Multi-Source AoI-Constrained Resource Minimization Under HARQ: Heterogeneous Sampling Processes

Saeid Sadeghi Vilni, Student Member, IEEE, Mohammad Moltafet, Member, IEEE, Markus Leinonen, Member, IEEE, and Marian Codreanu, Member, IEEE

Abstract—We consider a multi-source hybrid automatic repeat request (HARQ) based system, where a transmitter sends status update packets of random arrival (i.e., uncontrollable sampling) and generate-at-will (i.e., controllable sampling) sources to a destination through an error-prone channel. We develop transmission scheduling policies to minimize the average number of transmissions subject to an average age of information (AoI) constraint. First, we consider known environment (i.e., known system statistics) and develop a near-optimal deterministic transmission policy and a learned policy in high variability dynamic transmission (LC-DT) policy. The former policy is derived by casting the main problem into a constrained Markov decision process (CMDP) problem, which is then solved using the Lagrangian relaxation, relative value iteration algorithm, and bisection. The LC-DT policy is developed via the drift-plus-penalty (DPP) method by transforming the main problem into a sequence of per-slot problems. Finally, we consider unknown environment and devise a learning-based transmission policy by relaxing the CMDP problem into an MDP problem using the DPP method and then adopting the deep Q-learning algorithm. Numerical results show that the proposed policies achieve near-optimal performance and illustrate the benefits of HARQ in status updating.

Index Terms—Age of information (AoI), constrained Markov decision process (CMDP), dynamic programming, Lagrangian, Lyapunov, machine learning, multi-source status update.

I. INTRODUCTION

There is a growing demand for Internet-of-Things (IoT) and cyber-physical systems such as autonomous vehicles, wireless industrial automation, and health monitoring that rely heavily on real-time (fresh) status updates. In these systems, a source (containing a sensor) monitors a physical phenomenon such as temperature, pressure, or motion and sends status updates to a destination (e.g., a remote controller) for decision-making [1], [2], [3], [4]. The Age of Information (AoI) [1], [2], [3], [4] is a destination-centric metric for evaluating the freshness of information. AoI is defined as the difference between the current time and the generation time of the last received packet at a destination [1], [2], [3], [4]. Each status update packet contains a timestamp representing the time when the sample was generated and the measured value of the monitored process. At time instant $t$, denoting the timestamp of the last received status update packet by $U_t$, the AoI, $\delta_t$, is defined as $\delta_t = t - U_t$ [1], [2], [3], [4], [5], [6].

The reliability of data transmissions under an error-prone communication channel can be enhanced via retransmission protocols [7]. Automatic repeat request (ARQ) protocols are standard error control methods, where after each transmission, the transmitter receives a feedback about the reception status of the packet as acknowledgement/negative-acknowledgement (ACK/NACK) [7]. The transmitter keeps retransmitting each packet until it receives an ACK or reaches the maximum allowed number of retransmissions. The ARQ protocols use only the last received version of a packet for decoding, whereas the hybrid ARQ (HARQ) protocols use all received versions, thereby increasing the probability of successfully decoding the packet [7], [8].

In this article, we consider a multi-source HARQ-based status update system, where the sources are connected to a transmitter that sends status update packets to a receiver over an unreliable wireless channel (see Fig. 1). We assume a slotted communication, in which the transmitter can send no more than a single packet within each slot. The sources, which monitor some time-varying random processes, are classified into two categories based on their sampling processes: 1) random arrival sources (i.e., uncontrollable sampling) which generate status update packets according to a Bernoulli process, and 2) generate-at-will sources (i.e., controllable sampling) which can be commanded to generate a status update packets at any slot. The generate-at-will is a model for status update systems where the sampling process can be controlled [9], [10], [11], [12], [13], [14]. Our considered system may represent a multi-source node such as a multi-sensor IoT device that is equipped with a single transmitter to communicate all the different sensed data to a
remote location through a wireless channel. In such scenario, the transmitter has control over the sampling of some sources, e.g., on-demand requesting of samples from a temperature or moisture sensor. On the contrary, the sampling of some sources depends on other grounds, such as energy to generate a sample (e.g., the source needs to harvest energy) or time to generate a sample (e.g., the source needs to scan an area which takes a random amount of time), leading to generating packets at random times. We also consider that each random arrival source has a buffer to store the latest (randomly) generated packet. However, it is important to note that because generate-at-will sources can produce a new packet at any time, having a buffer to store the last generated packet may not be optimal for minimizing AoI. Furthermore, in order to benefit from the HARQ, the transmitter has a memory to store the last transmitted but not successfully decoded packet of each source as this packet has a higher chance of being decoded than a new packet.

Apart from freshness requirements, the radio resources (e.g., power and channel utilization) also play an essential role in the operation of status update systems [15]. Hence, we investigate the problem of minimizing the average number of transmissions subject to the average AoI constraint. The solution of the problem determines the transmission status at each time slot: transmit a fresh packet from a source, retransmit the previously transmitted but not successfully decoded packet from a source, or stay idle.

A scenario where the controller knows the probabilities of possible outcomes after making a decision in a system is referred to as a known environment [16]. In our system, the known environment corresponds to the case where the transmitter knows the packet arrival rates of the random arrival sources and the probability of successful decoding after each transmission attempt. However, in some applications, the known environment may not be accessible, which motivates us to investigate the problem in both known and unknown environments. We propose three solutions to the problem, namely, a (stationary) deterministic transmission policy and a low-complexity dynamic transmission (LC-DT) policy for the known environment, and a learning-based transmission policy for the unknown environment. To obtain the deterministic transmission policy in the known environment, we cast the problem as a constrained Markov decision process (CMDP) problem, which is then transformed into an MDP problem using the Lagrangian relaxation. In general, an optimal policy for the CMDP problem is a randomized mixture of two deterministic policies, where one deterministic policy is feasible (i.e., satisfies the constraint) and the other policy is infeasible [17] (also, see recent applications [11, 12, 18]). However, since obtaining such randomized policy is often computationally intractable, we propose a near-optimal practical deterministic transmission policy (feasible deterministic policy) using the relative value iteration algorithm (RVIA) and the bisection algorithm. As a byproduct, this also gives a lower bound (infeasible deterministic policy) for benchmarking purposes. Since the number of states to explore in RVIA increases exponentially in the number of sources, and RVIA is run at each bisection iteration, obtaining the deterministic transmission policy is computationally inefficient. Therefore, we propose the LC-DT policy by using the drift-plus-penalty (DPP) method [19]. The DPP method transforms the average AoI constraint into a queue stability constraint, and subsequently, the (main) time average problem is transformed into an optimization problem that is to be solved at each slot. To devise the learning-based transmission policy, we use the DPP method to transform the CMDP problem into an MDP problem which aims at minimizing a time average DPP function; the policy is developed by solving the MDP problem with a deep Q-learning (DQL) algorithm [20]. In the numerical experiments, we analyze the effectiveness and complexity of the proposed policies and study the impact of employing HARQ in the system.

The main contributions of our paper are summarized as follows:

- We consider a multi-source HARQ-based status update system that consists of random arrival and generate-at-will sources. We minimize the average number of transmissions under the average AoI constraint in the known and unknown environment.
- For the known environment, we develop 1) a deterministic transmission policy using the Lagrangian relaxation, RVIA, and bisection, and 2) a low-complexity dynamic transmission policy using the DPP method.
- For the unknown environment, we develop a learning-based transmission policy by using the DPP method to cast the main problem as an MDP problem, which is then solved by applying DQL.
- The numerical results demonstrate the near-optimal performance of the proposed transmission policies compared to the lower bound policy and significant improvement with respect to a baseline policy. The results corroborate that HARQ improves the performance of the status update system and illustrate that the learning-based transmission policy performs close to the policies developed for the known environment.

A. Related Work

AoI characterization has extensively been studied from the perspective of queueing theory; see, e.g., [21, 22, 23, 24, 25, 26] and the references therein. One of the earliest studies to
analyze AoI under an HARQ protocol is [23], where the authors derived the closed-form expression of the average AoI for an HARQ-based M/G/1/1 queueing system. Building on top of the model and the AoI result from [23], the work [24] studied the age-optimal redundancy allocation problem under a constraint on the decoding error probability for chase combining and incremental redundancy HARQ protocols. An M/M/1 queueing system with network-code-HARQ protocol is considered in [25], where the closed-form expression of AoI is derived. The authors in [26] proposed several scheduling policies for various multiple access techniques in an energy harvesting powered status update system and derived a closed-form expression for the peak AoI under each policy.

Besides the analysis, the AoI has been studied in the retransmission-based status update systems from the perspective of sampling and transmission policies [11], [12], [13], [27], [28], [29], [30], [31]. In [13], the authors considered a multi-source and generate-at-will-based status update system and minimized the average AoI by proposing a source selection policy under three pre-defined transmission policies. In [27], the authors derived the closed-form expression of the average AoI in an HARQ-based status update system in which two energy harvesting sources send the same information for providing diversity at the destination. The work [28] considered a multi-source status update system in which the transmitter harvests energy and uses a greedy retransmission policy. They minimized the average AoI by determining a set of transmission times and choosing a source to send status update packet. An HARQ-based non-orthogonal multiple access system with two users are considered in [29], where the average AoI is minimized by determining the transmit power and transmission status, i.e., transmitting a new packet or retransmitting the previously transmitted but not successfully decoded packet, at each slot. In [30], the authors investigated the average AoI minimization problem in a status update system with a pre-defined retransmission policy to find the times for updating the destination. The work [31] studied the average AoI minimization problem in an HARQ-based status update system. They calculated the probability of decoding failure through an erasure channel and developed a threshold-based transmission policy that decides between the transmission of a new packet and the retransmission of the previously transmitted one.

The most related works to our paper are [11], [12]. The work [11] considered a similar HARQ-based status update system to ours, yet with the following differences. The authors in [11] considered a single generate-at-will source, while we consider both random arrival and generate-at-will sources as a multi-source system. Considering the random arrival sources makes the system more complicated, as the transmitter does not know the availability of the fresh packets at the subsequent slots. We study the problem of minimizing the average number of transmissions subject to the average AoI constraint, while they studied the average AoI minimization problem subject to the average number of transmissions constraint. Similarly as in [11], we use the CMDP approach along with the Lagrangian relaxation to solve the problem in the known environment; however, we also propose the low-complexity Lyapunov-based dynamic transmission policy. Furthermore, for the unknown environment, they proposed a learning-based transmission policy by the Lagrangian relaxation which involves running the learning procedure for several Lagrangian multipliers. Differently, our learning-based transmission policy utilizes the Lyapunov optimization theory, and thus, the learning procedure needs to be run only once. In [12], which is an extension of [11], the authors considered an HARQ-based status update system that contains one generate-at-will source and several users (destinations), in which at most one user is served at each slot. They constructed a CMDP problem for minimizing the weighted average AoI subject to the average number of transmissions constraint. They solved the CMDP problem with the Lagrangian relaxation for the known environment; for the unknown environment, they proposed different learning-based transmission policies by the Lagrangian relaxation.

B. Organization

The rest of this article is organized as follows. The system model and problem formulation are presented in Section II. The CMDP formulation and the corresponding solution are presented in Section III. In Section IV, we present the LC-DT policy and the learning-based transmission policy. Numerical results are presented in Section V. Finally, concluding remarks are made in Section VI.

II. SYSTEM MODEL AND PROBLEM FORMULATION

A. System Model

We consider a multi-source status update system that consists of $K$ sources, one transmitter, and one receiver, as depicted in Fig. 1. The sources and the transmitter are integrated modules, therefore, the communication between them is assumed instantaneous and error-free. The receiver is interested in timely information about different random processes monitored by the $K$ sources. The transmitter sends status update packets to the receiver through an error-prone wireless channel with the aid of an HARQ protocol. The system operates in discrete time with unit time slots $t \in \{1, 2, \dots \}$.

The $K$ sources are divided into two classes based on their sampling processes: 1) a set $I$ of $I$ random arrival sources whose sampling processes are uncontrollable and 2) a set $J$ of $J$ generate-at-will sources, where the transmitter can sample the process at any time. Each source $k \in I$ generates status update packets randomly and independently at the beginning of slots according to a Bernoulli random process with parameter $\lambda_k$. We denote the set of all sources by $K = I \cup J = \{1, \ldots, K\}$, where $K = I + J$.

Each random arrival source has a buffer of size one to store the last generated packet. As long as a new packet does not arrive, the buffer keeps the last generated packet. The transmitter has a memory of size $K$ packets to store the previously transmitted but

1Each status update packet contains a timestamp representing the time when the sample was generated and the measured value of the monitored process.

2The random arrival source is a commonly used model for uncontrollable sensors that performs measurements at random times [11], [2], [3], [21], [22], [23].
not successfully decoded packets of each source. Note that, after a number of unsuccessful transmission attempts of a packet from a source, the transmitter may decide to transmit a packet from the other sources. In this case, the transmitter retains the previously transmitted but not successfully decoded packet of each source in the transmitter’s memory for possible future retransmissions, since this packet is more likely to be decoded than a new packet from that source due to the HARQ protocol. We term a packet in the transmitter’s memory an under-process packet. Thus, the maximum number of packets stored in the system is \( I + K \) packets, i.e., \( I \) packets at the buffers of random arrival sources and \( K \) packets at the transmitter’s memory.

We assume that the transmitter\(^3\) can transmit at most one packet per slot. At each slot, the transmitter decides whether to send a packet or stay idle. The possible transmission options for a random arrival source \( k \in \mathcal{I} \) are either transmitting the packet from its buffer or retransmitting the under-process packet from the transmitter’s memory. The possible transmission options for a generate-at-will source \( k \in \mathcal{J} \) are either generating a new sample or retransmitting the under-process packet from the transmitter’s memory. We refer to the packets in the buffers of the random arrival sources and to the newly generated packets of the generate-at-will sources as fresh packets. If the transmitter decides to transmit a fresh packet from a given source, this packet replaces the source’s under-process packet in the transmitter’s memory.

1) Transmission Model: At each slot \( t \), the transmitter takes one of the following actions: 1) transmit a fresh packet from a source, 2) retransmit an under-process packet of a source, or 3) stay idle. Let \( u_{t,k} \in \{0, 1\} \) denote the decision variable about transmitting a fresh packet from source \( k \) at slot \( t \), where \( u_{t,k} = 1 \) indicates that the transmitter sends the fresh packet, and \( u_{t,k} = 0 \) otherwise. Let \( r_{t,k} \in \{0, 1\} \) denote the decision variable about retransmitting the under-process packet of source \( k \) at slot \( t \), where \( r_{t,k} = 1 \) indicates that the transmitter sends the under-process packet, and \( r_{t,k} = 0 \) otherwise. Since the transmitter can transmit at most one packet per slot, we have \( \sum_{k \in \mathcal{K}} u_{t,k} + r_{t,k} \leq 1 \).

HARQ protocol: In the considered HARQ protocol, every packet transmission attempt is followed by an instantaneous error-free ACK/NACK feedback signal from the receiver. Let \( b_{t,k} \in \{0, 1\} \) denote the packet reception status at slot \( t \), where \( b_{t,k} = 1 \) indicates that the transmitted packet was decoded successfully (ACK), and \( b_{t,k} = 0 \) indicates that either the transmitted packet was not decoded successfully (NACK) or the transmitter remained idle. In the HARQ protocol, the receiver uses all previously received versions of a packet to decode it. Therefore, the probability of successfully decoding a packet is an increasing function of the number of attempted transmissions of the packet. Let \( x_{t,k} \) denote the number of attempted transmissions of a packet of source \( k \) up to slot \( t \). The evolution of \( x_{t,k} \) is given as

\[
x_{t+1,k} = \begin{cases} 1 & u_{t,k} = 1, r_{t,k} = 0, \\ x_{t,k} + 1 & r_{t,k} = 1. 
\end{cases}
\]

To account for the fact that most practical HARQ protocols allow only a finite number of retransmissions, we limit the number of transmission attempts of a packet to \( x_{\text{max}} \) i.e., \( x_{t+1,k} \leq x_{\text{max}} \). The function representing the probability of successful decoding after \( x_{t+1,k} \) transmissions is denoted by \( f(x_{t+1,k}) \). In practice, \( f(\cdot) \) is a complicated function of several parameters such as the channel conditions, the channel coding methods, and the combining technique utilized in the HARQ protocol [32, 33, 34, 35].

2) Age of Information: The AoI is defined as the time elapsed since the generation of the most recently received status update packet at a destination. Let \( \delta_{t,k} \) denote the AoI of source \( k \) at the receiver at slot \( t \); we refer to this simply as the AoI of source \( k \) hereinafter. We use the common assumption (see, e.g., [11], [12], [18], [36], [37]) that all AoI values in the system are upper bounded by \( \delta_{\text{max}} \). Besides making the analysis tractable, this supports the fact that an AoI value exceeding a high enough upper bound carries the same timeliness information as the upper bound for the destination’s decision making (e.g., control actions for a drone). To characterize the AoI of each source, we need to define the age of a fresh packet at a source and the age of an under-process packet in the transmitter’s memory. These are defined in the following.

Age of the fresh packets: Let \( \delta^f_{t,k} \) denote the age of the fresh packet of source \( k \) at slot \( t \). For a random arrival source, if a packet arrives at the buffer at the beginning of slot \( t \), the age of the fresh packet becomes zero, otherwise it is incremented by one. Let \( b_{t,k} \in \{0, 1\} \) denote the packet arrival status of source \( k \in \mathcal{I} \) at slot \( t \), where \( b_{t,k} = 1 \) indicates a packet arrives at the buffer, and \( b_{t,k} = 0 \) otherwise. Let \( \text{Pr}(b_{t,k} = 1) = \lambda_k \). For the generate-at-will sources, the transmitter can generate a fresh packet at any time so that the age of the fresh packet is always zero. Thus, the evolution of \( \delta^f_{t,k} \) with the initial value \( \delta^f_{0,k} = 0 \) is given as

\[
\delta^f_{t,k} = \begin{cases} 0 & b_{t,k} = 1, \quad k \in \mathcal{I} \\ \min\{\delta^f_{t-1,k} + 1, \delta_{\text{max}}\} & b_{t,k} = 0, \quad k \in \mathcal{I} \\ \min\{\delta^f_{t,k}, \delta_{\text{max}}\} & k \in \mathcal{J}, \end{cases}
\]

(2)

Age of the under-process packets: Let \( \delta^p_{t,k} \) denote the age of the under-process packet of source \( k \) at slot \( t \). If the transmitter sends a fresh packet of source \( k \) at slot \( t \), the age of the under-process packet of the source at the next slot drops to \( \min\{\delta^f_{t,k} + 1, \delta_{\text{max}}\} \). In other cases (i.e., retransmission or staying idle), the age of the under-process packet is incremented by one. The evolution of \( \delta^p_{t,k} \) with the initial value \( \delta^p_{0,k} = 0 \) is given by

\[
\delta^p_{t+1,k} = \begin{cases} \min\{\delta^p_{t,k} + 1, \delta_{\text{max}}\} & u_{t,k} = 1, \\ \min\{\delta^p_{t,k} + 1, \delta_{\text{max}}\} & \text{otherwise}. 
\end{cases}
\]

(3)

AoI at the receiver: Having defined \( \delta^f_{t,k} \) and \( \delta^p_{t,k} \), we now characterize the evolution of the AoI at the receiver. If the transmitter sends a fresh packet of source \( k \) at slot \( t \) (i.e., \( u_{t,k} = 1 \) and the packet is decoded successfully at the receiver (i.e., \( d_{t} = 1 \)), the AoI of the source at the next slot drops to \( \min\{\delta^f_{t,k} + 1, \delta_{\text{max}}\} \), otherwise (i.e., \( d_{t} = 0 \)), the AoI increases by one. If the transmitter retransmits the under-process packet of source \( k \) (i.e., \( r_{t,k} = 1 \) and it is decoded successfully at the receiver, the AoI of the source at the next slot drops to \( \min\{\delta^p_{t,k} + 1, \delta_{\text{max}}\} \), otherwise (i.e., \( d_{t} = 0 \)), the AoI increases by one. If, at slot \( t \), the transmitter does not transmit any packet...
of source $k$ (i.e., $u_{t,k} + r_{t,k} = 0$), the AoI of the source at the next slot increases by one. The evolution of $\delta_{t,k}$ with the initial value $\delta_{0,k} = 0$ is given as

$$
\delta_{t+1,k} = \begin{cases} 
\min \{\delta_{t,k}^i + 1, \delta_{\text{max}}\}, & u_{t,k}d_k = 1 \\
\min \{\delta_{t,k}^p + 1, \delta_{\text{max}}\}, & r_{t,k}d_k = 1 \\
\min \{\delta_{t,k}^d + 1, \delta_{\text{max}}\}, & u_{t,k}(1 - d_k) = 1 \\
\min \{\delta_{t,k}^l + 1, \delta_{\text{max}}\}, & r_{t,k}(1 - d_k) = 1 \\
\min \{\delta_{t,k} + 1, \delta_{\text{max}}\}, & u_{t,k} + r_{t,k} = 0.
\end{cases}
$$

(4)

Note that the conditions in (4) are mutually exclusive and collectively exhaustive.

### B. Problem Formulation

Our main goal is to minimize the average number of transmissions subject to the average AoI constraint by finding a transmission policy that determines the transmission decision variables at each slot $t$, $\{u_{t,k}, r_{t,k}\}_{k \in \mathcal{K}}$. The transmission decision for slot $t$ is based on the age of the fresh packets, $\delta_{t,k}^f$, the age of the under-process packets, $\delta_{t,k}^o$, the AoI of each source, $\delta_{t,k}$, and the number of previous transmission attempts of each under-process packet, $x_{t,k}$.

Let $\tau_t \in \{0, 1\}$ denote the transmission status at slot $t$, where $\tau_t = 1$ indicates that the transmitter sends a packet, and $\tau_t = 0$ otherwise. Thus, we have

$$
\tau_t = \begin{cases} 
1 & \sum_{k \in \mathcal{K}} u_{t,k} + r_{t,k} = 1 \\
0 & \text{otherwise}
\end{cases}
$$

(5)

Let $\bar{\tau}$ denote the expected long-term average number of transmissions, defined as

$$
\bar{\tau} = \limsup_{T \to \infty} \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}\{\tau_t\},
$$

(6)

where $\mathbb{E}\{\cdot\}$ is the expectation with respect to the randomness of the system (i.e., packet arrival processes of the random arrival sources and randomness in the communication channel) and the decision variables $\{u_{t,k}, r_{t,k}\}_{k \in \mathcal{K}}$. Finally, let $\bar{\delta}$ denote the expected long-term average AoI of $\delta_t$, given as

$$
\bar{\delta} = \limsup_{T \to \infty} \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}\{\delta_t\},
$$

(7)

Using (6) and (7), the main problem of this paper is formulated as the following stochastic optimization problem:

minimize $\bar{\tau}$

subject to $\bar{\delta} \leq \delta_{\text{max}}$

$x_{t+1,k} \leq x_{\text{max}}, k \in \mathcal{K}, t \in \mathbb{N}$

$\sum_{k \in \mathcal{K}} u_{t,k} + r_{t,k} \leq 1, t \in \mathbb{N}$

(9a)

(9b)

(9c)

(9d)

(9e)

with variables $\{u_{t,k}, r_{t,k}\}_{k \in \mathcal{K}}$ for all $t \in \mathbb{N}$, where $\delta_{\text{max}}$ is the maximum allowed average AoI. The constraints of problem (9) are as follows. Inequality (9b) ensures the average AoI constraint. Inequality (9c) ensures that the number of transmission attempts for each packet cannot exceed $x_{\text{max}}$. Inequality (9d) ensures that the transmitter can transmit at most one packet per slot. Expression (9e) indicates the binary nature of the decision variables.

### III. Deterministic Transmission Policy

In this section, we propose a (near-optimal) solution to main problem (9) for the known environment, i.e., the packet arrival probability of each random arrival source, $\lambda_k$, and the probability of successful decoding function, $f(\cdot)$, are known. We cast problem (9) as a constrained Markov decision process (CMDP) problem. Then, we use the Lagrangian relaxation to find a near-optimal deterministic transmission policy.

### A. CMDP Formulation

The CMDP is defined by a tuple of five elements $(\mathcal{S}, \mathcal{A}, \mathcal{P}, C, D)$: state space, action space, state transition probabilities, and two cost functions, which are defined next.

**State:** Let $s_{t,k} = \{\delta_{t,k}^f, \delta_{t,k}^o, \delta_{t,k}, x_{t,k}\}$ denote the state of source $k$ at slot $t$. The system state at slot $t$ is defined as $s_t = \{s_{t,k}\}_{k \in \mathcal{K}} \in \mathcal{S}$, where $\mathcal{S}$ is the state space. The initial state is denoted with $s_0 = \{s_{0,k}\}_{k \in \mathcal{K}}$, where $s_{0,k} = \{0, 0, 0, 0\}$ for all $k \in \mathcal{K}$.

**Action:** Let $a_t = \{a_{t,k}\}_{k \in \mathcal{K}} \in \mathcal{A}$ denote the action of the transmitter at slot $t$, where $a_{t,k} = \{u_{t,k}, r_{t,k}\}$ represents the action for source $k$, and $\mathcal{A}_s$ is a space of feasible actions in state $s_t$, defined as $\mathcal{A}_s = \{u_{t,k}, r_{t,k} \in \{0, 1\} | k \in \mathcal{K}, \sum_{k \in \mathcal{K}} u_{t,k} + r_{t,k} \leq 1, r_{t,k}(x_{t,k} + 1) \leq \delta_{\text{max}}\}$.

**Cost functions:** The CMDP has two cost functions: 1) transmission cost, defined as $C(a_t) = \tau_t$, i.e., $C(a_t) = 1$ if the transmitter makes a transmission attempt at slot $t$, otherwise $C(a_t) = 0$, and 2) AoI cost, defined as $D(s_t) = \delta_t$, i.e., the average AoI over sources at slot $t$

**State transition probabilities:** Let $\mathcal{P}(s' | s, a) = \Pr(s' = s_{t+1} | s_t = s, a = a_t)$ denote the state transition probabilities, defined as the probability of moving from current state $s = s_t$ to a next state $s' = s_{t+1}$ under action $a = a_t$. Given an action, the one-slot evolution of the AoI values (at the source, memory, and destination) and of the number of transmissions of the under-process packets is independent among the sources. Therefore, the state transition probability factors as $\mathcal{P}(s' | s, a) = \prod_{k \in \mathcal{K}} \Pr(s_{t+1,k} | s_{t,k}, a_{t,k})$. Let us denote $\delta_{t,k}^f \triangleq \min\{\delta_{t,k}^f + 1, \delta_{\text{max}}\}$, $\delta_{t,k}^o \triangleq \min\{\delta_{t,k}^o + 1, \delta_{\text{max}}\}$, $\delta_{t,k} \triangleq \min\{\delta_{t,k} + 1, \delta_{\text{max}}\}$, $\tilde{f}(\cdot) \triangleq 1 - f(\cdot)$, and $\lambda_k \triangleq 1 - \lambda_k$. Given the state $s_{t,k} = \{\delta_{t,k}^f, \delta_{t,k}^o, \delta_{t,k}, x_{t,k}\}$, the state transition probabilities for a random arrival source $k \in \mathcal{K}$ under different actions can be expressed as

$$
\Pr\left(\{0, \delta_{t,k}^f, \delta_{t,k}^o, 1\} \mid s_{k}, a_{k} = \{1, 0\}\right) = f(1) \lambda_k
$$

(10a)

$$
\Pr\left(\{\delta_{t,k}^f, \delta_{t,k}, \delta_{t,k}^o, 1\} \mid s_{k}, a_{k} = \{1, 0\}\right) = f(1) \tilde{\lambda}_k
$$

(10b)

$$
\Pr\left(\{0, \delta_{t,k}^f, \delta_{t,k}, 1\} \mid s_{k}, a_{k} = \{1, 0\}\right) = \tilde{f}(1) \lambda_k
$$

(10c)

$$
\Pr\left(\{\delta_{t,k}^f, \delta_{t,k}, \delta_{t,k}, 1\} \mid s_{k}, a_{k} = \{1, 0\}\right) = \tilde{f}(1) \tilde{\lambda}_k
$$

(10d)

$$
\Pr\left(\{0, \delta_{t,k}^f, \delta_{t,k}, x_{k} + 1\} \mid s_{k}, a_{k} = \{0, 1\}\right)
$$

(10e)
problem, the MDP problem has only one cost function that is defined as \( L(s, a, \beta) = C(a_t) + \beta(D(s_t) - \Delta_{\text{max}}) \), whereas the other elements, i.e., the state space, action space, and state transition probabilities, are the same. Let \( \bar{L}(\pi, \beta) = \lim_{T \to \infty} \frac{1}{T} \sum_{t=1}^{T} E(C(a_t) + \beta(D(s_t) - \Delta_{\text{max}})) \) denote the Lagrangian corresponding to CMDP problem (11), where \( \beta \) is the Lagrangian multiplier. Following the standard Lagrangian relaxation procedure, we restrict to the set of deterministic policies and construct the following MDP problem associated with the CMDP problem (11)

\[
\text{minimize}_{\pi \in \Pi_D} \bar{L}(\pi, \beta),
\]

where \( \Pi_D \) is the set of all deterministic policies. Let \( \eta^*_\beta \) denote an optimal policy that solves problem (12) for a given \( \beta \), which is called a \( \beta \)-optimal policy.

The following remark expresses the relation between the optimal values of CMDP problem (11) and MDP problem (12).

**Remark 1:** The cost function in the objective of CMDP problem (11) is bounded below, i.e., \( C(a_t) \geq 0 \) for all \( t \in \mathbb{N} \). Moreover, the state space, \( S \), is finite. Therefore, the two conditions in [40, Corollary 12.2] are satisfied in our CMDP formulation, and we have

\[
\tau^* = \sup_{\beta > 0} \min_{\pi \in \Pi_D} \bar{L}(\pi, \beta),
\]

According to Remark 1, the optimal CMDP value (11), \( \tau^* \), is obtained via the solution of the right-hand side of (13), which means finding the optimal Lagrangian multiplier \( \beta^* \) and its corresponding \( \beta^* \)-optimal policy, \( \eta^*_\beta \). If policy \( \eta^*_\beta \) satisfies the constraint of CMDP problem (11) with equality, i.e., \( \delta^* = \Delta_{\text{max}} \), then \( \tau^* \) is an optimal policy for the CMDP problem, i.e., \( \tau^* = \pi^*_\beta \). However, due to the discrete nature of the action space, in general, there is no guarantee that \( \tau^* \) satisfies the constraint with equality. To elaborate this, the following remark presents the structure of an optimal policy \( \tau^* \).

**Remark 2:** An optimal policy for CMDP problem (11), \( \tau^* \), is a randomized mixture of two deterministic \( \beta \)-optimal policies, from which one policy satisfies the constraint and the other one violates it. The two policies are mixed with a randomization factor such that the obtained optimal policy satisfies \( \delta^* = \Delta_{\text{max}} \) [11], [17], [18].

According to Remark 2, two deterministic \( \beta \)-optimal policies and the optimal randomization factor to mix between these policies should be found to obtain an optimal policy, \( \tau^* \). However, finding these becomes readily computationally intractable even for the moderate number of states, especially because of its combinatorial nature, the optimal randomization factor can be found only numerically [41, Sec. 3.2]. Therefore, in order to solve CMDP problem (11), we propose a practical deterministic policy, which is numerically shown to provide near-optimal performance in Section V. More specifically, we develop an iterative algorithm based on bisection and the relative value iteration algorithm (RVIA), as summarized in Algorithm 1. In brief, at each iteration, we find a \( \beta \)-optimal policy for a given \( \beta \) via the RVIA and subsequently update \( \beta \) according to the bisection rule. The iterative procedure continues until the best \( \beta \)-optimal policy among the feasible \( \beta \)-optimal policies is found. In the next two subsections, we delve into details of this procedure.

\[\text{It is worth noting that since the average cost functions of main problem (9) are expressed as expected time average values, their instantaneous values can be directly used as the immediate costs of the CMDP problem.}\]
1) Algorithm to Find a \( \beta \)-Optimal Policy: To obtain an optimal policy \( \pi_\beta \) for a given \( \beta \), we solve the MDP problem (12) via RVIA. By [42, Th. 8.4.3], there exists a relative value function \( h(s), s \in S \), that satisfies

\[
\tilde{L}^*(\beta) + h(s) = \min_{a \in A_s} \left[ L(s, a, \beta) + \sum_{s' \in S} \Pr(s' | s, a) h(s') \right],
\]

where \( \tilde{L}^*(\beta) \) is the optimal value of the MDP problem (12) for a given \( \beta \), defined as \( \tilde{L}^*(\beta) = \min_{\pi \in \Pi_\beta} \tilde{L}(\pi, \beta) \). Subsequently, the \( \beta \)-optimal policy, \( \pi_\beta \), is obtained as [42, Th. 8.4.4]

\[
\pi_\beta(s) = \arg \min_{a \in A_s} \left[ L(s, a, \beta) + \sum_{s' \in S} \Pr(s' | s, a) h(s') \right].
\]

To obtain the \( \beta \)-optimal policy, we use the RVIA, in which the relative value function for all states \( s \in S \) at each iteration \( i \in \{0, 1, \ldots \} \) is updated as \( h^i(s) = v^i(s) - v^i(s^{\text{ref}}) \). Where \( s^{\text{ref}} \in S \) is an arbitrary reference state which remains unchanged throughout the iterations. The term \( v^i(s) \), called value function, is obtained at each iteration as \( v^i(s) = \min_{a \in A_s} \left[ L(s, a, \beta) + \sum_{s' \in S} \Pr(s' | s, a) h^{i-1}(s') \right] \). For any state \( s \in S \) and initialization \( v^0(s) \), the sequences \( \{h^i(s)\}_{i=1,2,\ldots} \) and \( \{v^i(s)\}_{i=1,2,\ldots} \) converge, i.e., \( \lim_{i \to \infty} h^i(s) = h(s) \) and \( \lim_{i \to \infty} v^i(s) = v(s) \). The RVIA algorithm to find a \( \beta \)-optimal policy is presented in Steps 3–12 of Algorithm 1. After the convergence of RVIA, i.e., convergence of the relative value function, \( h(\cdot) \), and the value function, \( v(\cdot) \) (see Steps 3–9 in Algorithm 1), we obtain the \( \beta \)-optimal policy, \( \pi_\beta \), according to (14) (see Steps 10–12 in Algorithm 1). It is worth noting that the optimal value of the MDP problem (12) for a given \( \beta \) is given by \( \tilde{L}^*(\beta) = v(s^{\text{ref}}) \).

2) Algorithm to Find the Optimal Lagrangian Multiplier: According to [17, Lemma 3.1], for a given \( \beta \)-optimal policy \( (\pi_\beta) \), the objective function of the CMDP problem, \( \tilde{L}^{\ast \beta} \), and the objective function of the MDP problem, \( \tilde{L}^*(\beta) \), are increasing in \( \beta \), while the constraint of the CMDP problem, \( \delta^{\ast \beta} \), is decreasing in \( \beta \). Therefore, we are interested in the smallest Lagrangian multiplier that satisfies the constraint in CMDP problem (11), defined as

\[
\tilde{\beta} \triangleq \inf \left\{ \beta \geq 0 \mid \delta^{\ast \beta} \leq \Delta^{\max} \right\}. \tag{15}
\]

To search for \( \tilde{\beta} \), we use the bisection algorithm which takes advantage of the monotonicity of \( \delta^{\ast \beta} \) with respect to \( \beta \), as presented in Algorithm 1 (see Steps 1–18). We initialize the bisection algorithm with \( \beta_u \) and \( \beta_i \) in such a way that \( \delta^{\ast \beta_u} \leq \Delta^{\max} \) and \( \delta^{\ast \beta_i} \geq \Delta^{\max} \), which also implies \( \beta_u \leq \beta_i \). The algorithm termination criterion is \( \beta_u - \beta_i < \kappa \), where \( \kappa \) is a sufficiently small constant. After termination of the bisection algorithm, we set \( \tilde{\beta} = \beta_u \), and obtain the most feasible \( \beta \)-optimal policy as \( \pi^{\ast \beta} = \pi^{\beta_u} \). Moreover, the algorithm returns the infeasible policy associated with \( \beta_i \), which represents a lower bound to an optimal solution of (11).

IV. LYAPUNOV-BASED TRANSMISSION POLICIES

In this section, we use the Lyapunov optimization theory to derive a solution for problem (9). While we developed the deterministic transmission policy for the known environment in Section III (Algorithm 1), finding the policy has relatively high complexity. Namely, RVIA needs to explore all states and actions. When the number of sources increases, the number of states increases exponentially. Besides this, the RVIA is run for each bisection iteration. These make obtaining the policy computationally inefficient. Therefore, we develop a low-complexity dynamic transmission (LC-DT) policy using the DPP method for the known environment in Section IV-A. The numerical results in Section V show that LC-DT policy performs close to the optimal solution.

Furthermore, we develop a Lyapunov-based transmission policy for the unknown environment in Section IV-B. We transform CMDP problem (11) into an MDP problem using the DPP method. Then, by using deep Q-learning (DQL), we solve the MDP problem and provide the learning-based transmission policy. It is worth noting that, alternatively, the transformation of CMDP problem (11) into an MDP problem could be done via the Lagrangian relaxation method. However, using the Lagrangian method, we need to run the DQL for several Lagrangian

---

**Algorithm 1. The Deterministic Transmission Policy to Solve CMDP Problem (11).**

**Input:** 1) System parameters: \( \Delta^{\max}, f(\cdot), \) and \( \lambda_k \) for all \( k \in K \), 2) RVIAs: \( s^{\text{ref}}, \epsilon, \) and 3) Bisection parameters: \( \beta_u, \beta_i, \kappa \)

// Bisection Algorithm

1. While \( \beta_u - \beta_i \geq \kappa \)
   2. \( \tilde{\beta} = \frac{\beta_u + \beta_i}{2} \)
   3. **RVIA** for the given \( \tilde{\beta} 
   4. Initialize: 1) \( \beta = 1 \), 2) set \( h^0(s) = 1, h_1(s) = 0, v^0(s) = 0 \) for all \( s \in S \)
   5. While \( \max_{s \in S} [h^i(s) - h^{i-1}(s)] \geq \epsilon \)
      6. \( i = i + 1 \)
      7. For \( s \in S \)
         8. \( v^i(s) = \min_{a \in A_s} \left[ L(s, a, \tilde{\beta}) + \sum_{s' \in S} \Pr(s' | s, a) h^{i-1}(s') \right] \)
         9. \( h^i(s) = v^i(s) - v^{i-1}(s) \)
      10. End
   11. End
   12. If \( \delta^{\ast \tilde{\beta}} \leq \Delta^{\max} \) then
      13. \( \beta_u = \tilde{\beta} \)
      14. Else
         15. \( \beta_i = \tilde{\beta} \)
      16. End
   17. End

**Output:** Lagrangian multiplier: \( \tilde{\beta} = \beta_u \), feasible policy: \( \pi^{\ast \beta} = \pi^{\beta_u} \), (infeasible) lower bound policy: \( \pi^{\beta_i} \)
multipliers, whereas via the DPP method, we need to run DQL only once.

A. Low-Complexity Dynamic Transmission Policy: Known Environment

We use the Lyapunov drift-plus-penalty (DPP) method [19], where average AoI constraint (9b) is enforced by transforming it into a queue stability constraint. In particular, the constraint is modeled by a virtual queue so that the stability of the virtual queue implies the feasibility of the constraint.

Let \( Q_t \) denote the virtual queue associated with average AoI constraint (9b) at slot \( t \). The virtual queue evolves as follows:

\[
Q_{t+1} = \max\{Q_t - \Delta_{t+1} + \delta_{t+1}, 0\}. \tag{16}
\]

To ensure that average AoI constraint (9b) is satisfied, we use the notion of strong stability, where the virtual queue is (strongly) stable if \( \lim_{T \to \infty} \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}\{Q_t\} < \infty \) [19, Ch. 2]. The satisfying of the average AoI constraint (9b) is guaranteed by the strong stability of queue (16) [19, Ch. 4]. Note that in the virtual queue (16), \( \delta_{t+1} \) plays the role of arrival rate, and \( \Delta_{t+1} \) represents the service rate. Thus, the strong stability of queue (16) implies that its expected long-term average arrival rate (given by (7)) must be no larger than the service rate \( \Delta_{t+1} \), which is precisely the average AoI constraint (9b). We refer the interested reader to [19, Ch. 4] for more details.

To impose the queue stability condition, we introduce the Lyapunov function and its drift. A quadratic Lyapunov function is defined as \( L(Q_t) = \frac{1}{2} Q_t^2 \) [19, Ch. 3]. The Lyapunov function measures the network congestion, and thus, by minimizing the expected change of the Lyapunov function from one slot to the next slot, the virtual queue can be stabilized [19, Ch. 4].

Let \( \xi_t = \{\{\delta_{t,k}, \delta_{t,k}^p, \delta_{t,k}, x_{t,k}\}_{k \in K}, Q_t\} \) denote the network state at slot \( t \). The conditional Lyapunov drift, \( \alpha(\xi_t) \), is defined as the expected change in the Lyapunov function over one slot, given the network state at slot \( t \), i.e., \( \alpha(\xi_t) = \mathbb{E}\{L(Q_{t+1}) - L(Q_t) | \xi_t\} \) [19, Ch. 4].

By following the DPP minimization approach [19, Ch. 3], a solution for (9) can be derived by solving the following problem at each slot \( t \):

\[
\begin{align*}
\text{minimize} & \quad \mathbb{E}\{\tau_t | \xi_t\} + \alpha(\xi_t) \tag{17a} \\
\text{subject to} & \quad x_{t+1,k} \leq x_{\max}, \quad k \in K \tag{17b} \\
& \quad \sum_{k \in K} u_{t,k} + r_{t,k} \leq 1 \tag{17c} \\
& \quad u_{t,k}, r_{t,k} \in \{0, 1\}, \quad k \in K, \tag{17d}
\end{align*}
\]

with variables \( \{u_{t,k}, r_{t,k}\}_{k \in K} \). The objective function of problem (17) represents the DPP function, in which the positive parameter \( V \) is used to adjust the tradeoff between minimizing the original objective function (9a) and the size of the virtual queue backlog. A larger value of \( V \) puts more emphasis on the original objective function, i.e., minimizing the average number of transmissions. According to [43, Sec. 4.1.2], by solving problem (17) at each slot, the virtual queue in (16) would be strongly stable. According to [43, Th. 2.8], if the virtual queue \( Q_t \) is strongly stable, it is also mean-rate stable, and consequently, we have \( \delta \leq \Delta_{\max} \).

According to the standard procedure used in the DPP method, an upper bound for the drift part is derived, whereas the penalty part (i.e., the original objective function) remains unchanged [19, Ch. 4]. We stress that using such upper bound of the conditional Lyapunov drift in the optimization procedure does not affect the virtual queue’s stabilizing logic. We derive the upper bound using the following inequality, where for any \( \gamma, \tilde{\gamma} \geq 0, \gamma \geq 0 \), and \( \tilde{\gamma} \geq 0 \), we have [36]

\[
(max\{\tilde{\gamma} - \gamma, 0\})^2 \leq \tilde{\gamma}^2 + \gamma^2 + \gamma^2 + 2\tilde{\gamma}(\gamma - \tilde{\gamma}). \tag{18}
\]

By applying (18) to (16), an upper bound for \( Q_{t+1}^2 \) is given as

\[
Q_{t+1}^2 \leq Q_t^2 + (\Delta_{\max})^2 + \delta_{t+1}^2 + 2Q_t \left( \delta_{t+1} - \Delta_{\max} \right). \tag{19}
\]

By applying (19) to (15), the upper bound of objective function (17a) is given as

\[
\begin{align*}
& \quad V \mathbb{E}\{\tau_t | \xi_t\} + \alpha(\xi_t) \leq V \sum_{k \in K} \mathbb{E}\{u_{t,k} | \xi_t\} + \frac{1}{2} \mathbb{E}\left((\Delta_{\max})^2 + \delta_{t+1}^2\right) + 2Q_t \left( \delta_{t+1} - \Delta_{\max} \right) \\
& \quad + \frac{1}{2} \left((\Delta_{\max})^2 + \mathbb{E}\{\delta_{t+1}^2 | \xi_t\}\right) + 2Q_t \left( \mathbb{E}\{\delta_{t+1} | \xi_t\} - \Delta_{\max} \right). \tag{20}
\end{align*}
\]

To complete the derivation of (20), we calculate \( \mathbb{E}\{\delta_{t+1} | \xi_t\} \) and \( \mathbb{E}\{\delta_{t+1}^2 | \xi_t\} \), which are given by the following lemmas.

**Lemma 1:** The conditional expectation \( \mathbb{E}\{\delta_{t+1} | \xi_t\} \) is given as

\[
\mathbb{E}\{\delta_{t+1} | \xi_t\} = \frac{1}{K} \sum_{k \in K} \mathbb{E}\{u_{t,k} | \xi_t\} f(1)\delta_{t,k}^p + \mathbb{E}\{r_{t,k} | \xi_t\} f(x_{t,k} + 1)\delta_{t,k}^p + [1 - f(1)\mathbb{E}\{u_{t,k} | \xi_t\}] f(x_{t,k} + 1)\mathbb{E}\{r_{t,k} | \xi_t\} \delta_{t,k}. \tag{21}
\]

**Proof:** The proof is presented in Appendix A.

**Lemma 2:** The conditional expectation \( \mathbb{E}\{\delta_{t+1}^2 | \xi_t\} \) is given as

\[
\mathbb{E}\{\delta_{t+1}^2 | \xi_t\} = \frac{1}{K} \sum_{k \in K} \mathbb{E}\{u_{t,k} | \xi_t\} f(1)\delta_{t,k}^p^2 + \mathbb{E}\{r_{t,k} | \xi_t\} f(x_{t,k} + 1)\delta_{t,k}^p^2 + [1 - f(1)\mathbb{E}\{u_{t,k} | \xi_t\}] f(x_{t,k} + 1)\delta_{t,k}^p \delta_{t,k}. \]

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
Proposed Low-Complexity Dynamic Transmission (LC-DT) Policy.

**Algorithm 2.** Proposed Low-Complexity Dynamic Transmission (LC-DT) Policy.

**Initialize:** Set $V$, and initialize $o_1$.

1. for $t = 1, 2, 3, \ldots$ do

2. Step 1: Update $\delta_{t,k}$ using (2).

3. Step 2: Find decision variables $u_{t,k}, r_{t,k}$ for all $k \in K$ by solving problem (24).

4. Step 3: Update $x_{t+1,k}$ using (1), $\delta_{t+1,k}$ using (3), $\tilde{\delta}_{t+1,k}$ using (4), and $Q_{t+1}$ using (16).

end

**Proof:** The proof is presented in Appendix B. \qed 

Having derived the upper bound of the DPP in (20), our goal is to minimize (20) subject to constraints (17b)–(17d) with variables $\{u_{t,k}, r_{t,k}\}_{k \in K}$. According to the standard procedure, we drop the expectations in (20) [19, p. 59]. We denote the upper bound of the DPP after dropping the expectations at slot $t$ by $W_t$, which its derivation is presented in Appendix C, can be expressed as

$$W_t = V \sum_{k \in K} u_{t,k} + r_{t,k} + \frac{1}{2} K^2 \left[ \sum_{k \in K} u_{t,k} f(1) \left( \delta_{t,k}' \right)^2 ight]$$

$$+ r_{t,k} f(x_{t,k} + 1) \left( \delta_{t,k}' \right)^2$$

$$+ \left[ 1 - u_{t,k} f(1) - r_{t,k} f(x_{t,k} + 1) \right] \left( \tilde{\delta}_{t,k} \right)^2$$

$$+ \sum_{k \in K, k \neq k'} \sum_{k' \in K} u_{t,k} f(1) \left( \delta_{t,k}' \right)^2 \tilde{\delta}_{t,k'}$$

$$+ r_{t,k} f(x_{t,k} + 1) \left( \delta_{t,k}' \right)^2 - u_{t,k} f(1) \delta_{t,k} \delta_{t,k'}$$

$$- r_{t,k} f(x_{t,k} + 1) \tilde{\delta}_{t,k} \delta_{t,k'} + u_{t,k} f(1) \delta_{t,k} \tilde{\delta}_{t,k'}$$

$$+ r_{t,k} f(x_{t,k} + 1) \left( \delta_{t,k}' \right)^2 \tilde{\delta}_{t,k} - u_{t,k} f(1) \tilde{\delta}_{t,k} \tilde{\delta}_{t,k'}$$

$$- r_{t,k} f(x_{t,k} + 1) \tilde{\delta}_{t,k} \tilde{\delta}_{t,k'}$$

$$+ 2 K Q_1 \sum_{k \in K} u_{t,k} f(1) \tilde{\delta}_{t,k}' + r_{t,k} f(x_{t,k} + 1) \tilde{\delta}_{t,k}$$

$$+ \left[ 1 - u_{t,k} f(1) - r_{t,k} f(x_{t,k} + 1) \right] \tilde{\delta}_{t,k}$$

$$+ \frac{1}{2} \left[ \left( \Delta_{\max} \right)^2 - 2 Q_1 \Delta_{\max} \right].$$

Accordingly, the solution of (17) can be derived by solving the following problem

**minimize** $W_t$ \hspace{1cm} (24a)

**subject to** (17b)–(17d), \hspace{1cm} (24b)

with variables $\{u_{t,k}, r_{t,k}\}_{k \in K}$.

The proposed LC-DT policy is summarized in Algorithm 2. In Step 1, the transmitter updates the age of random arrival sources’ fresh packets at slot $t$, $\delta_{t,k}'$, using (2). In Step 2, given the current network state, it solves problem (24) to find the optimal transmission decision at slot $t$. In Step 3, the network state (except $\delta_{t,k}'$ for all sources) is updated based on the current network state and the obtained transmission decision. Regarding finding the solution to problem (24), we observe that the number of feasible transmission decisions at each slot is a moderate number. Namely, the transmission options over the $K$ sources are sending a fresh packet (i.e., $K$ options), sending an under-process packet (i.e., $K$ options), or staying idle (i.e., 1 option). On the other hand, if the under-process packet of a source was sent $x_{\text{max}}$ times, the transmitter cannot send that packet. Thus, the number of feasible actions at each slot, i.e., the number of feasible combinations of variables $\{u_{t,k}, r_{t,k}\}_{k \in K}$ in problem (24), is at most $2K + 1$. Because $2K + 1$ is a small value for a reasonable system (increasing only linearly in $K$), we use the exhaustive search algorithm to solve problem (24).

**B. Learning-Based Transmission Policy: Unknown Environment**

In this subsection, we assume that the environment is unknown, i.e., the transmitter does not know the system statistics, namely: 1) the packet arrival probability of each random arrival source, $\lambda_k$, and 2) the probability of successful decoding function, $f(\cdot)$. In this scenario, we transform CMDP problem (11) into an unconstrained MDP problem via the Lyapunov DPP method. Then, we utilize the DQL algorithm [20] to solve the MDP problem. Even though the DQL algorithm cannot ensure the optimality of the solution, the algorithm can be applied 1) without knowing the system statistics, 2) in a system with a large state and action space, and 3) without upper bounding AoI.

Inspired by [44] and using the results of Section IV-A, we transform CMDP problem (11) into an MDP problem, in which our goal is to minimize the time average DPP function

$$\min_{\pi} \lim_{T \to \infty} \frac{1}{T} \sum_{t=1}^{T} \mathbb{E} \left[ V \delta_{t,k} + L(Q_{t+1}) - L(Q_t) \right] \left| o_t \right|.$$

(25)

The system state of the MDP at slot $t$ is $o_t = \{\{\delta_{t,k}', \delta_{t,k}, \tilde{\delta}_{t,k}, x_{t,k}\}_{k \in K}, Q_t\}$ (defined in Section IV-A), and the action at time slot $t$ is $a_t = \{u_{t,k}, r_{t,k}\}_{k \in K} \in \mathcal{A}_t$ (defined in Section III-A). The cost function $\tilde{c}_t$, which is defined as the DPP function, is given as $\tilde{c}_t = V \delta_{t,k} + L(Q_{t+1}) - L(Q_t)$. Note that the state transition probabilities of the MDP problem (25) are unknown, as the environment is unknown.

To solve the MDP problem (25), we use the DQL algorithm in [20, Algorithm 1]. According to the DQL algorithm, an action is selected at each state to maximize a cumulative discounted immediate reward. As we aim to minimize cost function $\tilde{c}_t$, the immediate reward is defined as $r_t = -\tilde{c}_t$. The implementation of DQL, along with the parameters, is presented in Section V.
with decreases, the average $\Delta$ versus time slots for different $p_0$. It can be seen from Fig. 2(a) that $\bar{\tau}$ increases when $p_0$ increases. This behavior is due to the fact that when $p_0$ increases, the probability of successful decoding decreases, and thus, more transmission attempts are needed to meet the average AoI constraint. For example, when $\Delta_{\text{max}} = 4$, by increasing $p_0$ from 0.4 to 0.6, $\bar{\tau}$ increases by about 50%. In addition, $\bar{\tau}$ decreases when $\Delta_{\text{max}}$ increases, because the transmitter needs fewer transmission attempts to satisfy the AoI constraint.

Fig. 2(b) shows the evolution of $\bar{\tau}$ with respect to time slots for the different packet arrival rate, $\lambda_k$, and $\Delta_{\text{max}}$. From Fig. 2(b), it can be seen that when $\lambda_k$ decreases, the average number of transmissions increases dramatically. For example, when $\Delta_{\text{max}} = 4$, by decreasing $\lambda_k$ from 0.5 to 0.2, the value of $\bar{\tau}$ increases by about 75%. This is because when $\lambda_k$ decreases, the availability of the fresh packets at the random arrival source decreases, and consequently, the AoI of this source increases. In this case, to satisfy the AoI constraint, the transmitter must send the generate-at-will source’s packets more frequently to compensate for the negative effect of the random arrival sources on the average AoI.

V. NUMERICAL RESULTS

In this section, we evaluate the performance of the proposed transmission scheduling policies, namely: 1) the deterministic transmission policy presented in Algorithm 1 in Section III (obtained via the CMDP formulation), 2) the low-complexity dynamic transmission (LC-DT) policy presented in Algorithm 2 in Section IV-A, and 3) the learning-based transmission policy presented in Section IV-B.

For the probability of successful decoding, we use the function in [31], i.e., $f(x_{t,k}) = 1 - p_0(1-p_0\tau)^{x_{t,k}}$, where $p_0 \in [0, 1]$ is the error probability of the first transmission of a packet and $\eta \in [0, 1]$ determines the effectiveness of the HARQ protocol. The parameters of this function depend on the channel conditions, channel coding methods, and combining technique employed in the HARQ protocol [32], [33], [34], [35]. Unless otherwise specified, we consider one random arrival source and one generate-at-will source, i.e., $K = 2$, and we set $\sigma_{\text{max}} = 18$ and $\beta_{\text{max}} = 5$. The rest of the parameters are specified in each figure.

A. Deterministic Transmission Policy

In Fig. 2, we evaluate the deterministic transmission policy and the impact of system parameters on the performance. For Algorithm 1, we set the bounds on the Lagrangian multiplier as $\beta_1 = 1$, $\beta_2 = 0$, the bisection stopping criterion as $\epsilon = 0.005$, and the RVIA stopping criterion as $\epsilon = 0.01$.

Fig. 2(a) illustrates the evolution of average number of transmissions, $\bar{\tau}$, with respect to time slots for different values of maximum allowable average AoI, $\Delta_{\text{max}}$, and the error probability of the first transmission, $p_0$. It can be seen from Fig. 2(a) that $\bar{\tau}$ increases when $p_0$ increases. This behavior is due to the fact that when $p_0$ increases, the probability of successful decoding decreases, and thus, more transmission attempts are needed to meet the average AoI constraint. For example, when $\Delta_{\text{max}} = 4$, by increasing $p_0$ from 0.4 to 0.6, $\bar{\tau}$ increases by about 50%. In addition, $\bar{\tau}$ decreases when $\Delta_{\text{max}}$ increases, because the transmitter needs fewer transmission attempts to satisfy the AoI constraint.

B. LC-DT Policy

In this subsection, we analyze the performance of the LC-DT policy for different values of the DPP trade-off parameter, $V$. Recall that $V$ is a positive parameter used to adjust the tradeoff between minimizing the objective function (i.e., the average number of transmissions) and the size of the (virtual) queue backlog which enforces the constraint on the (maximum) average AoI. A larger value of $V$ puts more emphasis on the objective function (i.e., minimizing the average number of transmissions) at an expense of increased queue backlog (which reduces margin in the AoI constraint).

Fig. 3(a) illustrates the evolution of the average number of transmissions, $\bar{\tau}$, as a function of number of time slots for different values of the DPP trade-off parameter, $V$. Correspondingly, Fig. 3(b) shows the average AoI as a function of number of time slots for the same values of $V$. As it can be seen in Fig. 3, the average AoI constraint is satisfied for all values of the parameter $V$ but there is a trade-off between the average number of transmissions and the average AoI: the higher the value of $V$ the smaller the average number of transmissions (i.e., better objective function) at the expense of a reduced margin in the average AoI constraint. Fig. 3 further shows that increasing $V$ beyond the value $V = 30$ leads to negligible improvement in the objective function. Hence, a value of $V = 30$ is a reasonable choice when employing the LC-DT policy in practice.

C. Learning-Based Transmission Policy

In this section, we examine the learning-based transmission policy by evaluating the average number of transmissions, $\bar{\tau}$.
Fig. 3. Performance of the LC-DT policy versus time slots for different values of $V$, where $\eta = 0.4$, $p_0 = 0.4$, $\lambda_k = 0.7$ for all $k \in \mathcal{I}$, and $\Delta_{\text{max}} = 4$. (a) Average number of transmissions. (b) Average AoI.

To implement DQL algorithm, we use a fully-connected deep neural network with two hidden layers. Each hidden layer has 256 neurons with ReLU activation function. The optimizer is Adam, the mini-batch size is 32, and the replay memory size is 100000. We set the learning rate as 0.001, the discount factor as 0.99, the number of steps per episode as 1000, and the target network update rate as $1/500$. The convergence of the policy for different values of $V$ are shown in Fig. 4(a). It can be seen that by taking about 800 episodes, the DQL-agent (transmitter) is learned. After this, we stop the learning process and use the learned transmitter to operate in the system. Fig. 4(b) shows the performance of the algorithm by evaluating $\bar{\tau}$ with respect to time slots for different values of $V$. Similar to Fig. 3, it can be seen that the performance is not considerably improved for $V$ larger than 30.

D. Comparison of the Proposed Policies

Fig. 5 shows the average number of transmissions, $\bar{\tau}$, as a function of $\Delta_{\text{max}}$ under different policies. In this figure, we plot the (infeasible) lower bound policy, obtained in Algorithm 1, as a benchmark. Furthermore, we consider a (feasible) baseline policy, where the transmitter sends a packet whenever the average AoI reaches $\Delta_{\text{max}}$. In every transmission attempt, the source with larger AoI is selected; if there are multiple sources with the largest AoI, one of them is selected randomly. The policy employs an HARQ protocol where the transmitter persistently re-transmits the packet at consecutive slots until it is transmitted successfully or reaches the maximum allowed number of transmissions $x_{\text{max}}$. According to Fig. 5, as expected, the lower bound policy outperforms the other proposed policies as it does not satisfy the constraint. Both the LC-DT policy and the deterministic transmission policy have a small gap with the lower bound policy, which implies their near-optimal performance. The learning-based transmission policy, which is obtained without knowing the system statistics, performs relatively close to the policies for the known environment, especially for $\Delta_{\text{max}} \geq 4$. In
In Fig. 6, we evaluate the impact of HARQ protocol on the system’s performance in terms of the average number of transmissions, \( \bar{\tau} \). Here, without loss of generality, we use the LC-DT policy. Fig. 6(a) shows the effect of the maximum allowed number of transmissions for a packet, \( x_{\text{max}} \), where \( x_{\text{max}} = 1 \) indicates that the system operates without HARQ protocol. As it can be seen, for a high probability of successful decoding (small \( p_0 \)), i.e., for good channel conditions, HARQ is not that beneficial. This is because, with a high probability of successful decoding, most packets are successfully decoded in the first transmission attempt. However, in bad channel conditions, HARQ plays an important role. For example, when \( p_0 = 0.6 \), the performance improves substantially by increasing \( x_{\text{max}} \) from 1 to 2, i.e., by merely activating the HARQ with only one allowed retransmission. It is worth noting that when \( p_0 = 0.7 \), the transmitter cannot satisfy the average AoI constraint without HARQ (\( x_{\text{max}} = 1 \)). This is because when \( p_0 \) is large, the first transmit attempts tend to fail, in this case, the retransmissions would be the crucial enabler for successful receptions of the packets.

In Fig. 6(b), we study the effect of employing HARQ by evaluating \( \bar{\tau} \) with respect to the number of sources \( K \), where \( K = I + J \) and \( I = J \). According to this figure, \( \bar{\tau} \) increases by increasing \( K \). This is due to the fact that the transmitter has to send each source’s packets more frequently to satisfy the AoI constraint, the increased number of sources inevitably leads to more transmissions in the system. Moreover, similar to Fig. 6(a), it can be seen in Fig. 6(b) that employing HARQ decreases \( \bar{\tau} \), as HARQ benefits from the increased probability of successful decoding via retransmissions. For example, with \( p_0 = 0.6 \) and \( K = 6 \), \( \bar{\tau} \) with HARQ is 30% lower than that in the case without HARQ. Interestingly, the system cannot support \( K = 10 \) sources for \( p_0 \geq 0.4 \), unless HARQ is employed.

**E. Impact of HARQ**

**F. AoI for Individual Sources**

Fig. 7 depicts the average AoI of each source as well as the average AoI of the system with four sources versus the maximum allowable average AoI, \( \Delta_{\text{max}} \), under different packet arrival rates. From the figure, the average AoI of the system is close to the average AoI of each source for the considered parameters. Moreover, we can see that the average AoI of the generate-at-will sources are smaller than the average AoI of random arrival sources, as expected. This is because the generate-at-will sources access a fresher packet at each time slot.

---

**TABLE I**

| Policy          | Offline phase | Online phase |
|-----------------|---------------|--------------|
| Deterministic policy | 105013.745 s  | 0.031 ms     |
| LC-DT policy    | 0             | 0.342 ms     |
| Learning-based policy | 4029.38 s    | 0.934 ms     |
VI. CONCLUSION

We studied an HARQ-based multi-source status update system with random arrival and generate-at-will sources, communicating through an error-prone channel. We solved the problem of minimizing the average AoI constraint in the known and unknown environments. We developed a deterministic transmission policy using the RVIA and the bisection for the known environment. For the sake of reduced computational complexity, we developed the LC-DT policy using the DPP method for the known environment. For the unknown environment, we utilized the DPP method and the DQL algorithm to develop a learning-based transmission policy. The numerical results showed the near-optimal performance of the deterministic transmission policy and the LC-DT policy. Also, the learning-based policy attained performance relatively close to the policies developed for the known environment. Overall, the results showed about 40% performance gain for the proposed policies over a baseline scheduling policy and demonstrated the great potential of HARQ to improve information freshness in multi-source status update systems.

APPENDIX A

PROOF OF LEMMA 1

To derive \( \mathbb{E}\{\hat{\delta}_{t+1} \mid o_t\} \), we use the definition in (8), \( \hat{\delta}_t = \frac{1}{K} \sum_{k=1}^{K} \hat{\delta}_{t,k} \), and re-express \( \delta_{t+1,k} \) via (4) as

\[
\begin{align*}
\delta_{t+1,k} &= u_{t,k}d_{t,k} \hat{\delta}_{t,k} + r_{t,k}d_{t,k} \hat{\delta}_p \left[ u_{t,k} - u_{t,k} \right] \hat{\delta}_{t,k} \\
&\quad + r_{t,k}(1 - d_{t,k}) + \left(1 - u_{t,k} - r_{t,k}\right) \hat{\delta}_{t,k} \\
&= u_{t,k}d_{t,k} \hat{\delta}_{t,k} + r_{t,k}d_{t,k} \hat{\delta}_p + [u_{t,k} - u_{t,k} - d_{t,k}] \hat{\delta}_{t,k} \\
&\quad + r_{t,k} - r_{t,k}d_{t,k} + 1 - u_{t,k} - r_{t,k} \hat{\delta}_{t,k} \\
&= u_{t,k}d_{t,k} \hat{\delta}_{t,k} + r_{t,k}d_{t,k} \hat{\delta}_p + [1 - d_{t,k}(u_{t,k} + r_{t,k})] \hat{\delta}_{t,k}.
\end{align*}
\]

Taking conditional expectation in (26), \( \mathbb{E}\{\hat{\delta}_{t+1,k} \mid o_t\} \) is expressed as

\[
\mathbb{E}\{\hat{\delta}_{t+1,k} \mid o_t\} = \mathbb{E}\{u_{t,k}d_{t,k} \hat{\delta}_{t,k} \mid o_t\} + \mathbb{E}\{r_{t,k}d_{t,k} \hat{\delta}_p \mid o_t\} + \mathbb{E}\{[1 - d_{t,k}(u_{t,k} + r_{t,k})] \hat{\delta}_{t,k} \mid o_t\}
\]

where equality (a) follows from the fact that \( \hat{\delta}_{t,k} = \hat{\delta}_{p} \), and \( \hat{\delta}_{t,k} \) are given by the network state.

We need to calculate \( \mathbb{E}\{u_{t,k}d_{t,k} \mid o_t\} \) and \( \mathbb{E}\{r_{t,k}d_{t,k} \mid o_t\} \) in (27) which are given by the following two lemmas.

Lemma 3: For any source \( k \), the conditional expectation \( \mathbb{E}\{u_{t,k}d_{t,k} \mid o_t\} \) is given as

\[
\mathbb{E}\{u_{t,k}d_{t,k} \mid o_t\} = \mathbb{E}\{u_{t,k} \mid o_t\} f(1).
\]

Proof: Based on the law of iterated expectations, we have

\[
\mathbb{E}\{u_{t,k}d_{t,k} \mid o_t\} = \mathbb{E}\{ \mathbb{E}\{u_{t,k}d_{t,k} \mid o_t, u_{t,k}\} \mid o_t\}
\]

\[
= \mathbb{E}\{1d_{t,k} \mid o_t, u_{t,k} = 1\} \Pr(u_{t,k} = 1 \mid o_t) + \mathbb{E}\{0d_{t,k} \mid o_t, u_{t,k} = 0, \Pr(u_{t,k} = 0 \mid o_t)
\]

\[
= f(1)(1 + f(1)) \Pr(u_{t,k} = 1 \mid o_t) = f(1) \Pr(u_{t,k} = 1 \mid o_t)
\]

Lemma 4: For any source \( k \), the conditional expectation \( \mathbb{E}\{r_{t,k}d_{t,k} \mid o_t\} \) is given as

\[
\mathbb{E}\{r_{t,k}d_{t,k} \mid o_t\} = \mathbb{E}\{r_{t,k} \mid o_t\} f(x_{t,k} + 1).
\]

Proof: Following the same steps as in the proof of Lemma 3, we have

\[
\mathbb{E}\{r_{t,k}d_{t,k} \mid o_t\} = \mathbb{E}\{ \mathbb{E}\{r_{t,k}d_{t,k} \mid o_t, r_{t,k}\} \mid o_t\}
\]

\[
= \mathbb{E}\{1d_{t,k} \mid o_t, r_{t,k} = 1\} \Pr(r_{t,k} = 1 \mid o_t) + \mathbb{E}\{0d_{t,k} \mid o_t, r_{t,k} = 0\} \Pr(r_{t,k} = 0 \mid o_t)
\]

\[
= f(x_{t,k} + 1) \Pr(r_{t,k} = 1 \mid o_t) = \mathbb{E}\{r_{t,k} \mid o_t\} f(x_{t,k} + 1),
\]

where the equality (a) comes from the following equality

\[
\mathbb{E}\{r_{t,k} \mid o_t\} = 0\Pr(u_{t,k} = 0 \mid o_t) + 1\Pr(u_{t,k} = 1 \mid o_t) = \Pr(u_{t,k} = 1 \mid o_t).
\]

Using Lemmas 3 and 4, the expression in (27) becomes

\[
\mathbb{E}\{\hat{\delta}_{t+1} \mid o_t\} = \frac{1}{K} \sum_{k=1}^{K} \mathbb{E}\{u_{t,k} \mid o_t\} f(1) \hat{\delta}_{t,k}
\]

\[
+ \mathbb{E}\{r_{t,k} \mid o_t\} f(x_{t,k} + 1) \hat{\delta}_{p} + [1 - f(1)] \mathbb{E}\{u_{t,k} \mid o_t\} \hat{\delta}_{t,k}
\]

\[
- f(x_{t,k} + 1) \mathbb{E}\{r_{t,k} \mid o_t\} \hat{\delta}_{t,k}.
\]
To derive \( E\{\delta_{t+1}^2 | o_t\} \), we use the definition in (8),
\[
\hat{\delta}_t = \frac{1}{K} \sum_{k=1}^{K} \delta_{t,k},
\]
and re-express it as
\[
E\{\hat{\delta}_{t+1}^2 | o_t\} = \frac{1}{K^2} \sum_{k \in K} E\{\delta_{t+1,k}^2 | o_t\}
\]
\[
+ \sum_{k \in K} \sum_{k' \in K, k' \neq k} E\{\delta_{t+1,k,k'} | o_t\}.
\]
(33)

We need to calculate the terms \( E\{\delta_{t+1,k}^2 | o_t\} \) and \( E\{\delta_{t+1,k} \delta_{t+1,k'} | o_t\} \). As the conditions in (4) are mutually exclusive and collectively exhaustive and the square of a binary variable equals the variable itself, \( \delta_{t+1,k}^2 \) can be calculated from (26) as
\[
\delta_{t+1,k}^2 = u_{t,k} d_t \hat{\delta}_t^2 + r_{t,k} d_t \hat{\delta}_t^2
\]
\[
+ [1 - d_t(u_{t,k} + r_{t,k})] \hat{\delta}_t^2.
\]
(34)

Taking conditional expectation in (34) as
\[
E\{\delta_{t+1,k}^2 | o_t\} = E\{u_{t,k} d_t \hat{\delta}_t^2 | o_t\}
\]
\[
+ E\{r_{t,k} d_t \hat{\delta}_t^2 | o_t\}
\]
\[
+ E\{[1 - d_t(u_{t,k} + r_{t,k})] \hat{\delta}_t^2 | o_t\}
\]
\[
\equiv\ E\{u_{t,k} d_t | o_t\} \hat{\delta}_t^2 + E\{r_{t,k} d_t | o_t\} \hat{\delta}_t^2
\]
\[
+ [1 - E\{d_t(u_{t,k} + r_{t,k}) | o_t\}] \hat{\delta}_t^2,
\]
(35)

where equality (a) follows from the fact that \( \delta_{t,k}^2, \hat{\delta}_t^2, \) and \( \delta_{t,k} \) are given by the network state.

Using Lemmas 3 and 4, the conditional expectation in (35) is given as
\[
E\{\delta_{t+1,k}^2 | o_t\} = E\{u_{t,k} | o_t\} f(1) \hat{\delta}_t^2
\]
\[
+ E\{r_{t,k} | o_t\} f(x_{t,k} + 1) \hat{\delta}_t^2
\]
\[
+ [1 - f(1)] E\{u_{t,k} | o_t\}
\]
\[
- f(x_{t,k} + 1) E\{r_{t,k} | o_t\} \hat{\delta}_t^2.
\]
(36)

Now, we calculate \( \delta_{t+1,k} \delta_{t+1,k'} \) in expression (33). Based on (26), we can express it as
\[
\delta_{t+1,k} \delta_{t+1,k'} = u_{t,k} d_t - d_t(u_{t,k} + r_{t,k}) \hat{\delta}_t \hat{\delta}_t
\]
\[
+ r_{t,k} d_t - d_t(u_{t,k} + r_{t,k}) \hat{\delta}_t \hat{\delta}_t
\]
\[
+ (1 - d_t(u_{t,k} + r_{t,k})) d_t u_{t,k} \hat{\delta}_t \hat{\delta}_t
\]
\[
+ (1 - d_t(u_{t,k} + r_{t,k})) d_t r_{t,k} \hat{\delta}_t \hat{\delta}_t
\]
\[
+ (1 - d_t(u_{t,k} + r_{t,k})) (1 - d_t(u_{t,k} + r_{t,k})) \times \hat{\delta}_t \hat{\delta}_t.
\]
(37)

Because the transmitter can transmit one packet per slot, we have \( u_{t,k} r_{t,k} = 0 \) and \( u_{t,k} r_{t,k} = 0 \). Thus, the expression in (37) is rewritten as
\[
\delta_{t+1,k} \delta_{t+1,k'} = u_{t,k} d_t \hat{\delta}_t \hat{\delta}_t + r_{t,k} d_t \hat{\delta}_t \hat{\delta}_t
\]
\[
+ u_{t,k} d_t \hat{\delta}_t \hat{\delta}_t + r_{t,k} d_t \hat{\delta}_t \hat{\delta}_t
\]
\[
- u_{t,k} d_t \hat{\delta}_t \hat{\delta}_t - r_{t,k} d_t \hat{\delta}_t \hat{\delta}_t
\]
\[
- u_{t,k} d_t \hat{\delta}_t \hat{\delta}_t - r_{t,k} d_t \hat{\delta}_t \hat{\delta}_t + \hat{\delta}_t \hat{\delta}_t.
\]
(38)

Using Lemmas 3 and 4, the conditional expectation of the expression in (38) is calculated as
\[
E\{\delta_{t+1,k} \delta_{t+1,k'} | o_t\} = E\{u_{t,k} | o_t\} f(1) \hat{\delta}_t^2
\]
\[
+ E\{r_{t,k} | o_t\} f(x_{t,k} + 1) \hat{\delta}_t^2
\]
\[
+ E\{r_{t,k} | o_t\} f(1) \hat{\delta}_t^2
\]
\[
- E\{u_{t,k} | o_t\} f(1) \hat{\delta}_t^2
\]
\[
- E\{r_{t,k} | o_t\} f(x_{t,k} + 1) \hat{\delta}_t^2
\]
\[
- E\{u_{t,k} | o_t\} f(1) \hat{\delta}_t^2
\]
\[
- E\{r_{t,k} | o_t\} f(x_{t,k} + 1) \hat{\delta}_t^2.
\]
(39)

Substituting (36) and (39) into (33), we derive
\[
E\{\hat{\delta}_{t+1}^2 | o_t\} = \frac{1}{K^2} \sum_{k \in K} E\{u_{t,k} | o_t\} f(1) \hat{\delta}_t^2
\]
\[
+ E\{r_{t,k} | o_t\} f(x_{t,k} + 1) \hat{\delta}_t^2 + [1 - f(1) E\{u_{t,k} | o_t\}
\]
\[
- f(x_{t,k} + 1) E\{r_{t,k} | o_t\} \hat{\delta}_t^2
\]
\[
+ \sum_{k \in K} \sum_{k' \in K, k' \neq k} E\{u_{t,k} | o_t\} f(1) \hat{\delta}_t^2
\]
\[
+ E\{r_{t,k} | o_t\} f(x_{t,k} + 1) \hat{\delta}_t^2
\]
\[
+ E\{r_{t,k} | o_t\} f(1) \hat{\delta}_t^2
\]
\[
- E\{u_{t,k} | o_t\} f(1) \hat{\delta}_t^2
\]
\[
- E\{r_{t,k} | o_t\} f(x_{t,k} + 1) \hat{\delta}_t^2
\]
\[
- E\{r_{t,k} | o_t\} f(x_{t,k} + 1) \hat{\delta}_t^2.
\]
(40)

**APPENDIX C**

**The Derivation of \( W_t \) Presented in (23)**

To derive \( W_t \), we rewrite (20) using Lemmas 1 and 2 as
\[
V E\{r_t | o_t\} + \alpha(o_t) \leq V \sum_{k \in K} E\{u_{t,k} | o_t\}
\]
\[
+ E\{r_{t,k} | o_t\} + \frac{1}{2} (\Delta_{\text{max}}^2)
\]
\[
+ \frac{1}{K^2} \sum_{k \in K} E\{u_{t,k} | o_t\} f(1) \hat{\delta}_t^2
\]
\[
+ E\{r_{t,k} | o_t\} f(x_{t,k} + 1) \hat{\delta}_t^2 + |1
\]

\[\]
\[
- f(1)E\{u_{t,k} \mid o_t\} - f(x_{t,k} + 1)E\{r_{t,k} \mid o_t\} \left(\tilde{\delta}_{t,k}\right)^2 \\
+ \sum_{k \in K} \sum_{k' \neq k} E\{u_{t,k} \mid o_t\} f(1)\tilde{\delta}_{t,k}\tilde{\delta}_{t,k'} \\
+ E\{r_{t,k} \mid o_t\} f(x_{t,k} + 1)\tilde{\delta}_{t,k}\tilde{\delta}_{t,k'} \\
+ E\{u_{t,k'} \mid o_t\} f(1)\tilde{\delta}_{t,k'}\tilde{\delta}_{t,k} \\
+ E\{r_{t,k'} \mid o_t\} f(x_{t,k'} + 1)\tilde{\delta}_{t,k'}\tilde{\delta}_{t,k} \\
- E\{u_{t,k'} \mid o_t\} f(1)\tilde{\delta}_{t,k}\tilde{\delta}_{t,k'} \\
- E\{r_{t,k} \mid o_t\} f(x_{t,k} + 1)\tilde{\delta}_{t,k}\tilde{\delta}_{t,k'} \\
- E\{u_{t,k'} \mid o_t\} f(1)\tilde{\delta}_{t,k'}\tilde{\delta}_{t,k} \\
- E\{r_{t,k'} \mid o_t\} f(x_{t,k'} + 1)\tilde{\delta}_{t,k'}\tilde{\delta}_{t,k} + \tilde{\delta}_{t,k}\tilde{\delta}_{t,k'} \\
+ 2Q_t \left(\frac{1}{K} \sum_{k \in K} E\{u_{t,k} \mid o_t\} f(1)\tilde{\delta}_{t,k}^2 \right) \\
+ E\{r_{t,k} \mid o_t\} f(x_{t,k} + 1)\tilde{\delta}_{t,k} - f(x_{t,k} + 1)E\{r_{t,k} \mid o_t\} \left(\tilde{\delta}_{t,k} - \Delta_{\text{max}}\right) \right). \tag{40}
\]

Dropping the expectation in (40), \(W_t\) is derived as

\[
W_t = V \sum_{k \in K} u_{t,k} + r_{t,k} + \frac{1}{2}K^2 \left(\sum_{k \in K} u_{t,k} f(1)\tilde{\delta}_{t,k}^2 \right) \\
+ r_{t,k} f(x_{t,k} + 1)\tilde{\delta}_{t,k}^2 \\
+ \left[1 - u_{t,k} f(1) - r_{t,k} f(x_{t,k} + 1)\right] \left(\tilde{\delta}_{t,k} - \Delta_{\text{max}}\right)^2 \\
+ \sum_{k \in K} \sum_{k' \neq k} u_{t,k} f(1)\tilde{\delta}_{t,k}\tilde{\delta}_{t,k'} \\
+ r_{t,k} f(x_{t,k} + 1)\tilde{\delta}_{t,k} - u_{t,k} f(1)\tilde{\delta}_{t,k}\tilde{\delta}_{t,k'} \\
- r_{t,k} f(x_{t,k} + 1)\tilde{\delta}_{t,k} - u_{t,k} f(1)\tilde{\delta}_{t,k}\tilde{\delta}_{t,k'} \\
+ r_{t,k} f(x_{t,k} + 1)\tilde{\delta}_{t,k} - u_{t,k} f(1)\tilde{\delta}_{t,k}\tilde{\delta}_{t,k'} \\
- r_{t,k} f(x_{t,k} + 1)\tilde{\delta}_{t,k} - \tilde{\delta}_{t,k}\tilde{\delta}_{t,k'} \\
+ 2KQ_t \left(\sum_{k \in K} u_{t,k} f(1)\tilde{\delta}_{t,k}^2 + r_{t,k} f(x_{t,k} + 1)\tilde{\delta}_{t,k}^2 \right) \\
+ \left[1 - u_{t,k} f(1) - r_{t,k} f(x_{t,k} + 1)\right] \tilde{\delta}_{t,k} \\
+ \frac{1}{2} \left(\Delta_{\text{max}}^2 - 2Q_t\Delta_{\text{max}}\right).
\]
[29] Z. Deng, S. Wu, C. Guo, J. Jiao, N. Zhang, and Q. Zhang, “Age-optimal transmission policy for intelligent HARQ-CC aided NOMA systems,” in Proc. IEEE Int. Conf. Commun., vol. 17, no. 7, pp. 4988–4998, 2021.

[30] S. Feng and J. Yang, “Age of Information minimization for an energy harvesting source with updating erasures: Without and with feedback,” IEEE Trans. Commun., vol. 69, no. 8, pp. 5091–5105, Aug. 2021.

[31] Y. Wang, S. Wu, J. Jiao, W. Wu, Y. Wang, and Q. Zhang, “Age-optimal transmission policy with HARQ for freshness-critical vehicular status updates in space-air-ground integrated networks,” IEEE Internet Things J., vol. 9, no. 8, pp. 5719–5729, Apr. 2022.

[32] X. Lagrange, “Throughput of HARQ protocols on a block fading channel,” IEEE Commun. Lett., vol. 14, no. 3, pp. 257–259, Mar. 2010.

[33] P. Frenger, S. Parkvall, and E. Dahlman, “Performance comparison of HARQ with chase combining and incremental redundancy for HSDPA,” in Proc. IEEE Veh. Technol. Conf., 2001, pp. 1829–1833.

[34] K. Chen, K. Niu, Z. He, and J. Lin, “Polar coded HARQ scheme with chase combining,” in Proc. IEEE Wireless Commun. Netw. Conf., 2014, pp. 474–479.

[35] F. Nadeem, Y. Li, B. Vuetic, and M. Shirvani Moghadam, “Analysis and optimization of HARQ for URLLC,” in Proc. IEEE Globecom Workshops, 2021, pp. 1–6.

[36] M. Moltafet, M. Leinonen, M. Codreanu, and N. Pappas, “Power minimization for Age of Information constrained dynamic control in wireless sensor networks,” IEEE Trans. Commun., vol. 70, no. 1, pp. 419–432, Jan. 2022.

[37] A. Maatouk, S. Krouile, M. Assad, and A. Ephremides, “On the optimality of the Whittle’s index policy for minimizing the Age of Information,” IEEE Trans. Wireless Commun., vol. 20, no. 2, pp. 1263–1277, Feb. 2021.

[38] G. Yao, A. Bedewy, and N. B. Shroff, “Age-optimal low-power status update over time-correlated fading channel,” IEEE Trans. Mobile Comput., vol. 22, no. 8, pp. 4500–4514, Aug. 2023.

[39] D. V. Djonin and V. Krishnamurthy, “MIMO transmission control in fading channels—A constrained Markov decision process formulation with monotone randomized policies,” IEEE Trans. Signal Process., vol. 55, no. 10, pp. 5069–5083, Oct. 2007.

[40] E. Altman, Constrained Markov Decision Processes. Boca Raton, FL, USA: CRC Press, 1999.

[41] D.-J. Ma, A. M. Makowski, and A. Shwartz, “Estimation and optimal control for constrained Markov chains,” in Proc. 25th IEEE Conf. Decis. Control, 1986, pp. 994–999.

[42] M. L. Puterman, Markov Decision Processes: Discrete Stochastic Dynamic Programming. Hoboken, NJ, USA: Wiley, 1994.

[43] M. J. Neely, “Queue stability and probability 1 convergence via Lyapunov optimization,” 2010, arXiv:1008.3519. [Online]. Available: https://arxiv.org/abs/1008.3519

[44] W. Wu, P. Yang, W. Zhang, C. Zhou, and X. Shen, “Accuracy-guaranteed collaborative DNN inference in industrial IoT via deep reinforcement learning,” IEEE Trans. Ind. Inform., vol. 17, no. 7, pp. 4988–4998, Jul. 2021.

[45] Mohammad Moltafet (Member, IEEE) received the M.Sc. degree in communications engineering from Tarbiat Modares University, Tehran, Iran, in 2016 and the Ph.D. degree in communications engineering from the University of Oulu, Oulu, Finland, in 2021. He was a Visiting Ph.D. Researcher with Linköping University, Linköping, Sweden, in 2019, a Postdoctoral Researcher with the University of Oulu, and the Technical University of Munich, Munich, Germany, in 2021 and 2022, respectively. He is currently a Postdoctoral Researcher with the University of California Santa Cruz, Santa Cruz, CA, USA. His research interests include information freshness, information theory, stochastic optimization, and reinforcement learning for wireless applications.

[46] Markus Leinonen (Member, IEEE) received the B.Sc. (Tech.) and M.Sc. (Tech.) degrees in electrical engineering and the D.Sc. (Tech.) degree in communications engineering from the University of Oulu, Oulu, Finland, in 2010, 2011, and 2018, respectively. He received the master’s thesis with the title Power Minimization in Single-Sink Data Gathering Wireless Sensor Network via Distributed Source Coding in 2011. In 2010, he was with the Centre for Wireless Communications, University of Oulu, as a master’s Thesis Student. He directly continued to work on his doctoral research resulting in the thesis title Distributed Compressed Data Gathering in Wireless Sensor Networks. He was the recipient of the Academy of Finland Post-Doctoral Researcher position for his project Time-Critical Communications for Internet of Things Systems from 2021 to 2024. In 2013, he was the Guest Researcher with the Technical University of Munich, Munich, Germany. In 2020, he was a Visiting Postdoctoral Researcher with the University of California San Diego, San Diego, CA, USA. In February 2023, he was conferred the title of Docent (Adjunct Professor) with the University of Oulu. He has authored and coauthored more than 70 journal and conference papers in the area of wireless communications. His research interests included time-critical and sparsity-aware wireless communications, including optimization and analysis of information freshness in status update systems and design of sparse signal recovery methods for channel estimation as well as for user activity detection in massive machine-type communications. He passed away on April 4, 2023.

[47] Marian Codreanu (Member, IEEE) received the M.Sc. degree from the University Politehnica of Bucharest, Bucharest Romania, in 1998 and the Ph.D. degree from the University of Oulu, Oulu, Finland, in 2008. In 2008, he was a Visiting Postdoctoral Researcher with the University of Maryland, College Park, MD, USA. Dr. Codreanu authored and coauthored more than 150 journal and conference papers in the areas of wireless communications, including optimization and analysis of information freshness in status update systems and design of sparse signal processing methods for channel estimation as well as for user activity detection in massive machine-type communications. He passed away on April 4, 2023.

[48] Saed Sadeghi Vilni (Student Member, IEEE) received the M.Sc. (Tech.) degree in electrical engineering from the Tarbiat Modares University, Tehran, Iran, in 2017. He is currently working toward the Ph.D. degree with the Centre for Wireless Communications, University of Oulu, Oulu, Finland. His research interests include the Age of Information, machine learning for wireless applications, and radio resource allocation and optimization in wireless networks.