definition of $\tilde{\mathsf{A}}_{i-2}$ in (21) and because $\mathsf{f}_2 = \mathsf{a}_3$ in (70), the event $\{Y_{i-2} = y, \tilde{\mathsf{A}}_{i-2} > \mathsf{a}_3^+ + x\}$ implies $\{Z_{i-2} > x - y\}$, which in turn implies an impossible event $\{Z_{i-2} > z_{\max}\}$ due to (141). Because (142) is zero for all $\mathsf{f}_2^{[\text{new}]} < -z_{\max}$, when evaluating the right-hand side of (62), we only need to consider the $\mathsf{f}_2^{[\text{new}]}$ values satisfying $\mathsf{f}_2^{[\text{new}]} \geq -z_{\max}$. Therefore, for this case, the $f_{\mathsf{ini}}^{[2]}(\mathsf{f}_2^{[\text{new}]}, \mathsf{a}_3^{[\text{new}]})$ that needs to be evaluated always has its input arguments being $(\mathsf{f}_2^{[\text{new}]}, \mathsf{a}_3^{[\text{new}]}) = (\mathsf{f}_2^{[\text{new}]}, 0)$, which satisfies (71).

Part 4 of the proof:

Suppose $(\mathsf{f}_2, \mathsf{a}_3)$ satisfies (71) and we consider the evaluation of the right-hand side of (60)–(62).

Part 4.1: We first consider the term $f_{\mathsf{a}}^{[\text{cmb}]}\left((\mathsf{f}_2 + y)^+, \mathsf{a}_3^+ + k\right)$ in (60). Specifically, define

$$\mathsf{f}_1^{[\text{new}]} = (\mathsf{f}_2 + y)^+ \tag{143}$$

$$\mathsf{a}_2^{[\text{new}]} = \mathsf{a}_3^+ + k = k \tag{144}$$

where (y, k) satisfies

$$\max(\mathsf{f}_2 + y, \mathsf{a}_3) + z = \mathsf{a}_3^+ + k = k \tag{145}$$

and $k \leq x$ since we are conditioning on the event $\tilde{\mathsf{A}}_{i-2} = \mathsf{a}_3^+ + k$ with $k \leq x$. We now show that the above $(\mathsf{f}_1^{[\text{new}]}, \mathsf{a}_2^{[\text{new}]})$ must satisfy the condition (69).

Specifically, (i) we have $\mathsf{f}_1^{[\text{new}]} \geq 0$ by its definition in (143); (ii) we have $\mathsf{f}_1^{[\text{new}]} \leq y_{\max}$ since $\mathsf{f}_2 \leq 0$ and $y \leq y_{\max}$; (iii) because of $\mathsf{a}_3 \leq 0$, we have

$$\mathsf{a}_2^{[\text{new}]} - \mathsf{f}_1^{[\text{new}]} = k - (\mathsf{f}_2 + y)^+ \tag{146}$$

$$\leq k - \max(\mathsf{f}_2 + y, \mathsf{a}_3) = z \tag{147}$$

where the equality in (147) follows from (145). Eq. (147) then implies that $\mathsf{a}_2^{[\text{new}]} - \mathsf{f}_1^{[\text{new}]} \leq z_{\max}$; (iv) If $\mathsf{f}_1^{[\text{new}]} = 0$, then we have $\mathsf{a}_2^{[\text{new}]} - \mathsf{f}_1^{[\text{new}]} = k \geq 0$ by (143) and (146). If $\mathsf{f}_1^{[\text{new}]} > 0$, then $\mathsf{f}_2 + y > 0$, which, when combined with (145) and $\mathsf{a}_3 \leq 0$, implies $\mathsf{a}_2^{[\text{new}]} - \mathsf{f}_1^{[\text{new}]} = z \geq 0$.

From (i)–(iv), we have proven that $(\mathsf{f}_1^{[\text{new}]}, \mathsf{a}_2^{[\text{new}]})$ satisfy (69).

Part 4.2: We now consider the term $f_{\mathsf{ini}}^{[\text{cmb}]}\left((\mathsf{f}_2 + y)^+ - (\mathsf{a}_3^+ + x)\right)$ in (62). Specifically, define

$$\mathsf{f}_2^{[\text{new}]} = (\mathsf{f}_2 + y)^+ - (\mathsf{a}_3^+ + x) = (\mathsf{f}_2 + y)^+ - x \tag{148}$$

$$\mathsf{a}_3^{[\text{new}]} = \left(\mathsf{f}_2^{[\text{new}]}\right)^+ \tag{149}$$

$$\mathsf{f}_1^{[\text{new}]} = \left(\mathsf{f}_2^{[\text{new}]}\right)^+ \tag{150}$$

$$\mathsf{a}_2^{[\text{new}]} = \left(\mathsf{f}_2^{[\text{new}]}\right)^+ \tag{151}$$

where the last equality in (148) follows from $\mathsf{a}_3 \leq 0$. Because of (59), the expression $f_{\mathsf{ini}}^{[\text{cmb}]}\left((\mathsf{f}_2 + y)^+ - (\mathsf{a}_3^+ + x)\right)$ in (62) actually contains two more terms $f_{\mathsf{a}}^{[2]}(\mathsf{f}_1^{[\text{new}]}, \mathsf{a}_2^{[\text{new}]})$ and $f_{\mathsf{ini}}^{[2]}(\mathsf{f}_2^{[\text{new}]}, \mathsf{a}_3^{[\text{new}]})$. That is why we expand the single term $\mathsf{f}_2^{[\text{new}]}$ in (148) to also consider the $\mathsf{a}_3^{[\text{new}]}$, $\mathsf{f}_1^{[\text{new}]}$, and $\mathsf{a}_2^{[\text{new}]}$ terms in (149)–(151).

It is straightforward to see that the $(\mathsf{f}_1^{[\text{new}]}, \mathsf{a}_2^{[\text{new}]})$ in (150) and (151) satisfy (69) since $\mathsf{f}_2 \leq 0$, $y \leq y_{\max}$, and $0 \leq x$.

We now show that the $(\mathsf{f}_2^{[\text{new}]}, \mathsf{a}_3^{[\text{new}]})$ in (148) and (149) must satisfy either (70) or (71).

Specifically, if $\mathsf{f}_2^{[\text{new}]} \geq 0$, then we must have $\mathsf{a}_3^{[\text{new}]} = \mathsf{f}_2^{[\text{new}]} \in [0, y_{\max}]$ since $\mathsf{f}_2 \leq 0$, $y \leq y_{\max}$, and $0 \leq x$, which satisfies (70).

We now consider the case of $\mathsf{f}_2^{[\text{new}]} < 0$, which immediately implies $\mathsf{a}_3^{[\text{new}]} = 0$. We argue that for any (x, y) value satisfying

$$\mathsf{f}_2^{[\text{new}]} = (\mathsf{f}_2 + y)^+ - x < -z_{\max} \tag{152}$$

the following probability must be zero

$$\mathbb{P}\left(Y_{i-2} = y, \tilde{\mathsf{A}}_{i-2} > \mathsf{a}_3^+ + x \middle| \tilde{\mathsf{A}}_{i-2} > \mathsf{a}_3^+ \right) = 0. \tag{153}$$

The reason is that we are considering the case of $\mathsf{f}_2 \leq \mathsf{a}_3 \leq 0$. In this case, by the definition of $\tilde{\mathsf{A}}_{i-2}$ in (21), the event $\{Y_{i-2} = y, \tilde{\mathsf{A}}_{i-2} > 0 + x\}$ implies $\{Z_{i-2} > x - \max(\mathsf{f}_2 + y, \mathsf{a}_3)\}$, which in turn implies an impossible event $\{Z_{i-2} > z_{\max}\}$ since

$$x - \max(\mathsf{f}_2 + y, \mathsf{a}_3) \geq x - (\mathsf{f}_2 + y)^+ > z_{\max} \tag{154}$$

where the inequality follows from (152). Because (153) is zero for all $\mathsf{f}_2^{[\text{new}]} < -z_{\max}$, when evaluating the right-hand side of (62), we only need to consider the $\mathsf{f}_2^{[\text{new}]}$ values satisfying $\mathsf{f}_2^{[\text{new}]} \geq -z_{\max}$. Therefore, for this case, the $f_{\mathsf{ini}}^{[2]}(\mathsf{f}_2^{[\text{new}]}, \mathsf{a}_3^{[\text{new}]})$ that needs to be evaluated always has its input arguments $(\mathsf{f}_2^{[\text{new}]}, \mathsf{a}_3^{[\text{new}]}) = (\mathsf{f}_2^{[\text{new}]}, 0)$ satisfying (71).

The above 4-part analysis completes the proof of Lemma 9.

APPENDIX G
DERIVATION OF THE BELLMAN EQUATIONS OF ORDER-2 ACHIEVABILITY SCHEMES

The derivation of the Bellman equations consists of four parts: Parts 1 to 3 are derivations of type-1 to type-3 Bellman equations, respectively. Part 4 is about deriving the *self-contained* parameter set in (80) and (81). Each part will be discussed in a separate subsection below.

A. Type-1 Bellman Equations

The intuition of type-1 Bellman equations in (72) is the simplest. At the decision time $t = A_{i-1} = S_{i-1} + \mathsf{a}_1$ of Rule A6, we use x to denote the additional waiting time, after the absolute time A_{i-1}, or, equivalently, after the *relative time* a_1. 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 the ACPS-semi-MDP.

The last term $g^{[2]}_{\mathsf{a}}(0, 0)$ represents the next state value. Specifically, in Rule A6, we send P_i after A_{i-1}. Therefore, for the next packet index $i_{\rm nx} = i + 1$, the decision time must be $\max(S_{i_{\rm nx}-1}, A_{i_{\rm nx}-3}) = S_i = S_{i_{\rm nx}-1}$. This also means that $A_{i_{\rm nx}-2} \leq S_{i_{\rm nx}-1}$ since $A_{i-1} \leq S_i$. However, because the forward delay $Y_i \geq 1$ with probability one, we also have $A_{i_{\rm nx}-1} > S_{i_{\rm nx}-1}$. Jointly, it means that when deciding the send time of the next packet $P_{i_{\rm nx}}$, $\mathsf{Ack}_{i_{\rm nx}-2}$ has returned back to s but $\mathsf{Ack}_{i_{\rm nx}-1}$ has not. Therefore, the scheme will skip Rule A4 and apply Rule A5 (without using Rule A6) for packet $P_{i_{\rm nx}}$. Note that in this scenario, at the decision time $S_{i_{\rm nx}-1}$, the new state values become

$$\mathsf{f}^{[\rm new]}_1 = (D_{i_{\rm nx}-2} - S_{i_{\rm nx}-1})^+ = (D_{i-1} - S_i)^+ = 0 \quad (155)$$

$$\mathsf{a}^{[\rm new]}_2 = (A_{i_{\rm nx}-2} - S_{i_{\rm nx}-1})^+ = (A_{i-1} - S_i)^+ = 0 \quad (156)$$

where (155) follows from (12); and (156) follows from (13). Therefore, it will enter the next state $g^{[2]}_{\mathsf{a}}(0, 0)$, where the subscript "a" signifies that only one acknowledgement packet has returned to s (i.e., $g^{[2]}_{\mathsf{a}}(\cdot, \cdot)$ is the value function under Rule A5).

B. Type-2 Bellman Equations

The intuition of type-2 Bellman equations in (73)–(74) is as follows. Given a fixed pair of $(\mathsf{f}_1, \mathsf{a}_2)$ values, $\tilde{\mathsf{A}}_{i-1}$ in (14) is a random variable representing when s will receive the feedback Ack_{i-1} under a relative time scale with respect to S_{i-1}. Specifically, in the definition (14), the term f_1+Y_{i-1} is the time that P_{i-1} is delivered to d, a_2 is when the backward queue becomes open and can process Ack_{i-1}, and Z_{i-1} is the service time for Ack_{i-1} to clear the backward queue. We then note that at the decision time $t = \max(A_{i-2}, S_{i-1}) = S_{i-1}+\mathsf{a}_2$ of Rule A5, it is possible that we would immediately skip Rule A5 and move on to Rule A6 instead if Ack_{i-1} has returned back to s before the (relative) decision time a_2. As a result, 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 (73) and (74) are conditioned on $\{\tilde{\mathsf{A}}_{i-1} > \mathsf{a}_2\}$.

The term in (73) represents the events that we will skip Rule A5 and switch to Rule A6 instead, i.e., the scenario in which Ack_{i-1} returns back before the tentative decision $S_{i-1}+\mathsf{a}_2+x$. Specifically, the event $\{\tilde{\mathsf{A}}_{i-1} = \mathsf{a}_2 + k\}$ is equivalent to the event $\{A_{i-1} = S_{i-1} + \mathsf{a}_2 + k\}$. When this event happens, we will switch to Rule A6 and the new state value becomes $\mathsf{a}_1 = A_{i-1} - S_{i-1} = \mathsf{a}_2 + k$. That is why the next state value in (73) is set to $g^{[2]}_{\mathsf{aa}}(\mathsf{a}_2 + k)$.

The term $\gamma\big(\mathsf{a}_2 + x, m^+_{Y,1}(x)\big)$ in (74) quantifies the AoI cost of the decision. Herein, the average delay experienced by P_i is not the simple form $\mathbb{E}\{Y_i\}$ as described in the type-1 Bellman equations in (72). Instead, packet P_i could be blocked by the previous packet P_{i-1} that is still in the forward queue. To quantify the additional delay (caused by P_{i-1} clogging the forward queue), we introduce the $m^+_{Y,1}(x)$ term in (15). Specifically, for the conditional expectation expression in (15), $(\mathsf{f}_1 + Y_{i-1})$ is when P_{i-1} will leave the forward queue. If the delivery time of P_{i-1} is later than the scheduled transmission time $(\mathsf{a}_2 + x)$, it will start to block the processing of P_i. Eq. (15) describes that if we eventually commit to the send time, thus the conditioning event $\tilde{\mathsf{A}}_{i-1} > \mathsf{a}_2 + x$, how much average delay of P_i we should expect once we include the service delay Y_i *plus the queueing delay caused by* P_{i-1}. Plugging $m^+_{Y,1}(x)$ into the cost function $\gamma(\cdot, \cdot)$ gives us the AoI cost of the decision.

The term $-\mathsf{v} \cdot (\mathsf{a}_2 + x)$ is once again the average-cost adjustment term of the ACPS-semi-MDP.

The last term $g^{[2]}_{\mathsf{ini}}(\mathsf{f}_1 - (\mathsf{a}_2 + x), -x)$ represents the next state value. Specifically, in Rule A5, we send P_i after the return of Ack_{i-2} but before the return of Ack_{i-1}. Therefore, for the next packet index $i_{\rm nx} = i + 1$, the decision time must be $\max(S_{i_{\rm nx}-1}, A_{i_{\rm nx}-3}) = S_i = S_{i_{\rm nx}-1}$. By the same observation, at the next decision time $S_{i_{\rm nx}-1}$, the feedback packet $\mathsf{Ack}_{i_{\rm nx}-2}$ has not returned to s yet since $A_{i-1} > S_i$. Jointly it means that we must be activating Rule A4 when we try to decide the send time for $P_{i_{\rm nx}}$. That is why we use the type-3 Bellman equations $g^{[2]}_{\mathsf{ini}}(\cdot, \cdot)$ that is tailored for Rule A4.

We now analyze the next state values $(\mathsf{f}^{[\rm new]}_2, \mathsf{a}^{[\rm new]}_3)$. Specifically, we have

$$\begin{aligned}
\mathsf{f}^{[\rm new]}_2 &= \max(S_{i_{\rm nx}-2}, D_{i_{\rm nx}-3}) - S_{i_{\rm nx}-1} && (157)\\
&= \max(S_{i-1}, D_{i-2}) - S_i && (158)\\
&= (S_{i-1} + \mathsf{f}_1) - (S_{i-1} + \mathsf{a}_2 + x) && (159)\\
&= \mathsf{f}_1 - (\mathsf{a}_2 + x) &&\\
\mathsf{a}^{[\rm new]}_3 &= \max(A_{i_{\rm nx}-3}, S_{i_{\rm nx}-2}) - S_{i_{\rm nx}-1} && (160)\\
&= \max(A_{i-2}, S_{i-1}) - S_i && (161)\\
&= (S_{i-1} + \mathsf{a}_2) - (S_{i-1} + \mathsf{a}_2 + x) = -x && (162)
\end{aligned}$$

where (157) follows from (18); (158) follows from $i_{\rm nx} = i+1$; (159) follows from (12) and our target send time is $S_i = S_{i-1} + \mathsf{a}_2 + x$; (160) follows from (20); (161) follows from $i_{\rm nx} = i + 1$; (162) follows from (13) and our target send time is $S_i = S_{i-1} + \mathsf{a}_2 + x$. The above argument shows that it will enter the next state $g^{[2]}_{\mathsf{ini}}(\mathsf{f}_1 - (\mathsf{a}_2 + x), -x)$, where the subscript "ini" signifies s will activate Rule A4 when deciding the send time of packet $P_{i_{\rm nx}}$.

C. Type-3 Bellman Equations

The intuition of type-3 Bellman equations (77)–(79) is as follows. Given a fixed pair of $(\mathsf{f}_2, \mathsf{a}_3)$ values, $\tilde{\mathsf{A}}_{i-2}$ in (21) is a random variable representing when s will receive the feedback Ack_{i-2} under a relative time scale with respect to S_{i-1}. Specifically, in the definition (21), the term $\mathsf{f}_2 + Y_{i-2}$ is the time that P_{i-2} is delivered to d, a_3 is when the backward queue becomes open and can process Ack_{i-2}, and Z_{i-2} is the service time for Ack_{i-2} to clear the backward queue. We then note that the decision time of Rule A4 satisfies

$$t = \max(S_{i-1}, A_{i-3}) = \max(S_{i-2}, A_{i-3}, S_{i-1}) \quad (163)$$
$$= S_{i-1} + \max(\max(S_{i-2}, A_{i-3}) - S_{i-1}, 0) \quad (164)$$
$$= S_{i-1} + \mathsf{a}_3^+ \quad (165)$$

where the last equality follows from (20). Since it is possible that we would immediately skip Rule A4 and activate Rule A5 instead if Ack_{i-2} has returned back to s before the (relative) decision time a_3^+, whenever we are making the optimal decision for Rule A4, we are implicitly assuming Ack_{i-2} returns *after* (the relative) time a_3^+. That is, we are under the event $\{\tilde{\mathsf{A}}_{i-2} > \mathsf{a}_3^+\}$. That is why all the *state transition probabilities* in (77), (78), and (79) are conditioned on $\{\tilde{\mathsf{A}}_{i-2} > \mathsf{a}_3^+\}$.

The term in (77) represents the events that we will skip Rule A4 and switch to Rule A5 instead, i.e., the scenario in which Ack_{i-2} returns back to s before the tentative decision $S_{i-1} + \mathsf{a}_3^+ + x$. Specifically, the event $\{\tilde{\mathsf{A}}_{i-2} = \mathsf{a}_3^+ + k\}$ is equivalent to the event $\{A_{i-2} = S_{i-1} + \mathsf{a}_3^+ + k\}$. When this event happens, we will switch to Rule A5 and the new state values become

$$\mathsf{f}_1 = (D_{i-2} - S_{i-1})^+ \quad (166)$$
$$= ((\max(S_{i-2}, D_{i-3}) + Y_{i-2}) - S_{i-1})^+ \quad (167)$$
$$= (\mathsf{f}_2 + Y_{i-2})^+ \quad (168)$$
$$\mathsf{a}_2 = (A_{i-2} - S_{i-1})^+ \quad (169)$$
$$= (\mathsf{a}_3^+ + k)^+ = \mathsf{a}_3^+ + k \quad (170)$$

where (166) follows from the definition of f_1 in (12); (167) follows from the queue evolution equation (2); (168) follows from the definition of f_2 in (18); (169) from the definition of a_2 in (13); (170) follows from that this discussion is under the event $\{A_{i-2} = S_{i-1} + \mathsf{a}_3^+ + k\}$. Jointly, it implies that the next state value in (77) is set to $g_{\mathsf{a}}^{[\mathrm{cmb}]}\left((\mathsf{f}_2 + y)^+, \mathsf{a}_3^+ + k\right)$.

Nonetheless, unlike the type-2 Bellman equations, there is some subtlety for the type-3 Bellman equations. That is, even if we skip Rule A4 and move on to Rule A5 (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}. If that happens, we will immediately skip Rule A5 again and move on to Rule A6 instead. Namely, s could switch directly from Rule A4 to Rule A6 without activating Rule A5 at all. As a result, we introduce the combined value function $g_{\mathsf{a}}^{[\mathrm{cmb}]}(\cdot,\cdot)$ in (75), which further allows for the switching to Rule A6 depending on the arrival time of Ack_{i-1}.

Specifically, by (75), the term $g_{\mathsf{a}}^{[\mathrm{cmb}]}\left((\mathsf{f}_2 + y)^+, \mathsf{a}_3^+ + k\right)$ is actually a summation of two sub-terms

$$\mathbb{P}\left(\max\left((\mathsf{f}_2 + y)^+ + Y_{i-1}, \mathsf{a}_3^+ + k\right) + Z_{i-1} > \mathsf{a}_3^+ + k\right) \cdot g_{\mathsf{a}}^{[2]}\left((\mathsf{f}_2 + y)^+, \mathsf{a}_3^+ + k\right) \quad (171)$$

and

$$\mathbb{P}\left(\max\left((\mathsf{f}_2 + y)^+ + Y_{i-1}, \mathsf{a}_3^+ + k\right) + Z_{i-1} = \mathsf{a}_3^+ + k\right) \cdot g_{\mathsf{aa}}^{[2]}\left(\mathsf{a}_3^+ + k\right). \quad (172)$$

For (171), the probabilistic weight quantifies the probability of the event

$$\left\{A_{i-1} > A_{i-2} = S_{i-1} + \mathsf{a}_3^+ + k\right\}. \quad (173)$$

Under this event, source s will abandon Rule A4, switch to Rule A5 for packet P_i without activating Rule A6, which is why we multiply the $g_{\mathsf{a}}^{[2]}(\cdot,\cdot)$ term in (171).

For (172), the probabilistic weight quantifies the probability of the event

$$\left\{A_{i-1} = A_{i-2} = S_{i-1} + \mathsf{a}_3^+ + k\right\}. \quad (174)$$

Under this event, two acknowledgement packets Ack_{i-2} and Ack_{i-1} return back to s at the same time instant. Therefore, s will abandon Rule A4, then immediately abandon Rule A5 again, and switch to Rule A6 for packet P_i, which is why we multiply the $g_{\mathsf{aa}}^{[2]}(\cdot)$ term in (172). When applying Rule A6, we need to know the state value a_1, which, by definition, is $A_{i-1} - S_{i-1} = A_{i-2} - S_{i-1} = \mathsf{a}_3^+ + k$ in this event. That is why we have the $g_{\mathsf{aa}}^{[2]}\left(\mathsf{a}_3^+ + k\right)$ term in (172). The discussion of the term in (77) is complete.

The term $\gamma\left(\mathsf{a}_3^+ + x, m_{Y,2}^+(x)\right)$ in (78) quantifies the AoI cost of the decision. Herein, the average delay experienced by P_i could come from the processing times of P_i, P_{i-1}, and P_{i-2} since P_i can get stuck behind two packets P_{i-1} and P_{i-2} in the forward queue. To quantify the expected processing plus queueing delay of P_i, we introduce the $m_{Y,2}^+(x)$ term in (22). Also see the discussion in Appendix F-B. Plugging $m_{Y,2}^+(x)$ into the cost function $\gamma(\cdot,\cdot)$ gives us the AoI cost of the decision.

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

The last term $g_{\mathsf{ini}}^{[\mathrm{cmb}]}(\cdot,\cdot)$ in (79) represents the next state value function. Specifically, we analyze the state values $(\mathsf{f}_2^{[\mathrm{new}]}, \mathsf{a}_3^{[\mathrm{new}]})$ for the next packet index $i_{\mathrm{nx}} = i+1$ as follows.

$$\mathsf{f}_2^{[\mathrm{new}]} = \max(S_{i-1}, D_{i-2}) - S_i \quad (175)$$
$$= (S_{i-1} + (D_{i-2} - S_{i-1})^+) - (S_{i-1} + \mathsf{a}_3^+ + x) \quad (176)$$
$$= (\mathsf{f}_2 + Y_{i-2})^+ - (\mathsf{a}_3^+ + x) \quad (177)$$

and

$$\mathsf{a}_3^{[\mathrm{new}]} = \max(A_{i-2}, S_{i-1}) - S_i \quad (178)$$
$$= \max(\max(D_{i-2}, A_{i-3}) + Z_{i-2}, S_{i-1}) - (S_{i-1} + \mathsf{a}_3^+ + x) \quad (179)$$
$$= \left(\max(D_{i-2} - S_{i-1}, A_{i-3} - S_{i-1}) + Z_{i-2}\right)^+ - (\mathsf{a}_3^+ + x) \quad (180)$$
$$= \left(\max(\mathsf{f}_2 + Y_{i-2}, A_{i-3} - S_{i-1}) + Z_{i-2}\right)^+ - (\mathsf{a}_3^+ + x) \quad (181)$$
$$= (\max(\mathsf{f}_2 + Y_{i-2}, \mathsf{a}_3) + Z_{i-2})^+ - (\mathsf{a}_3^+ + x) \quad (182)$$
$$= \max(\mathsf{f}_2 + Y_{i-2}, \mathsf{a}_3) + Z_{i-2} - (\mathsf{a}_3^+ + x) \geq 0 \quad (183)$$

where (175) follows from (18) and $i_{\mathrm{nx}} = i+1$; (176) follows from that our target send time is $S_i = S_{i-1} + \mathsf{a}_3^+ + x$; (177) follows from rewriting D_{i-2} using the queue evolution equation (2) and by the definition of f_2 in (13).

Eq. (178) follows from (20) and $i_{\text{nx}} = i+1$; (179) follows from replacing A_{i-2} by the queue evolution equation (3) and our target send time being $S_i = S_{i-1} + \mathsf{a}_3^+ + x$; (180) follows from subtracting S_{i-1} in all terms of (179); (181) follows from rewriting D_{i-2} using the queue evolution equation (2) and by the definition of f_2 in (13); (182) follows from that we can rewrite the $A_{i-3} - S_{i-1}$ by $\max(\mathsf{f}_2, A_{i-3} - S_{i-1}) = \mathsf{a}_3$ in (19) and by noting $\mathsf{f}_2 + Y_{i-2} \geq \mathsf{f}_2$; and finally since we are in the event that $\bar{\mathsf{A}}_{i-2} > \mathsf{a}_3^+ + x$ and by the definition of (21), we can remove the $(\cdot)^+$ operator in (182) without loss of generality.

Even though we have figured out the $(\mathsf{f}_2^{[\text{new}]}, \mathsf{a}_3^{[\text{new}]})$ value as in (177) and (183), respectively, there is a subtle relationship when considering the next state value function. That is, even though one may expect that we will be applying Rule A4 again for packet $P_{i_{\text{nx}}}$ under state values $(\mathsf{f}_2^{[\text{new}]}, \mathsf{a}_3^{[\text{new}]})$, it is possible that it may skip Rule A4 and move on to Rule A5 if $\mathsf{Ack}_{i_{\text{nx}}-2}$ returns to s before we make the decision at time $\max(S_{i_{\text{nx}}-1}, A_{i_{\text{nx}}-3})$. Also see our earlier discussion of the $g_{\mathsf{a}}^{[\text{cmb}]}(\cdot,\cdot)$ term in (77).

As a result, we introduce the combined value function $g_{\mathsf{ini}}^{[\text{cmb}]}(\cdot)$ in (76), which further allows for the switching to Rule A5 depending on the arrival time of $\mathsf{Ack}_{i_{\text{nx}}-2}$.

Specifically, by (76), the term $g_{\mathsf{ini}}^{[\text{cmb}]}\left(\mathsf{f}_2^{[\text{new}]}, \mathsf{a}_3^{[\text{new}]}\right)$ in (79) is actually a summation of two sub-terms

$$\mathbb{P}\Big(\max\left(\mathsf{f}_2^{[\text{new}]} + Y_{i-1}, \mathsf{a}_3^{[\text{new}]}\right) + Z_{i-1} > \mathsf{a}_3^{[\text{new}]}\Big) \cdot g_{\mathsf{ini}}^{[2]}\left(\mathsf{f}_2^{[\text{new}]}, \mathsf{a}_3^{[\text{new}]}\right) \tag{184}$$

and

$$\sum_{y=1}^{y_{\max}} \mathbb{P}\Big(Y_{i-1} = y, \max\left(\mathsf{f}_2^{[\text{new}]} + y, \mathsf{a}_3^{[\text{new}]}\right) + Z_{i-1} = \mathsf{a}_3^{[\text{new}]}\Big) \cdot g_{\mathsf{a}}^{[\text{cmb}]}\left((\mathsf{f}_2^{[\text{new}]} + y)^+, \mathsf{a}_3^{[\text{new}]}\right). \tag{185}$$

with the $(\mathsf{f}_2^{[\text{new}]}, \mathsf{a}_3^{[\text{new}]})$ values specified in (177) and (183), respectively.

For (184), because $\mathsf{a}_3^{[\text{new}]} \geq 0$, the probabilistic weight quantifies the probability of the event

$$\begin{aligned}&\left\{A_{i-1} > A_{i-2} = S_i + \mathsf{a}_3^{[\text{new}]}\right\} \\ &= \left\{A_{i_{\text{nx}}-2} > A_{i_{\text{nx}}-3} = S_{i_{\text{nx}}-1} + \mathsf{a}_3^{[\text{new}]}\right\}.\end{aligned} \tag{186}$$

By (165), the source s will make the scheduling decision of $P_{i_{\text{nx}}}$ at time $S_{i_{\text{nx}}-1} + \mathsf{a}_3^{[\text{new}]}$. The above event (186) then implies that when making the scheduling decision for $P_{i_{\text{nx}}}$, the feedback $\mathsf{Ack}_{i_{\text{nx}}-2}$ has not returned yet. Therefore, source s will apply Rule A4 for packet $P_{i_{\text{nx}}}$, which is why we multiply the $g_{\mathsf{ini}}^{[2]}(\mathsf{f}_2^{[\text{new}]}, \mathsf{a}_3^{[\text{new}]})$ term in (184).

For (185), the probabilistic weight quantifies the probability of the event

$$\begin{aligned}&\left\{A_{i-1} = A_{i-2} = S_i + \mathsf{a}_3^{[\text{new}]}, Y_{i-1} = y\right\} \\ &= \left\{A_{i_{\text{nx}}-2} = A_{i_{\text{nx}}-3} = S_{i_{\text{nx}}-1} + \mathsf{a}_3^{[\text{new}]}, Y_{i_{\text{nx}}-2} = y\right\}.\end{aligned} \tag{187}$$

Under this event, acknowledgement packet $\mathsf{Ack}_{i_{\text{nx}}-2}$ returns back to s at the same time when s tries to make a decision for $P_{i_{\text{nx}}}$. Therefore, s will skip Rule A4 and directly switch to Rule A5 for packet $P_{i_{\text{nx}}}$ (when it originally intended to apply Rule A4), which is why we multiply the $g_{\mathsf{a}}^{[\text{cmb}]}(\cdot,\cdot)$ term in (185).

When applying Rule A5, we need to know the state values $(\mathsf{f}_1^{[\text{new}]}, \mathsf{a}_2^{[\text{new}]})$. To that end, we note that for packet $P_{i_{\text{nx}}} = P_{i+1}$, we have

$$\mathsf{f}_1^{[\text{new}]} = (D_{i_{\text{nx}}-2} - S_{i_{\text{nx}}-1})^+ \tag{188}$$

$$= (\mathsf{f}_2^{[\text{new}]} + Y_{i_{\text{nx}}-2})^+ \tag{189}$$

$$\mathsf{a}_2^{[\text{new}]} = (A_{i_{\text{nx}}-2} - S_{i_{\text{nx}}-1})^+ = \mathsf{a}_3^{[\text{new}]} \tag{190}$$

where (188) follows from the definition of f_1 in (12) but applied to the next packet $P_{i_{\text{nx}}}$; (189) follows from the definition of f_2 in (18); the first equality of (190) follows from (13); and the second equality of (190) follows from that we are under the event described by (187). Jointly, (189) and (190) is the reason for the input arguments of $g_{\mathsf{a}}^{[\text{cmb}]}\left((\mathsf{f}_2^{[\text{new}]} + y)^+, \mathsf{a}_3^{[\text{new}]}\right)$ term in (185).

An astute reader may question why we use the combined function $g_{\mathsf{a}}^{[\text{cmb}]}\left((\mathsf{f}_2^{[\text{new}]} + y)^+, \mathsf{a}_3^{[\text{new}]}\right)$ in (185) (and thus in (76)) instead of the vanilla version $g_{\mathsf{a}}^{[2]}(\cdot,\cdot)$. The reason is that the event in (187) quantifies the probability that for the next packet $P_{i_{\text{nx}}}$, how likely we would skip Rule A4 and switch to Rule A5. However, there is a chance that after switching to Rule A5, source s may decide to further switch to Rule A6 immediately. Therefore, we use the combined function $g_{\mathsf{a}}^{[\text{cmb}]}(\cdot,\cdot)$, see its definition in (75), to further take into account the probability of a *stacked switch* from Rules A4 to A5 and then from A5 to A6 all at the same time instant. The discussion of the term in (79) is complete.

D. *The Self-Contained Parameter Set for The Order-2 Achievability Scheme*

Before introducing the detailed proof of the self-contained parameter set for the order-2 achievability schemes (as described in (80)–(81)), we would like to refer the readers to our discussion of the behind-the-scene construction of the self-contained parameter set for the order-2 genie-aided schemes in Appendix F-D. Essentially, we use the same iterative parameter set construction method for both the order-2 genie-aided schemes and the order-2 achievability schemes. The discussion in Appendix F-D could shed some insight on why a self-contained parameter set must exist in the first place.

Part 1 of the proof:

We first need to show that the *ground state* belongs to the conditions described in (80)–(81). It is obvious that the ground state $(\mathsf{f}_1, \mathsf{a}_2) = (0, 0)$ satisfies (80).

Part 2 of the proof:

Suppose $(\mathsf{f}_1, \mathsf{a}_2)$ satisfies (80) and we consider the evaluation of the right-hand side of (74). We now show that

$$\mathsf{f}_2^{[\text{new}]} = \mathsf{f}_1 - (\mathsf{a}_2 + x) \tag{191}$$

$$\mathsf{a}_3^{[\text{new}]} = -x \tag{192}$$

will satisfy the condition (81), provided that x belongs to the range $\mathcal{X}_{\mathsf{a}}(\mathsf{f}_1, \mathsf{a}_2)$ defined in (67).

Specifically, (i) we have $\mathsf{a}_3^{[\text{new}]} \leq 0$ since $x \geq 0$; (ii) we have $\mathsf{f}_2^{[\text{new}]} \leq \mathsf{a}_3^{[\text{new}]}$ since $\mathsf{a}_2 - \mathsf{f}_1 \geq 0$. Note that (i) and (ii) jointly imply $\mathsf{f}_2^{[\text{new}]} \leq 0$; and (iii) by plugging in the largest possible $x \in \mathcal{X}_{\mathsf{a}}(\mathsf{f}_1, \mathsf{a}_2)$ to (191), we have

$$\mathsf{f}_2^{[\text{new}]} \geq (\mathsf{f}_1 - \mathsf{a}_2) - (\mathsf{f}_1 + y_{\max} - \mathsf{a}_2)^+ - z_{\max} \tag{193}$$
$$= -\max(y_{\max}, \mathsf{a}_2 - \mathsf{f}_1) - z_{\max} \tag{194}$$
$$\geq -y_{\max} - 2z_{\max} - \max(y_{\max}, z_{\max}) \tag{195}$$

where (194) follows from basic algebraic simplification; and (195) follows from choosing the largest a_2 and the smallest $\mathsf{f}_1 = 0$ in (80). From (i)–(iii), we have proven that $(\mathsf{f}_2^{[\text{new}]}, \mathsf{a}_3^{[\text{new}]})$ satisfy (81).

Part 3 of the proof:

Suppose $(\mathsf{f}_2, \mathsf{a}_3)$ satisfies (81) and we consider the evaluation of the right-hand side of (77)–(79).

Part 3.1: We first consider the term $g_{\mathsf{a}}^{[\text{cmb}]}\left((\mathsf{f}_2 + y)^+, \mathsf{a}_3^+ + k\right)$ in (77). Specifically, define

$$\mathsf{f}_1^{[\text{new}]} = (\mathsf{f}_2 + y)^+ \tag{196}$$
$$\mathsf{a}_2^{[\text{new}]} = \mathsf{a}_3^+ + k \tag{197}$$

where (y, k) satisfies

$$\max(\mathsf{f}_2 + y, \mathsf{a}_3) + z = \mathsf{a}_3^+ + k \tag{198}$$

and $k \leq x \in \mathcal{X}_{\mathsf{ini}}(\mathsf{f}_2, \mathsf{a}_3)$ since we are conditioning on the event $\tilde{\mathsf{A}}_{i-2} = \mathsf{a}_3^+ + k$ with $k \leq x$. We now show that the above $(\mathsf{f}_1^{[\text{new}]}, \mathsf{a}_2^{[\text{new}]})$ must satisfy the condition of (80).

Specifically, (i) we have $\mathsf{f}_1^{[\text{new}]} \geq 0$ by its definition in (196); (ii) we have $\mathsf{f}_1^{[\text{new}]} \leq \mathsf{a}_2^{[\text{new}]}$ since if $\mathsf{f}_1^{[\text{new}]} = 0$, then we obviously have $\mathsf{f}_1^{[\text{new}]} \leq \mathsf{a}_2^{[\text{new}]}$ because $\mathsf{a}_2^{[\text{new}]} \geq 0$ by (197). If $\mathsf{f}_1^{[\text{new}]} > 0$, then $\mathsf{f}_1^{[\text{new}]}$ in (196) is no larger than the left-hand side of (198). This again implies $\mathsf{f}_1^{[\text{new}]} \leq \mathsf{a}_2^{[\text{new}]}$.

Finally, (iii) $\mathsf{a}_2^{[\text{new}]}$, by definition (197), is the same as the value of the left-hand side of (198). If we choose the largest possible f_2 and a_3 that satisfy (81) and choose the largest $y = y_{\max}$ and $z = z_{\max}$, then the left-hand side of (198) is upper bounded by

$$\mathsf{a}_2^{[\text{new}]} \leq y_{\max} + z_{\max} + \max(y_{\max}, z_{\max}). \tag{199}$$

From (i)–(iii), we have proven that $(\mathsf{f}_1^{[\text{new}]}, \mathsf{a}_2^{[\text{new}]})$ satisfy (80).

Part 3.2: We now consider the term

$$g_{\mathsf{ini}}^{[\text{cmb}]}\left((\mathsf{f}_2 + y)^+ - (\mathsf{a}_3^+ + x),\right.$$
$$\left.\max(\mathsf{f}_2 + y, \mathsf{a}_3) + z - (\mathsf{a}_3^+ + x)\right) \tag{200}$$

in (79). Specifically, define

$$\mathsf{f}_2^{[\text{new}]} = (\mathsf{f}_2 + y)^+ - (\mathsf{a}_3^+ + x) \tag{201}$$
$$\mathsf{a}_3^{[\text{new}]} = \max(\mathsf{f}_2 + y, \mathsf{a}_3) + z - (\mathsf{a}_3^+ + x) \tag{202}$$
$$\mathsf{f}_1^{[\text{new}]} = \left(\mathsf{f}_2^{[\text{new}]} + \tilde{y}\right)^+ \tag{203}$$
$$\mathsf{a}_2^{[\text{new}]} = \mathsf{a}_3^{[\text{new}]}. \tag{204}$$

where $(\mathsf{f}_2, \mathsf{a}_3, y, z, x)$ satisfy the inequality

$$\max(\mathsf{f}_2 + y, \mathsf{a}_3) + z > \mathsf{a}_3^+ + x \tag{205}$$

and $(\mathsf{f}_2^{[\text{new}]}, \mathsf{a}_3^{[\text{new}]}, \tilde{y}, \tilde{z})$ satisfy the inequality

$$\max(\mathsf{f}_2^{[\text{new}]} + \tilde{y}, \mathsf{a}_3^{[\text{new}]}) + \tilde{z} = \mathsf{a}_3^{[\text{new}]}. \tag{206}$$

Because of (76), the expression $g_{\mathsf{ini}}^{[\text{cmb}]}(\cdot,\cdot)$ in (79) actually contains two more terms $g_{\mathsf{a}}^{[\text{cmb}]}(\mathsf{f}_1^{[\text{new}]}, \mathsf{a}_2^{[\text{new}]})$ and $g_{\mathsf{ini}}^{[2]}(\mathsf{f}_2^{[\text{new}]}, \mathsf{a}_3^{[\text{new}]})$. Specifically, the event $\{\tilde{\mathsf{A}}_{i-2} > \mathsf{a}_3^+ + x\}$ in (79) implies (205) and the event $\{\max(\mathsf{f}_2^{[\text{new}]} + \tilde{y}, \mathsf{a}_3^{[\text{new}]}) + Z_{i-1} = \mathsf{a}_3^{[\text{new}]}\}$ in (76) implies (206).

We first show that the $(\mathsf{f}_2^{[\text{new}]}, \mathsf{a}_3^{[\text{new}]})$ in (201) and (202) must satisfy (81). We note that (i)

$$\mathsf{f}_2^{[\text{new}]} \geq -(\mathsf{a}_3^+ + x) \tag{207}$$
$$\geq -\max(\mathsf{f}_2 + y, \mathsf{a}_3) - z \tag{208}$$
$$\geq -\max(2y_{\max}, y_{\max} + z_{\max}) + z_{\max} \tag{209}$$

where (207) follows from (201); (208) is because $(\mathsf{a}_3^+ + x)$ must satisfy (205); (209) follows from choosing the largest $\mathsf{f}_2 = y_{\max}$ and $\mathsf{a}_3 = y_{\max} + z_{\max}$ that still satisfy (81) and the largest $y = y_{\max}$ and $z = z_{\max}$.

Additionally we have (ii)

$$\mathsf{f}_2^{[\text{new}]} \leq (\mathsf{f}_2 + y)^+ - \mathsf{a}_3^+ \tag{210}$$

which follows from (201) while setting $x = 0$. If $\mathsf{f}_2 \geq 0$, then because $\mathsf{a}_3 \geq \mathsf{f}_2$, the above inequality (210) implies $\mathsf{f}_2^{[\text{new}]} \leq y_{\max}$. If $\mathsf{f}_2 < 0$, then (210) implies $\mathsf{f}_2^{[\text{new}]} \leq y - 0 \leq y_{\max}$ as well. As a result, we always have $\mathsf{f}_2^{[\text{new}]} \leq y_{\max}$.

We now observe (iii)

$$\mathsf{a}_3^{[\text{new}]} - \mathsf{f}_2^{[\text{new}]}$$
$$= \max(\mathsf{f}_2 + y, \mathsf{a}_3) + z - (\mathsf{f}_2 + y)^+ \tag{211}$$

If $\mathsf{f}_2 + y \leq 0$, then because $\max(\mathsf{f}_2 + y, \mathsf{a}_3) + z > \mathsf{a}_3^+ + x \geq 0$ due to (205), we must have $\mathsf{a}_3^{[\text{new}]} - \mathsf{f}_2^{[\text{new}]} \geq 0$. If $\mathsf{f}_2 + y > 0$, then (211) implies

$$(211) = (\mathsf{a}_3 - \mathsf{f}_2 - y)^+ + z \geq 0. \tag{212}$$

Finally, we have (iv)

$$\mathsf{a}_3^{[\text{new}]} \leq \max(\mathsf{f}_2 + y, \mathsf{a}_3) + z - \mathsf{a}_3 \tag{213}$$
$$= (\mathsf{f}_2 + y - \mathsf{a}_3)^+ + z \leq y_{\max} + z_{\max} \tag{214}$$

where (213) follows from (202) and from setting $x = 0$ and removing the $(\cdot)^+$ operator; and the inequality of (214) follows from $\mathsf{f}_2 \leq \mathsf{a}_3$ and from choosing the largest $y = y_{\max}$ and $z = z_{\max}$. From (i)–(iv), we have proven that the new $(\mathsf{f}_2^{[\text{new}]}, \mathsf{a}_3^{[\text{new}]})$ satisfy (81).

We now show that the $(\mathsf{f}_1^{[\text{new}]}, \mathsf{a}_2^{[\text{new}]})$ in (203) and (204) must satisfy (80). We have (i) $\mathsf{f}_1^{[\text{new}]} \geq 0$ by (203); (ii) We have

$$\mathsf{a}_2^{[\text{new}]} - \mathsf{f}_1^{[\text{new}]}$$
$$= \mathsf{a}_3^{[\text{new}]} - \left(\mathsf{f}_2^{[\text{new}]} + \tilde{y}\right)^+ \tag{215}$$
$$\geq \mathsf{a}_3^{[\text{new}]} - \max(\mathsf{f}_2^{[\text{new}]} + \tilde{y}, \mathsf{a}_3^{[\text{new}]}) = \tilde{z} \geq 0. \tag{216}$$

where (215) follows from (204) and (203); the first inequality of (216) follows from $\mathsf{a}_3^{[\text{new}]} > 0$, which is proven by comparing (202) and (205); the equality of (216) follows from (206). As a result, we have proven $\mathsf{a}_2^{[\mathrm{new}]} - \mathsf{f}_1^{[\mathrm{new}]} \geq 0$.

Finally, we have (iv) $\mathsf{a}_2^{[\mathrm{new}]} = \mathsf{a}_3^{[\mathrm{new}]} \leq y_{\max} + z_{\max}$ since we have already proven $\mathsf{a}_3^{[\mathrm{new}]}$ satisfies (81).

From (i)–(iv), we have proven that the new $(\mathsf{f}_1^{[\mathrm{new}]}, \mathsf{a}_2^{[\mathrm{new}]})$ satisfy (80).

The above 3-part analysis completes the proof of the self-contained parameter set in (80) and (81).