Adaptive Age of Information Optimization in Rateless Coding-Based Multicast-Enabled Sensor Networks

Hung-Chun Lin, Kuang-Hsun Lin, *Member, IEEE*, and Hung-Yu Wei, *Senior Member, IEEE*

***Abstract*—As the demand for real-time information in Internet of Things and wireless sensor networks (WSN) scenarios grows with the evolution of bandwidth-intensive 5G applications, multicast transmission becomes increasingly vital. This article delves into the significant role of multicast in WSN, exploring a novel perspective of using rateless codes over the traditionally employed hybrid automatic repeat request for eliminating retransmissions in a wireless multicast system. We aim to optimize data freshness, quantified by the age of information (AoI), and analyze strategies to minimize time-average AoI in a multicast environment with diverse channel conditions. Specifically, our policy focuses on optimizing the time-average AoI based on sensor devices' feedback. We transform the problem into a Markov decision process to locate optimal and low-complexity suboptimal policies. We present the first age-minimum scheme for rateless code-based wireless multicast systems. Our numerical simulations reveal that the proposed policies, developed considering unique system structural properties, consistently surpass baseline strategies. We are thus able to preempt updates at the most beneficial time, thereby addressing the issue of the bottleneck device's adverse impact on overall performance.**

***Index Terms*—Age of information (AoI), multicast, rateless code.**

NOMENCLATURE

List of main notations

Notation	Description
$\mathcal{K}$	Set of devices with the cardinality $\lvert\mathcal{K}\rvert = K$.
ξ	Number of packets needed to decode a status update.
p_k	Probability that device k fails to receive a packet.
$u_k(t)$	Indicator showing whether device k successfully receives a packet at time slot t.
$d(t)$	Action that the source takes in time slot t.
$\mathcal{D}$	Set of available actions for the source.
$P(t)$	Number of packets has been generated by the status update at the beginning of time slot t.
$\mathcal{P}$	State space of $P(t)$.
$A_k(t)$	Age of device k at the beginning of time slot t.
$\mathcal{A}$	State space of $A_k(t)$.
$\hat{A}$	Upper limit of the AoI.
$R_k(t)$	Number of packets still needed for device k to decode a update at the beginning of time slot t.
$\mathcal{R}$	State space of $R(t)$.
$X_k(t)$	System state vector of device k at slot t.
$\mathcal{X}_k$	System state space of device k.
$X(t)$	System state vector at time slot t.
$\mathcal{X}$	System state space.
Δ	MDP for the system.
Π	Set of stationary deterministic policies.
π	Stationary deterministic policy.
$V^{\pi}(X)$	Long-term time-average AoI under policy π with initial state X.
$\pi^{\star}$	Age-optimal policy.
$h(\cdot)$	Relative value function.
$h_n(\cdot)$	Relative value function at iteration n.
$Q(\cdot,\cdot)$	Q-factor.
$Q_n(\cdot,\cdot)$	Q-factor at iteration n.
$\mathbf{A}$	Vector of all A_k.
$\mathbf{R}$	Vector of all R_k.
$\pi_n^{\star}$	Age-optimal policy at iteration n.
$\widetilde{X}_k$	Extended state vector of device k.
$\widetilde{\mathcal{X}}_k$	Extended state space of device k.
Δ_k	MDP for device k.
$\widetilde{\pi}^{\star}$	Suboptimal policy.
$h_k(\cdot)$	Relative value function for Δ_k.
$h_k^n(\cdot)$	Relative value function at iteration n for Δ_k.
$Q_k(\cdot,\cdot)$	Q-factor for Δ_k.
$Q_k^n(\cdot,\cdot)$	Q-factor at iteration n for Δ_k.

Manuscript received 10 September 2023; revised 29 November 2023, 30 March 2024, and 11 May 2024; accepted 27 May 2024. Date of publication 30 May 2024; date of current version 20 June 2024. This work was supported in part by the Qualcomm under Grant NAT-435534, and in part by the National Science and Technology Council (NSTC), Taiwan, under Grant 112-2628-E-002-025-. Recommended by Lead Guest Editor Hossein Fotouhi and Guest Editor Mohammad Loni. *(Corresponding author: Hung-Yu Wei.)*

Hung-Chun Lin and Kuang-Hsun Lin are with the Graduate Institute of Communication Engineering, National Taiwan University, Taipei City 106319, Taiwan (e-mail: r10942073@ntu.edu.tw; f03942140@ntu.edu.tw).

Hung-Yu Wei is with the Graduate Institute of Communication Engineering and the Department of Electrical Engineering, National Taiwan University, Taipei City 106319, Taiwan (e-mail: hywei@ntu.edu.tw).

Digital Object Identifier 10.1109/JSAS.2024.3407689

I. INTRODUCTION

WITH the ubiquitous 5G applications opening up a new era of wireless communication, the demand for bandwidth has rapidly increased. In addition, in Internet of Things (IoT)

© 2024 The Authors. This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

and wireless sensor networks (WSN) scenarios, where countless devices ranging from self-driving cars to sensors require real-time information, the need for efficient communication methods is paramount. As the most influential mobile communication standard organization, the third generation partnership project (3GPP) is dedicated to providing solutions that enable IoT and WSN in 3GPP-based systems, such as 5G new radio (NR) or the upcoming 6G system, while considering the limitations of wireless resources. If all receivers need the same information, multicast communication, a transmission strategy that allows packets to be sent from a single sender to multiple receivers simultaneously, is a competitive technology for maintaining freshness and saving bandwidth [1]. In a smart factory, a server multicasts environmental data, like temperature or robot locations, to all robots. These robots act based on this data and their observations, then individually update the server. Similarly, in wildfire detection, positioning algorithms adjust the placement of sensors in real-time, considering factors such as wind direction and vegetation dryness. In high-risk areas, sensors are densely placed, whereas in low-risk areas, they are more dispersed. The central controller multicasts location directives, saving bandwidth and reducing overhead. In both scenarios, outdated data can lead to severe consequences, from robots' inefficiencies to potential mishaps in other applications. These examples highlight multicast's potential in WSN and the importance of data freshness in real-time applications.

To quantitatively measure the freshness of a device, the metric age of information (AoI) is proposed in [2], [3], and [4] and defined as the elapsed time since the generation of the most recent received update packet for a device. The AoI of the device rises with time and falls when a more recent update packet is obtained. A lower AoI at the device suggests that the data is fresher. Notice that AoI is destination-centric as opposed to packet-centric performance metrics, such as latency.

The 3GPP, renowned for its contributions to mobile communication standards, plays a crucial role in efficiently utilizing wireless resources and minimizing data redundancy, especially in scenarios like IoT and video streaming. 3GPP has developed comprehensive standards to facilitate multicast communication in various applications. For instance, 3GPP standards encompass group communications for narrowband IoT [5] and the multicast broadcast service (MBS) [6], catering to the specific multicast requirements of these applications. Multicast transmission in highly real-time is also a promising application, as demonstrated by the multicast usage scenario for mission-critical services discussed in the 3GPP report [7]. 3GPP adopts rateless codes as a forward error correction (FEC) technique to combat transmission error to improve reliability [8]. The rateless code, or digital fountain code, allows a source to generate an unlimited number of packets from data, and a receiver can decode its data with a sufficient number of packets. Rateless codes have been applied in the 4G long term evolution (LTE) multimedia broadcast multicast service (MBMS) and extensively studied, demonstrating their effectiveness in improving the reliability of multicast/broadcast transmissions [9]. Therefore, 3GPP has also proposed applying rateless codes in 5G NR MBS systems. Rateless codes present a significant advantage in multicast scenarios because they allow the transmitter to skip retransmitting packets with errors and instead send newly generated fountain packets.

Regarding scalability, fixed block-length codes typically require data to be sent at a uniform rate that all receivers can handle. This means the transmission rate must be set based on the receiver with the worst channel conditions (the bottleneck device), leading to inefficiencies as other receivers are underutilized. In contrast, rateless codes allow each receiver to gather packets at their own pace, efficiently addressing the bottleneck issue and enhancing network performance. The LT code [10], known as the first implemented rateless code, marks a pivotal development in this field. Following them, the Raptor codes [11] emerged as a successor, offering significant enhancements in terms of reduced complexity with linear encoding and decoding time. In addition, the RaptorQ code [12] has gained popularity as a practical variation of the Raptor code, noted for its improved performance in reducing failure probability and its capacity to support a greater number of source symbols [13].

Interestingly, attaining the minimum time-average AoI in a multicast network may not imply every delivered status update should be reliable [14]. Providing the source sends a new update after all devices decode the current update, the bottleneck device will still affect other devices' AoI performance. Therefore, preempting the current update at the right time and generating a new status update may benefit the overall AoI performance. Based on the feedback from devices, it is possible to find a policy that minimizes the overall time-average AoI. However, it is not trivial to find such a policy due to the complicated combinations of system information, especially as the number of devices in the network increases. Leveraging the knowledge of devices' feedback to develop a more efficient solution is still an open problem.

This article considers a status update system, where a source multicasts status updates of a physical process to multiple devices. The status update is encoded via rateless codes, and AoI is adopted as the metric for evaluating the freshness of each device. We aim to find *the best preemption policy to minimize the time-average AoI by generating a new status update at the appropriate time*. To the best of authors' knowledge, this is the first age-minimum scheme for a rateless code-based wireless multicast status update system. The contributions of this work are as follows.

1) For a detailed discussion on maintaining a real-time multicast transmission scenario, unlike previous studies using hybrid automatic repeat request (HARQ), we focus on a rateless code-based feedback mechanism to improve data freshness.
2) We formulate the system into a communicating Markov decision process (MDP), allowing us to solve the optimal policy by applying the Bellman equation. Further, we exhibit the unique properties of the optimal policy for this MDP, employing these properties to locate the optimal policy.
3) In the context of large-scale networks, securing the optimal policy becomes impractical due to the curse of dimensionality. As a solution, we explore a low-complexity decentralized algorithm, leveraging structural properties to obtain a suboptimal solution.

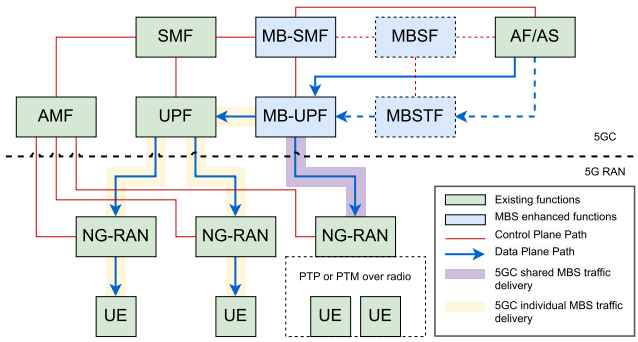


Fig. 1. 5G architecture with MBS enhancement and data transmission diagram [6]. Access and mobility management function (AMF), access stratum (AS).

4) We verify the optimal and suboptimal policies' structural properties via numerical simulations. We show that both our proposed policies consistently outperform the baseline strategies proposed in [14] with the assistance of AoI information at devices.

II. 5G MBS Architecture and Feedback Mechanisms

There are two delivery modes in 5G MBS: multicast and broadcast. For the broadcast mode, data is transmitted to all user equipment (UE) in coverage, regardless of radio resource control (RRC) states. However, the broadcast mode fails to manage UEs in the network, and there is no implemented feedback mechanism for the broadcast mode, making it hard to maintain reliability. Hence, in our work, we will focus on the multicast mode. The multicast delivery mode aims to provide high quality of service (QoS) services and ensure reliability for a set of registered UEs. We will discuss in more detail how reliability maintenance can be achieved through various feedback mechanisms.

The architecture of 5G MBS is listed in the specifications [6], [8], as depicted in Fig. 1. In addition to the original network functions in the 5G core (5GC) network, there are new network functions defined in the 5G MBS. We list the new functions for 5G MBS enhancement below.

1) *Multicast/Broadcast Session Management Function (MB-SMF):* MB-SMF manages sessions, configures MB-UPF for data transport based on policy rules, and optimizes broadcast and multicast MBS sessions with RAN and SMF.
2) *Multicast/Broadcast User Plane Function (MB-UPF):* MB-UPF processes packets, enforces QoS, and collaborates with MB-SMF for data delivery to RAN nodes or UPF.
3) *Multicast/Broadcast Service Function (MBSF):* MBSF bridges MBS and LTE MBMS, manages MBS sessions with application function (AF) and MB-SMF, and determines internet protocol (IP) multicast addresses.
4) *Multicast/Broadcast Service Transport Function (MBSTF):* MBSTF anchors MBS data traffic, sources IP multicast, and supports related applications, including framing and FEC.

For MBS data transmission, there are two possible methods between 5GC and next generation radio access network (NG-RAN).

1) *5GC Individual MBS Traffic Delivery Method:* The single copy of data received by MB-UPF is copied and distributed to individual UE through their dedicated packet data unit (PDU) sessions. Therefore, each such UE requires a PDU session associated with a multicast session.
2) *5GC Shared MBS Traffic Delivery Method:* The single copy of data is delivered directly to the NG-RAN, which then distributes it to one or more UEs. The NG-RAN has two possible delivery methods.
   a) *Point-to-point (PTP) delivery method:* Each UE receives a separate copy of the MBS data packet from the RAN node.
   b) *Point-to-multipoint (PTM) delivery method:* NG-RAN delivers one copy of the MBS data packets to UEs via radio.

While the design of the individual delivery mode seems redundant and less efficient in multicast, it is necessary for those legacy base stations that have not supported MBS yet. Especially when a UE moves from an MBS-capable base station (BS) to another that is incapable of MBS, the individual delivery mode is essential to guarantee continuity.

Three key layers ensure reliability for the 5G MBS service: 1) the physical/medium access control (PHY/MAC) layer based on HARQ, 2) the radio link control (RLC) layer based on automatic repeat-request (ARQ), and 3) the application layer-FEC (AL-FEC). For the PHY/MAC layer based on HARQ, the base station identifies the reception status of UEs through HARQ-ACK feedback, which comes in three variants. The first, acknowledgement/negative-acknowledgement (ACK/NACK)-based HARQ feedback, is suitable when the number of UEs is small and assigns a unique physical uplink control channel (PUCCH) resource to each UE. The second, NACK-only based HARQ-ACK feedback, shares the PUCCH feedback resource among all UEs, and feedback is given only when packet decoding fails. Although it lacks the ability to differentiate between UEs, it can prevent PUCCH overhead. Lastly, no HARQ-ACK feedback can be configured for PTM transmissions with lower reliability requirements to conserve PUCCH resources and reduce latency. The base station can switch between these dynamically, and the UE identifies the active scheme through RRC signaling or a downlink control indicator. Whenever the BS detects a NACK for a packet, it triggers retransmission for the failed packet.

For services that have extreme reliability requirements, PHY/MAC layer HARQ will not be sufficient. Therefore, acknowledged mode (AM) is supported by PTP, which means that ARQ is enabled in the PTP RLC, resulting in high reliability. However, for PTM transmission, only the unacknowledged mode (UM) is available. In addition to the higher design complexity associated with AM for PTM, enabling AM will also cause the performance of UEs with better channel environments to be affected by UEs with poorer environments. 5G MBS base stations can switch between PTP and PTM dynamically in order to meet QoS requirements. This is also a major feature of the 5G MBS standard.

Contrary to the previous two ARQ mechanisms, AL-FEC typically uses a fountain code, such as RaptorQ code, to ensure reliability. Since LTE MBMS, the use of fountain code in multicast has been widely regarded as a relatively efficient choice that can significantly improve performance degradation caused by bottlenecked UEs. In 5G MBS, AL-FEC is mainly implemented in MBSTF. Our work aims to provide a perspective different from traditional HARQ, to explore the possible impact of fountain code on the emerging indicator AoI in 5G MBS, and how to efficiently design its mechanism so that AoI can be optimized.

III. Related Work

Since the exploration of AoI in a simple first-come, first-served queuing model in [4], most research on AoI has focused on various applications of the queuing systems [15]. In addition, AoI analysis on different communication models, e.g., fading wireless channel [16], multihop wireless network [17], and carrier sensing random access protocols [18], has also been investigated recently. Other notable topics related to AoI include [19] in game theory and [20] in information-centric vehicular networks.

Consideration of a multiuser status update system is also a common model. However, it is important to note that not all such systems are multicast. Many systems focus on the scheduling policy of devices to optimize the average AoI. In these cases, they select appropriate devices for transmission among multiple transmitters or receivers. For example, Kadota et al. [21] studied the optimal policy to minimize AoI in a wireless broadcast network. Hsu et al. [22] considered a random packet arrival system with multisource and multiple users, and the results are further generalized by considering different queueing disciplines in [23]. Ceran et al. [24] studied the optimal scheduling problem with HARQ by different reinforcement learning methods. Zhang et al. [25] investigated AoI and peak age of information (PAoI) violation probabilities and distributions in an IoT multisource system with preemptive updates. They extend their analysis from single-source systems to account for preemptive disruptions, deriving models for the multisource scenario. Their approach introduces a novel minimization problem focusing on maximal violation probabilities, providing a more comprehensive understanding of timeliness optimization in such systems. Chen et al. [26] first analyzed the average PAoI and AoI in a dual update system with sensors having identical service time distributions. They then extended their study to consider sensors with different service time distributions, quantitatively comparing the AoI performance of dual-queue and single-queue systems. Their findings provide practical guidance for communication system design and resource allocation. Lang et al. [27] analyzed the average AoI in a remote monitoring system with multiple sensors updating from the same physical process. They employed the SHS method for analysis and found that the logarithm of average AoI decreases linearly with an increase in the logarithm of sensor numbers.

In contrast, the source will send a status update to all users in each time slot in the multicast scenario. Tang et al. [28] evaluated the optimal AoI performance in a multiuser system, comparing the broadcast and unicast transmission schemes across three packet management strategies when random packet arrivals are combined with finite blocklength coding. The remaining studies concerning AoI under multicast can be categorized into two classes. The first class, [29], [30], [31], [32], and [33], uses the assumption of shifted exponential service time as the delay time for each device. Each device is guaranteed to be able to receive a status update, while the transmission time differs among all devices, followed by an exponential time distribution with a shifted constant time. A common scheme to determine the transmission time is that the source will transmit a new update as long as k of the N users successfully receive a status update. It is called *earliest k stopping scheme* first proposed in [29]. The other class, [34], [35], and [36], focuses on optimizing the tradeoff between AoI and energy with different multicast system designs. It can be noticed that recent research rarely considers the effect of the feedback mechanism under multicast on AoI.

Because the source will decide whether to generate a new update in each time slot adaptively, the number of packets for each status update is not fixed. The studies [14], [37], [38], [39], [40], [41], [42], and [43] all focused on nonuniform status update sizes, but only Yates et al. [14] conducted a preliminary examination in the context of a multicast scenario. Yates et al. [14] analyzed the performance of AoI under two coding schemes: finite redundancy (FR) and infinite incremental redundancy (IIR). Unlike the work [14], our article assumes that devices can give feedback on the reception status of each packet so that the source can know all devices' current packet reception status. We investigate an optimal transmission strategy that can improve AoI more than FR or IIR in this case. In [37], the peak AoI in a large-scale wireless network using rateless code is analyzed considering sources' distribution. In addition to a different system setting compared to their study, our study focuses on the policy to minimize time-average AoI. Meng et al. [38] analyzed the upper bound of AoI using a new type of rateless code, spinal codes. The optimal packet preempting policy with nonuniform packet sizes for a Bernoulli source is considered in [39]. Wang et al. [40] studied the impact on two HARQ schemes with a multistate Markov source in space-air-ground-integrated networks. A joint design of sampling and scheduling policies to minimize AoI with nonuniform packet size is considered in [41], and Xie et al. [42] further extended the work by exploiting online policies. However, the work [41] and [42] concentrated on multiple physical processes and scheduling the best devices to transmit. Instead, we study the best time to preempt the packets under the multicast scenario when we use rateless code with a single process. Feng and Yang [43] considered the rateless code in an erasure channel with a single user and aim to find the optimal scheduling policy. Our system further depicts more special properties under the multicast scenario and proposes two practical algorithms as solutions. The closely related work is organized in Table I.

IV. System Model and Problem Formulation

In this section, we begin by outlining what we mean by a wireless multicast status update system in our study. Following that, we provide a comprehensive mathematical explanation of this system. We will also cover the system's state, including the

TABLE I
COMPARISON OF RELATED WORK

Paper	Year	Main target	Communication model[a]	Channel error[b]	Feedback[c]	Approach[d]
Our work	2023	Optimal preemptive policy for multicast with rateless code	1 to N, multicast	Non-uniform	Yes	MDP
Yates et al. [14]	2017	AoI analysis with given transmission schemes	1 to N, multicast	Uniform	Limited	Probability
Tang et al. [28]	2023	AoI performance using broadcast and unicast schemes with finite blocklength code	1 to N, multicast and unicast	Non-uniform	No	Probability
Zhong et al. [29]	2017	Optimal transmission time in a random delay network	1 to N, multicast	No	Yes	Probability
Li et al. [33]	2020	AoI and peak-AoI for earliest stopping k scheme subject to deadlines	1 to N, multicast	No	Yes	Probability
Chen et al. [32]	2020	Optimal transmission interval without feedback	1 to N, multicast	No	No	Probability
Feng et al. [43]	2019	Optimal transmission scheme with Rateless code	1 to 1, unicast	Yes	Yes	MDP
Wang et al. [40]	2022	Optimal transmission scheme for classical and IR HARQ with a multistate Markov source	1 to 1, unicast	Yes	Yes	MDP
Wang *et al.* [39]	2019	Optimal preemptive policy with packets of non-uniform sizes and random arrivals	1 to 1, unicast	No	No	MDP
Kadota et al. [21]	2018	Performance comparison of scheduling policies	1 to N, unicast	Non-uniform	Yes	DP, LO, RMAB
Ceran et al. [24]	2021	Optimal scheduling policy with HARQ	1 to N, unicast	Non-uniform	Yes	RL
Wei et al. [37]	2020	Peak-AoI with given coding schemes and networks	N to 1, unicast	Yes	Yes	Probability
Zhou et al. [41]	2020	Optimal scheduling and sampling policy with different packet sizes	N to 1, unicast	Non-uniform	Yes	MDP
Hsu et al. [22]	2020	Optimal scheduling policy with multiple sources	N to N, unicast	No	No	MDP

[a] 1 to N: There are one transmitter and multiple receivers; unicast: The packet is transmitted to only one receiver in each time slot; multicast: A packet is transmitted to all receivers.
[b]No: A packet will not be lost in the channel; Yes: There is only one receiver, and a packet may fail to be delivered due to channel noise; Uniform: All receivers have the same error probability; Nonuniform: The error probabilities can be different among all receivers.
[c]Limited: Feedback is only available after a full status update has been received; Yes: A receiver gives feedback every time when it receives a packet.
[d]DP: Dynamic programming; LO: Lyanopov optimization; RMAB: Rateless multiarmed bandit; RL: Reinforcement learning.

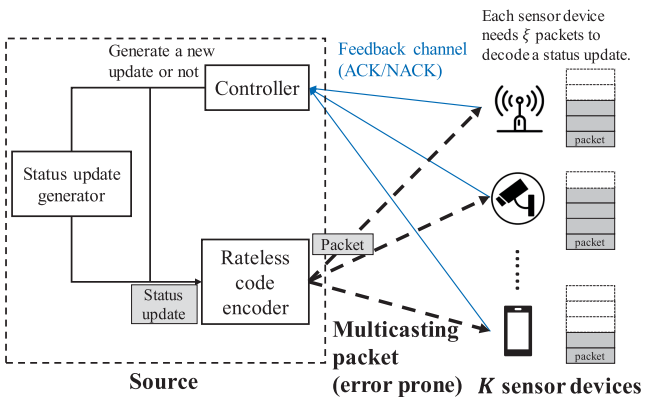


Fig. 2. Multicast-enabled WSN with rateless code.

AoI for both the status updates and the sensor devices, as well as the remaining packets that need to be received. Finally, we model the system's behavior and establish it as a MDP.

A. Wireless Multicast Status Update System

We consider a multicast-enabled WSN consisting of a source and K sensor devices, as shown in Fig. 2. The source sends status updates (e.g., humidity, temperature, location, etc.) to all other devices in the network via multicast. Since wireless transmission is susceptible to noise and interference, we use rateless code to protect status updates according to 3GPP recommendation [8] for multicast networks. Our assumption of rateless code is based on the practically implemented rateless code, RaptorQ code [12]. We assume the encoder can produce any number of encoded symbols on the fly from the status update as needed. Each symbol will have the same size and be transmitted as a MBS packet. In this discussion, it is imperative to distinguish between two key terms: a *status update* and *packets*. A *status update* refers to a complete snapshot of information that needs to be communicated from the source to the receiver. Each snapshot is then protected by rateless codes and transformed into smaller encoded units termed *packets*. It is assumed that each device is able to decode a status update after successfully receiving ξ packets. This uniform requirement of ξ packets for all devices is due to the nature of rateless codes. These codes allow for efficient decoding with a high probability upon receiving a slightly larger number of packets than the minimum required. Since all sensors request the same update and the decoding capability is not sensor-specific, ξ remains constant across different devices.

Each device sends an ACK or NACK through a perfect feedback channel, similar to the feedback channel of the HARQ process, to the BS after each packet is received. However, no feedback is needed whenever a status update is successfully decoded. This is to inform the source whether the packet was received successfully, but no retransmission will be triggered for NACK feedback. The controller in the source, guided by feedback from all devices, adaptively determines whether the next transmission should persist with the current status update so that more devices can decode correctly or whether it should generate a new status update but discard all packets associated with the previous status update. Our goal is to find the optimal strategy for the controller to minimize the sum of time-average AoI for all devices.

B. Mathematical Formulation of System Model

We consider a wireless multicast network consisting of one source and a set $\mathcal{K} = \{1, 2, \ldots, K\}$ of K devices. All of these

devices are keen to receive status updates. Let the time be slotted, indexed by $t = 1, 2, \ldots$, and the source can only transmit one packet in a time slot. Every sensor device can receive this packet within the duration of a time slot and will update its information when the slot concludes.[1] Consider a binary erasure channel. For device $k \in \mathcal{K}$, we assume it may fail to receive a packet with probability $p_k \in [0, 1)$, defined as the packet erasure rate of device k in our system. While the probability differs between devices, it does not vary with time. Let $u_k(t) \in \{0, 1\}$ denotes whether the device k receives a packet successfully in time slot t, where $u_k(t) = 1$ with probability $1 - p_k$ and $u_k(t) = 0$ with probability p_k. Each device cannot decode the status update until it successfully receives ξ packets, regardless of their order. The assumption of a rateless code led us to propose $\xi \geq 2$ in our work. The arrival model is a *generate-at-will* model (i.e., the source can generate a message at any time). As considered in [21], [33], and [41], we assume each device will give the source an ACK or NACK message via an error-free and instantaneous feedback channel. They do not have to report other information because the source can precisely track each device's status, as we will see in Section IV-C. This can help to reduce the uplink feedback overhead.

Upon receiving the feedback at the beginning of each time slot, the source is faced with a decision regarding the packet to transmit. The decision, denoted by $d(t)$, is about whether to *continue transmitting packets from the current status update* or to *commence with a new status update's packet*. Specifically, $d(t) = 0$ indicates the transmission of a packet from the ongoing status update, whereas $d(t) = 1$ signals the initiation of a new status update's packet transmission. Initiating a packet transmission from the new status update will render all remaining packets of the ongoing update outdated, prompting their discard across all devices. Since perfect feedback exists at the end of each time slot, the source can know the exact AoI at each device and how many packets each device still needs to decode the status update. This article aims to *find the best action for the source according to the current state, and thus, the time-average AoI performance is minimized.*

At the start of time slot t, we define the number of transmitted packets of the latest generated status update as $P(t)$, which represents the age of the status update. In addition, we define the age at the destination device k as $A_k(t)$ for each device. For $P(t)$ and $A_k(t)$, we further define their upper limit as $\hat{A}$. This approach is supported by the methodology adopted in related studies [22], [41], where finite upper limits are set to manage the computational feasibility of solving average-cost MDPs with potentially infinite states. In addition, this strategy addresses practical considerations; updates beyond a certain age threshold, $\hat{A}$, becomes extremely out-of-date and offers similar utility for the devices [44], [45]. Thus, imposing an upper limit on age, which may be considered large yet finite, ensures both computational tractability and practical utility. The state space of $P(t)$ and $A_k(t)$ are $\mathcal{P} \triangleq \{1, \ldots, \hat{A}\}$ and $\mathcal{A} \triangleq \{\xi, \xi + 1, \ldots, \hat{A}\}$, respectively. A device has to receive ξ packets to decode a status update, so its age will not be smaller than ξ. Thus, the smallest possible age of a destination will be ξ. $R_k(t) \in \mathcal{R} \triangleq \{0, 1, \ldots, \xi\}$ is used to denote the number of packets still needed for device k to decode the update at the beginning of the time slot t. If $R_k(t) = 0$, the device k has successfully decoded the most recent message, and there is no more packet required for device k before the advent of the next update.

[1]RaptorQ code has been shown to possess relatively low encoding and decoding complexity, which is shown to be linear. Therefore, we do not consider the decoding time in our work.

C. AoI Transition

By $X_k(t) \triangleq (A_k(t), R_k(t)) \in \mathcal{X}_k$, we define the system state vector of device k at slot t, and $\mathcal{X}_k \subset \mathcal{A} \times \mathcal{R}$ as the system state space of device k. Let $X(t) \triangleq (P(t), (X_k(t))_{k\in\mathcal{K}}) \in \mathcal{X}$ be the system state vector at slot t, and $\mathcal{X} \subset \mathcal{P} \times \prod_{k\in\mathcal{K}} \mathcal{X}_k$ is the system state space. Note that not all combinations of $P(t), X_k(t)$ belong to $\mathcal{X}$. We will discuss the formal definition of the system state space in Section V-A.

If $d(t) = 1$, the system will transform from the state $X(t) = (P(t), (X_k(t))_{k\in\mathcal{K}})$ to $X(t+1) = (P(t+1), (X_k(t+1))_{k\in\mathcal{K}})$ by the following:

$$P(t+1) = 1 \tag{1}$$

$$A_k(t+1) = [A_k(t) + 1]^+_{\hat{A}} \tag{2}$$

$$R_k(t+1) = \begin{cases} \xi - 1, & \text{if } u_k(t) = 1 \\ \xi, & \text{if } u_k(t) = 0 \end{cases} \tag{3}$$

where $[n]^+_{\hat{A}} \triangleq \min\{n, \hat{A}\}$ to ensure the age will not surpass the upper limit of AoI. Since the source selects to transmit the packet from a newly generated update, the age of the status update will reduce to one at the beginning of the next time slot. The age of the destination will increase by one because it cannot decode an update in the current time slot due to the assumption of $\xi \geq 2$. If the device k fails to receive the packet, the number of packages it needs to decode the status update, $R_k(t)$, will remain ξ. Otherwise, it will reduce to $\xi - 1$.

On the other hand, if $d(t) = 0$, the system will transform from the state $X(t)$ to $X(t+1)$ by the following:

$$P(t+1) = [P(t) + 1]^+_{\hat{A}} \tag{4}$$

$$X_k(t+1) = \begin{cases} \left([A_k(t) + 1]^+_{\hat{A}}, R_k(t)\right) \text{ if } u_k(t) = 0 \\ \left([A_k(t) + 1]^+_{\hat{A}}, \max\{R_k(t) - 1, 0\}\right) \\ \quad \text{if } u_k(t) = 1 \text{ and } R_k(t) \neq 1 \\ \left([P(t) + 1]^+_{\hat{A}}, 0\right) \\ \quad \text{if } u_k(t) = 1 \text{ and } R_k(t) = 1 \end{cases}. \tag{5}$$

Because the source decides to continue transmitting the packet from the previous update, the age of the message, $P(t)$, will increase by one. If the device k fails to receive the packet, its age will increase, and the number of packets needed to decode the status update remains. In the case of successfully receiving a packet, the device's age may be updated if it only needs one packet to decode the status update in this time slot ($R_k(t) = 1$). Interestingly, the age at the destination is the same as the number

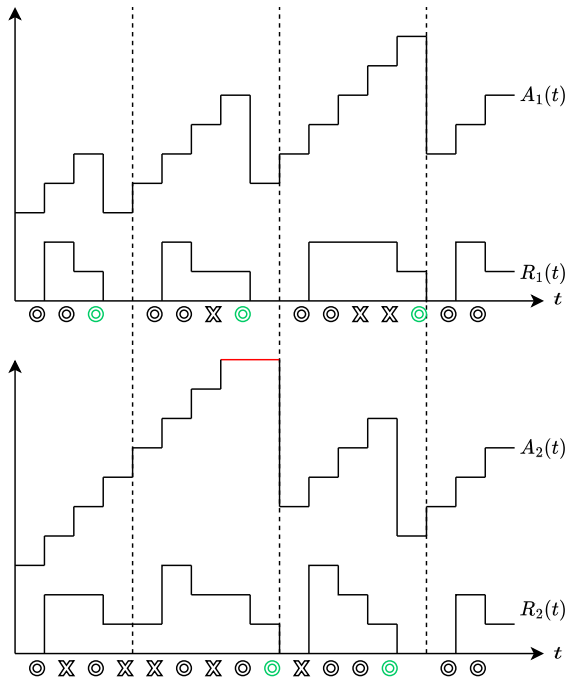


Fig. 3. Example illustration with system parameters $K = 2$, $\hat{A} = 10$, and $\xi = 3$. Both A_k and R_k are right-continuous. The circle and cross symbols indicate the ACK/NACK feedback. The green circles imply a device has decoded a status update, and thus AoI decreases. The red line means the upper limit of AoI. The source chooses to transmit a new update's packet at time slots $t = 1, 5, 10, 15$ in this example.

of packets that have been transmitted after successfully decoding a message. Conversely, if device k cannot decode a status update in the time slot ($R_k(t) \neq 1$), its age increases by one, and the number of packets required for decoding reduces by one. From the definition of $R_k(t+1)$, it will never be negative. Therefore, we take a maximum of zero and $R_k(t) - 1$. Fig. 3 illustrates an example of state transitions to help envision the above transition models. The key notations are listed in Nomenclature.

D. Markov Decision Process

Based on the AoI model, we can formulate our problem as a MDP Δ by the following components [46]:

1) *States:* At the beginning of each time slot t, we define the system state vector as follows:

$$\begin{aligned} X(t) &= (P(t), X_k(t)_{k\in\mathcal{K}}) \\ &= (P(t), (A_k(t), R_k(t))_{k\in\mathcal{K}}) \end{aligned} \tag{6}$$

where $P(t)$ is the number of packets transmitted for the latest update, and $X_k(t)$ is the state of the destination. $X_k(t)$ can be decomposed to $A_k(t)$ and $R_k(t)$, representing the age and the number of required packets at the destination, respectively. $\mathcal{X}$ is used to denote the state space, which is finite and bounded in our model.

2) *Actions:* At each time slot t, the source should select an action $d(t) \in \mathcal{D} = \{0, 1\}$, depending on the current state $X(t)$.

3) *Transition probabilities:* For simplicity, we omit the time slot t. By taking the action d, the transition probabilities from the state $X = (P, (X_k)_{k\in\mathcal{K}})$ to $X' = (P', (X'_k)_{k\in\mathcal{K}})$ can be written as below:

$$Pr\left[X'|X, d\right] = \prod_{k\in\mathcal{K}} Pr\left[X'_k|X_k, d\right]. \tag{7}$$

From (1) and (4), the transition of P is determined once the action d is selected. Therefore, it will not affect the transition probabilities, and we only need to consider the uncertainty of X_k. For each system state of device k, we have the following:

$$Pr[X'_k|X_k, d] = \begin{cases} 1 - p_k, & \text{if } X'_k = X_{k,s} \\ p_k, & \text{if } X'_k = X_{k,f} \\ 0, & \text{otherwise.} \end{cases} \tag{8}$$

Here, $X_{k,s}$ denotes the state where device k succeeds in receiving the packet, or it will transit to the state failing to get the packet, $X_{k,f}$. According to the (2), (3), and (5), we can describe them as follows:

$$X_{k,s} = \begin{cases} \left([A_k + 1]^+_{\hat{A}}, \xi - 1\right), & \text{if } d = 1 \\ \left([A_k + 1]^+_{\hat{A}}, \max\{R_k - 1, 0\}\right), & \\ \qquad \text{if } d = 0 \text{ and } R_k \neq 1 & \\ \left([P + 1]^+_{\hat{A}}, 0\right), & \\ \qquad \text{if } d = 0 \text{ and } R_k = 1 & \end{cases}. \tag{9}$$

$$X_{k,f} = \begin{cases} \left([A_k + 1]^+_{\hat{A}}, \xi\right), & \text{if } d = 1 \\ \left([A_k + 1]^+_{\hat{A}}, R_k\right), & \text{if } d = 0 \end{cases}. \tag{10}$$

4) *Costs:* The immediate cost of the MDP under state $X(t)$ and $C(X(t))$, is defined as the sum of AoI for all destinations. Thus, we have the following:

$$C(X(t)) = \sum_{k\in\mathcal{K}} A_k(t). \tag{11}$$

Then, we define the stationary and deterministic policy.

Definition 1: A *stationary* and *deterministic* policy π is a mapping from a state in the state space to a determined action. That is, $\pi : \mathcal{X} \to \mathcal{D}$. We define Π as a set of stationary deterministic policies for the above MDP.[2]

From [46], a policy may take action based on all of its historical states. However, finding such policies is challenging. Therefore, we will investigate the optimal policy within the space of Π. For a policy $\pi \in \Pi$, the long-term time-average AoI is as follows:

$$V^\pi(X_1) = \limsup_{T\to\infty} \frac{1}{T} \mathbb{E}_\pi \left[\sum_{t=1}^{T} C(X(t))|X(1) = X_1\right] \tag{12}$$

[2]In general, a policy $\pi = \{\mu_1, \mu_2, \ldots\}$ indicates the action for each time slot. Each $\mu_t \in \mathcal{M}$, usually referred to as a *decision rule*, is a mapping from a state X to an action d, and $\mathcal{M}$ is the set of mappings $\mu : \mathcal{X} \to \mathcal{D}$. Nevertheless, if we focus on *stationary* policies in the form $\pi = \{\mu^\star, \mu^\star, \ldots\}$, the action will only depend on the current state, and the influence of time can be ignored. Thus, we define the policy as a mapping from a state to an action in our discussion.

where $\mathbb{E}_\pi$ denotes the expected value under the policy π, and X_1 is the system state vector of the initial state. For simplicity, we suggest that $X_1 = (\xi, (\xi, 0)_{k\in\mathcal{K}})$ and eliminate X_1 subsequently.

Definition 2: A policy $\pi^\star \in \Pi$ is an *age-optimal policy* if it minimizes the long-term time-average AoI, as follows:

$$V^{\pi^\star} \triangleq \min_{\pi\in\Pi} V^\pi. \tag{13}$$

The *optimal average AoI* is $V^\star \triangleq V^{\pi^\star}$.

The objective of this article is to find an *age-optimal policy* $\pi^\star$ that minimizes the objective function V^π.

V. Optimal Policy and Structural Properties

In this section, we first delve into how the unique behavior of AoI in our system allows us to eliminate certain states. Next, we explore how to find the optimal policy by solving the Bellman equation through the relative value iteration algorithm (RVIA). However, facing the challenge of high complexity due to the curse of dimensionality, it becomes essential to identify structural properties that can simplify the algorithm. Finally, we integrate these insights to introduce a structure-aware RVIA, designed to efficiently determine the optimal policy.

A. State Space Reduction

Even though the state space size described above is large, some state vectors are unreasonable when we further explore the AoI transition model. Therefore, we shall exclude them from the state space in our discussion. All system state vectors should satisfy the following three propositions.

Proposition 1: For any $X \in \mathcal{X}$, if $\exists P \le \xi$, then $R_k \ge \xi - P \forall k \in \mathcal{K}$.

Proof: Intuitively, if we transmitted only $P \le \xi$ packets previously, the best-case scenario for device k is that all packets were successfully received, resulting in the minimum possible $R_k = \xi - P$. If $R_k > \xi - P$, it implies that one or more packets were lost.

Proposition 2: For any $X \in \mathcal{X}$, if $\exists k \in \mathcal{K}$ such that $R_k = 0$, then $P = A_k$. Otherwise, $A_k - P \ge \xi$ if $R_k > 0$ and $A_k \ne \hat{A}$.

Proof: According to the age transition model, if $R_k = 0$, it should have passed the transition with $X_k = (A_k, 1)$, action $d = 0$, and $u_k = 1$ previously. From (4) and (5), the new state after the transition is $P' = [P+1]^+_{\hat{A}}$ and $X'_k = ([P+1]^+_{\hat{A}}, 0)$, where $P' = A'_k$ and $R'_k = 0$. If the source has not yet generated a new status update, $R_k = 0$ remains with P and A_k increasing by one simultaneously; thus, $P = A_k$ is maintained.

On the other hand, if the source decides to generate a new status update, according to (1) and (2), the new state has $P' = 1$ and $A'_k = \min(\hat{A}, A_k + 1)$. Since the minimum of A_k is ξ, we have $A'_k - P' \ge \xi$. Hereafter, P and A_k will grow together before R_k becomes 0, so $A'_k - P' \ge \xi$ holds before A_k attains $\hat{A}$. If $A_k = \hat{A}$, A_k will not grow, but P may increase, so we cannot guarantee the inequality holds true. □

Proposition 3: For any $X \in \mathcal{X}$, if $\exists k, k' \in \mathcal{K}$, such that $A_k \ne \hat{A}$, $A_{k'} \ne \hat{A}$, and $A_k \ne A_{k'}$, then $|A_k - A_{k'}| \ge \xi$.

Proof: Assume $d = 1$ is taken on some states $X \in \mathcal{X}$. Then, we define the state in the next time slot as $\bar{X} = (1, (\bar{X}_k)_{k\in\mathcal{K}})$, where the bar is denoted as the initial state here. From (2), $\bar{A}_k \ge \xi + 1\ \forall k \in \mathcal{K}$. Note that the difference in A_k between the two devices remains if neither of them successfully decodes the update. Therefore, we assume device k' has successfully received ξ packets after $t-1$ transmissions of the same update, and we have the state $X = (t, (X_k)_{k\in\mathcal{K}})$ with $A_{k'} = t$. We now analyze the difference of AoI between device k and k', that is, the relationship of A_k and $A_{k'}$.

The first scenario is that device k also successfully decoded the message in this time slot. Therefore, $A_k = A_{k'} = t$. The second scenario is device k did not decode in this time slot, but $\hat{A} \ge t \ge \hat{A} - \xi$. In this case, since $\bar{A}_k \ge \xi + 1$ initially, $A_k = [\bar{A}_k + t - 1]^+_{\hat{A}} = \hat{A}$. Therefore, we have $\xi \ge A_k - A_{k'} \ge 0$. Finally, if $t < \hat{A} - \xi$ and the device k did not decode the message in this time slot, we have $A_k = \bar{A}_k + t - 1$. Hence, $A_k - A_{k'} = \bar{A}_k - 1 \ge \xi$. The difference will remain until the next time of successful message decoding, or one of the device's age attains $\hat{A}$.

In conclusion, if $\exists k, k' \in \mathcal{K}$, such that $A_k \ne \hat{A}$, $A_{k'} \ne \hat{A}$, and $A_k \ne A_{k'}$, it implies the difference is caused by the third scenario, and thus $|A_k - A_{k'}| \ge \xi$.

Definition 3: A *system state vector space* $\mathcal{X}$ is composed of all possible combinations of $(P, (A_k, R_k)_{k\in\mathcal{K}}) \in \mathcal{P} \times \prod_{k\in\mathcal{K}} (\mathcal{A} \times \mathcal{R})$, and the combinations should satisfy the three propositions.

B. Bellman Equation

After clearly defining the MDP formulation of this problem, we can obtain an age-optimal policy by solving the Bellman equation.

Lemma 1: There exists a scalar λ and a function $h(X)$ that satisfy the following Bellman equation:

$$\lambda + h(X) = \sum_{k\in\mathcal{K}} A_k + \min_{d\in\mathcal{D}} \sum_{X'\in\mathcal{X}} Pr\left[X'|X, d\right] h(X') \tag{14}$$

where λ is the value of optimal average AoI. λ is independent of the initial state and can be written formally as follows:

$$\lambda = \lim_{\gamma\to 1}(1-\gamma)V_\gamma(X) \quad \forall X \in \mathcal{X} \tag{15}$$

and $h(X)$ is the relative value function for any given fixed state o, satisfies

$$h(X) = \lim_{\gamma\to 1}(V_\gamma(X) - V_\gamma(o)) \quad \forall X \in \mathcal{X} \tag{16}$$

where $V_\gamma(X)$ is the value of the state X under the discount factor γ and the optimal policy π, given as follows:

$$V_\gamma(X) = \min_{\pi\in\Pi} \limsup_{T\to\infty} \frac{1}{T} E_\pi \left[\sum_{t=1}^{T} \gamma^{t-1} C(X(t)) | X(1) = X\right]. \tag{17}$$

In addition, if a policy $\pi^\star(X)$ attains the minimum of the right-hand side in (14) for all X, that is,

$$\pi^\star(X) = \arg\min_{d\in\mathcal{D}} \sum_{X'\in\mathcal{X}} Pr\left[X'|X, d\right] h(X') \quad \forall X \in \mathcal{X} \tag{18}$$

then the stationary policy $\pi^\star$ is age-optimal.

Proof: In our problem, we have to prove the MDP is *communicating*. That is, for every two states $X, X' \in \mathcal{X}$, there

exists a stationary policy f such that for some t, we have $Pr[X(t) = X'|X(0) = X, f] > 0$. By [47, Def. 4.2.2, Prop. 4.2.1 and 4.2.3], if a MDP is communicating, the weak accessibility condition holds for the MDP. Then, we can prove the optimal average AoI is the same for all initial states, so (14) to (17) hold. Besides, [46, Corollary 8.2.5] uses Laurent series expansion of V_γ to prove $\lambda = V^\star(X_1) \forall X_1 \in \mathcal{X}$. We can also find an age-optimal policy from the Bellman equation in (18), according to [46, Th. 8.4.3]. We put our proof about the communicating property of MDP in Appendix A.

From (18), the optimal policy depends on the relative value function $h(X)$. However, there is generally no closed-form solution for $h(X)$. Alternatively, we tend to perform the RVIA [48, Sec. 3.4.2] to solve $h(X)$ numerically. By initializing $h_0(X) = 0$ for all $X \in \mathcal{X}$ and starting with any fixed state $o \in \mathcal{X}$, a single iteration in RVIA is as follows:

$$Q_{n+1}(X, d) = \sum_{k \in \mathcal{K}} A_k + \sum_{X' \in \mathcal{X}} Pr\left[X'|X, d\right] h_n(X') - h_n(o) \tag{19}$$

$$h_{n+1}(X) = \min_{d \in \mathcal{D}} Q_{n+1}(X, d) \tag{20}$$

where $Q_n(X, d)$ and $h_n(X)$ are Q-factor and relative value function at iteration n, respectively. As described in [46, Sec. 9.5.3], $h_{n+1}(X) - h_n(X)$ will converge to 0 if the MDP is weakly communicating.[3] That is

$$\lim_{n \to \infty} h_n(X) = h(X) \quad \forall X \in \mathcal{X}. \tag{21}$$

Equation (21) also implies the convergence of $Q_n(X, d)$. Hence, we have $\lim_{n\to\infty} Q_n(X, d) = Q(X, d) \forall X \in \mathcal{X}, d \in \mathcal{D}$. Then, we can rewrite (14) as follows:

$$\lambda + Q(X, d) = \sum_{k \in \mathcal{K}} A_k + \sum_{X' \in \mathcal{X}} Pr\left[X'|X, d\right] h(X') \tag{22}$$

where $Q(X, d)$ can be called as Q-factor. Furthermore, the optimal average AoI is the relative value function of the fixed state, $\lambda = h(o)$, by comparing (19) and (22).

C. Structural Properties

In this section, we will further investigate the special structural properties in the MDP Δ to reduce the complexity of the RVIA. We define $\mathbf{A} = (A_k)_{k \in \mathcal{K}}$ and $\mathbf{R} = (R_k)_{k \in \mathcal{K}}$ as the vectors of A_k and R_k for all $k \in \mathcal{K}$. The symbol $\preceq$ denotes vector inequality in $\mathbb{R}^K$: $\mathbf{A}^1 \preceq \mathbf{A}^2$ means $A_k^1 \leq A_k^2$ for all $k \in \mathcal{K}$. Then, we start with the property of the relative value function.

Lemma 2: For any $X^1 = (P^1, \mathbf{A}^1, \mathbf{R}^1), X^2 = (P^2, \mathbf{A}^2, \mathbf{R}^2) \in \mathcal{X}$, if they satisfy $P^1 \leq P^2, \mathbf{A}^1 \preceq \mathbf{A}^2$ and $\mathbf{R}^1 \preceq \mathbf{R}^2$, then $h_n(X^1) \leq h_n(X^2)$ for all n. Hence, $h(X^1) \leq h(X^2)$.

Proof: See Appendix B. □

Lemma 2 indicates the relative value function of the state is monotonically increasing in P, $\mathbf{A}$ and $\mathbf{R}$. This implies state $(\hat{A}, (\hat{A}, \xi)_{k \in \mathcal{K}})$ is the worst state among all possible states. The lemma is essential to prove Theorem 1 and the structural properties of the optimal policy, as we can see in the following proof.

Theorem 1: For any $X^1 = (P^1, \mathbf{A}^1, \mathbf{R}^1), X^2 = (P^2, \mathbf{A}^2, \mathbf{R}^2) \in \mathcal{X}$, if they satisfy $\mathbf{A}^1 = \mathbf{A}^2$, then $Q_n(X^1, 1) = Q_n(X^2, 1)$ for all n. Therefore, $Q(X^1, 1) = Q(X^2, 1)$. Moreover, if $P^1 \leq P^2$ and $\mathbf{R}^1 \preceq \mathbf{R}^2$, $Q_n(X^1, 0) \leq Q_n(X^2, 0)$ for all n. Hence, $Q(X^1, 0) \leq Q(X^2, 0)$.

Proof: See Appendix C. □

When calculating the Q-factor, applying this theorem can reduce the complexity of RVIA. Originally, one has to calculate $Q_n(X, 1)$ for all $X \in \mathcal{X}$ in each iteration. However, according to Theorem 1, states with the same vector $\mathbf{A}$ will have the same $Q_n(X, 1)$ regardless of P and $\mathbf{R}$. Therefore, we can first find all possible values of $Q_n(X, 1)$ according to different $\mathbf{A}$ and store the results in a table indexed by $\mathbf{A}$. Then, when we need to update $Q_n(X, 1)$, we can do so from the table. This property will be exploited in Section VI-B. By combining Lemma 2 and Theorem 1, we can further obtain the structural property of the optimal policy.

Theorem 2: For any $X^1 = (P^1, \mathbf{A}^1, \mathbf{R}^1), X^2 = (P^2, \mathbf{A}^2, \mathbf{R}^2) \in \mathcal{X}$ with $P^1 \leq P^2, \mathbf{A}^1 = \mathbf{A}^2$, and $\mathbf{R}^1 \preceq \mathbf{R}^2$, if $\pi^\star(X^2) = 0$, then $\pi^\star(X^1) = 0$. Similarly, if $\pi^\star(X^1) = 1$, then $\pi^\star(X^2) = 1$.

Proof: From Theorem 1, we have $Q(X^1, 1) = Q(X^2, 1)$ and $Q(X^1, 0) \leq Q(X^2, 0)$. This follows the inequality:

$$Q\left(X^1, 0\right) - Q\left(X^1, 1\right) \leq Q\left(X^2, 0\right) - Q\left(X^2, 1\right). \tag{23}$$

In addition, $\pi^\star(X^2) = 0$ implies $Q(X^2, 0) - Q(X^2, 1) \leq 0$. Combining the above inequality, we obtain $Q(X^1, 0) - Q(X^1, 1) \leq 0$, which is equal to $\pi^\star(X^1) = 0$.

For the case that if $\pi^\star(X^1) = 1$, then $\pi^\star(X^2) = 1$, we can multiply -1 to (23) and applying $Q(X^1, 1) - Q(X^1, 0) \leq 0$. Hereafter, the proof is completed. □

Theorem 2 reveals the unique structure of this system and enables us to improve efficiency when executing RVIA. If we can find the optimal action for a state, then many optimal actions for other states can be derived from this theorem. Moreover, Proposition 4 indicates another structural property for an age-optimal policy.

Proposition 4: If a state $X \in \mathcal{X}$ has $\mathbf{R} = \boldsymbol{\xi}$ or $\mathbf{R} = \mathbf{0}$, $\pi^\star(X) = 1$.

Proof: See Appendix D. □

In addition to the rigorous proof, we can intuitively understand the above proposition. If all devices fail to receive a packet or successfully decode an update, the source should start transmitting a new status update. We can use this property to simplify the calculation of the optimal policy for some states.

D. Solution of the Optimal Policy With Structural Properties

Theoretically, we can obtain $h(X)$ for all X by RVIA and thus obtain the optimal policy. However, performing

[3] From [46] and [49], the sufficient condition for the convergence of RVIA is the aperiodicity of the Markov chain induced by the age-optimal policy $\pi^\star$. However, we can apply the aperiodicity transform mentioned in [46, Sect. 8.5.4] to ensure the RVIA is available. Therefore, the proof of aperiodicity of the Markov chain is not necessary here.

Algorithm 1: Structure-Aware RVIA.

1 **Initialize**: $h_0(X)=0, \forall X \in \mathcal{X}$, a fixed state $o \in \mathcal{X}$, $n=0$ and a tolerance $\epsilon > 0$
2 **do**
3 **for** ***each*** $X=(P,\mathbf{A},\mathbf{R}) \in \mathcal{X}$ **do**
4 **if** $\mathbf{R}=\boldsymbol{0}$ *or* $\mathbf{R}=\boldsymbol{\xi}$ **then**
5 $\pi_n^\star(X) \leftarrow 1$
6 **else if** *there exists a non-negative integer* c *and* K*-dimensional non-negative integers* $\boldsymbol{c}$ *such that* $\pi^\star(P+c,\mathbf{A},\mathbf{R}+\boldsymbol{c})=0$ **then**
7 $\pi_n^\star(X) \leftarrow 0$
8 **else if** *there exists a non-negative integer* c *and* K*-dimensional non-negative integers* $\boldsymbol{c}$ *such that* $\pi_n^\star(P-c,\mathbf{A},\mathbf{R}-\boldsymbol{c})=1$ **then**
9 $\pi_n^\star(X) \leftarrow 1$
10 **else**
11 $\pi_n^\star(X) \leftarrow \arg\min_{d\in\mathcal{D}} \sum_{X'\in\mathcal{X}} Pr[X'|X,d] h_n(X')$
12 $h_{n+1}(X) \leftarrow \sum_{k\in\mathcal{K}} A_k + \sum_{X'\in\mathcal{X}} Pr[X'|X,\pi_n^\star(X)] h_n(X') - h_n(o)$
13 $n \leftarrow n+1$
14 **while** $\max_{X\in\mathcal{X}} |h_n(X)-h_{n-1}(X)| > \epsilon$;
15 $\pi^\star \leftarrow \pi_n^\star$

the algorithm will take a considerable amount of time due to the curse of dimensionality [48]. As a result, we begin by applying the structural properties to reduce its complexity.

The structure-aware RVIA is proposed in Algorithm 1. The algorithm takes advantage of the Proposition 4 and Theorem 2. In Algorithm 1, we initialize the relative value function for all states to zero and select a fixed state o and tolerance ϵ. Hereafter, in each iteration, we seek the optimal action $\pi^\star(X)$ for each state $X \in \mathcal{X}$. If the state has the vector $\mathbf{R}=\mathbf{0}$ or $\mathbf{R}=\boldsymbol{\xi}$, the optimal action can be directly determined by Proposition 4. Otherwise, we will check whether we can apply the structural properties in Theorem 1. We can identify its optimal action if the structural properties are available, as in Lines 7 and 9. Then, if we can not apply all structural properties, we will calculate it directly in Line 11. After determining the optimal action for state X, we can update the relative value function of state X in Line 12. The algorithm will continue until the maximum difference between the value of the relative value function in the current and that in the last iteration is less than ϵ. From (21), this algorithm is guaranteed to converge.

Upon calculating the optimal action for all possible states before transmission begins, the source can determine its action at time slot t based on the state $X(t)$. Nevertheless, since we have to iterate through all possible states, the complexity is still high at $\mathcal{O}(\prod_{k\in\mathcal{K}} \hat{A}(\hat{A}-\xi+1)(\xi+1))$. The algorithm may not be feasible as the number of devices rises because the time complexity increases exponentially. Hence, we investigate an approximation method in the next section to overcome this issue.

VI. Low Complexity Suboptimal Solution

A. Decomposition of Bellman Equation

To resolve the curse of dimensionality, we will approximate $Q(X,d)$. Before the approximation, we first define $\widetilde{X}_k \triangleq (P,X_k)$ as the *extended state vector of device* k. $\widetilde{X}_k \in \widetilde{\mathcal{X}}_k \subset \mathcal{P}\times\mathcal{X}_k$ is the extended state space of device k. The concept behind this definition is that a device k should know the age of the status update P to update its local age A_k. However, only the source knows the system state of all devices. Therefore, we can consider the extended state as local information for a single device. Based on the extended state of device k, we formulate a new MDP denoted as Δ_k for each device k. It can be regarded as the original MDP Δ with $K=1$ and packet erasure rate p_k, comprising the following components:

1) *States:* At the beginning of each time slot t, the extended state of device k can be described as follows:
$$\widetilde{X}_k(t)=(P(t),X_k(t))=(P(t),A_k(t),R_k(t)). \quad (24)$$
2) *Actions:* At each time slot t, the source will select a decision $d(t)\in\mathcal{D}=\{0,1\}$.
3) *Transition probabilities:* Since there is only one device, the transition probability can be simply written as follows:
$$Pr\left[\widetilde{X}_k'|\widetilde{X}_k,d\right]=\begin{cases}1-p_k, & \widetilde{X}_k'=\widetilde{X}_{k,s}\\ p_k, & \widetilde{X}_k'=\widetilde{X}_{k,f}\\ 0, & \text{otherwise.}\end{cases} \quad (25)$$
Because the transition of the packet's age is deterministic, the transition probability will not be affected by P. Hence, (25) essentially has the same structure as in (8). We only need to slightly modify the states indicating successful and failed reception as follows:
$$\widetilde{X}_{k,s}=\begin{cases}\left(1,[A_k+1]_{\hat{A}}^+,\xi-1\right), & \text{if } d=1\\ \left([P+1]_{\hat{A}}^+,[A_k+1]_{\hat{A}}^+,\max\{R_k-1,0\}\right), & \text{if } d=0 \text{ and } R_k\neq 1\\ \left([P+1]_{\hat{A}}^+,[P+1]_{\hat{A}}^+,0\right), & \text{if } d=0 \text{ and } R_k=1\end{cases} \quad (26)$$
$$\widetilde{X}_{k,f}=\begin{cases}\left(1,[A_k+1]_{\hat{A}}^+,\xi\right), & \text{if } d=1\\ \left([P+1]_{\hat{A}}^+,[A_k+1]_{\hat{A}}^+,R_k\right), & \text{if } d=0.\end{cases} \quad (27)$$
4) *Costs:* The immediate cost of the new MDP under state $\widetilde{X}_k(t)$ is simply the current AoI of the device.
$$C\left(\widetilde{X}_k(t)\right)=A_k(t). \quad (28)$$

Remark 1: The new MDP Δ_k can be viewed as the original MDP Δ with only one device ($K=1$) in the network. The

device's packet erasure rate equals p_k. Therefore, the above propositions, lemmas, and theorems in Δ also hold for Δ_k.

Notice that our target is not to find the optimal policy for Δ_k. However, the new MDP Δ_k can take advantage of the properties we obtained in the original MDP Δ. Now, we approximate $Q(X,d)$ as follows:

$$Q(X,d) \approx \sum_{k\in\mathcal{K}} Q_k\left(\widetilde{X}_k, d\right) \tag{29}$$

where $Q_k(\widetilde{X}_k, d)$ satisfies

$$\lambda_k + Q_k\left(\widetilde{X}_k, d\right) = A_k + \sum_{\widetilde{X}'_k\in\widetilde{\mathcal{X}}_k} Pr\left[\widetilde{X}'_k|\widetilde{X}_k, d\right] \min_{d'\in\mathcal{D}} Q(\widetilde{X}'_k, d') \quad \forall \widetilde{X}_k \in \widetilde{\mathcal{X}}_k. \tag{30}$$

We can guarantee (30) holds by applying Lemma 1 and Remark 1. Hereafter, we propose a suboptimal policy under the approximation as follows:

$$\widetilde{\pi}^\star(X) = \arg\min_{d\in\mathcal{D}} \sum_{k\in\mathcal{K}} Q_k(\widetilde{X}_k, d). \tag{31}$$

The suboptimal policy now depends on $Q_k(\cdot,\cdot)$. Because (30) is essentially a Bellman equation, we can also solve $Q_k(\cdot,\cdot)$ by RVIA like an analogy of (19) and (20), where

$$Q_k^{n+1}(\widetilde{X}_k, d) = A_k + \sum_{\widetilde{X}'_k\in\widetilde{\mathcal{X}}_k} Pr\left[\widetilde{X}'_k|\widetilde{X}_k, d\right] h_k^n(\widetilde{X}'_k) - h_k^n(o) \tag{32}$$

$$h_k^{n+1}\left(\widetilde{X}_k\right) = \min Q_k^{n+1}(\widetilde{X}_k) \tag{33}$$

where $Q_k^n(X,d)$ and $h_k^n(X)$ are Q-factor and relative value function at iteration n for Δ_k, and $o \in \widetilde{\mathcal{X}}_k$ is prescribed. We can guarantee the convergence of each $Q_k(\widetilde{X}_k, d)$ based on (21). By repeating this process for different p_k, we can obtain the suboptimal policy. The structural properties of the optimal policy are not available here. This is because the optimal decision for k might not be optimal for the entire system. Both the Q-factors for decisions 0 and 1 are required for the source to determine the suboptimal policy, as indicated by (31). That is, Lines 4 to 11 will be ignored, and Line 12 will be replaced by calculating the Q-factor for both decisions 0 and 1 in Algorithm 1. Interestingly, the suboptimal policy also has the same structure properties as Theorem 2.

Corollary 1: For any $X^1 = (P^1, \mathbf{A}^1, \mathbf{R}^1), X^2 = (P^2, \mathbf{A}^2, \mathbf{R}^2) \in \mathcal{X}$ with $P^1 \le P^2$, $\mathbf{A}^1 = \mathbf{A}^2$, and $\mathbf{R}^1 \preceq \mathbf{R}^2$, if $\widetilde{\pi}^\star(X^2) = 0$, then $\widetilde{\pi}^\star(X^1) = 0$. Similarly, if $\widetilde{\pi}^\star(X^1) = 1$, then $\widetilde{\pi}^\star(X^2) = 1$.

Proof: For every $\widetilde{X}_k^1, \widetilde{X}_k^2$ derived from X^1, X^2, they also have $P^1 \le P^2$, $A_k^1 = A_k^2$, and $R_k^1 \le R_k^2$. Hence, from Theorem 1, $Q_k(\widetilde{X}_k^1, 0) \le Q_k(\widetilde{X}_k^2, 0)$ and $Q_k(\widetilde{X}_k^1, 1) = Q_k(\widetilde{X}_k^2, 1)$. Since the result holds for all devices $k \in \mathcal{K}$, the inequality remains when we add them up. Furthermore, if $\widetilde{\pi}^\star(X^2) = 0$, it implies $\sum_{k\in\mathcal{K}} Q_k(\widetilde{X}_k^2, 1) \ge \sum_{k\in\mathcal{K}} Q_k(\widetilde{X}_k^2, 0)$. By $\sum_{k\in\mathcal{K}} Q_k(\widetilde{X}_k^1, 1) = \sum_{k\in\mathcal{K}} Q_k(\widetilde{X}_k^2, 1) \ge \sum_{k\in\mathcal{K}} Q_k(\widetilde{X}_k^2, 0) \ge \sum_{k\in\mathcal{K}} Q_k(\widetilde{X}_k^1, 0)$, we thus get $\widetilde{\pi}^\star(X^1) = 0$. The structural property of suboptimal policy for Action 1 can be proven in a similar way. □

We will further explain how to apply the structural properties for calculating the Q-factor and the suboptimal policy to reduce the algorithm's complexity in the next section.

B. Decentralized Structure-Aware Low-Complexity RVIA

Based on (32), (33), and Theorem 1, we propose a decentralized structure-aware low-complexity RVIA for the suboptimal policy, as shown in Algorithm 2. This algorithm can be separated into two phases. From (31), we know the suboptimal policy can be obtained by calculating the Q-factors under different separated extended states of each device k. Therefore, in the first phase, we obtain the Q-factors of each Δ_k. Since we aim to get the Q-factors instead of the optimal policy under Δ_k, Theorem 2 cannot be applied. However, we can utilize Theorem 1. We have proven that for all states with the same age vectors, the Q-factors for action 1 are the same. Therefore, we can first define some reference extended states with different age vectors[4] and find their Q-factors, as we can see in Lines 3 to 5. The calculated results are stored in a table $f_k^n[\cdot]$, and will be assigned to the corresponding state in Line 8. By doing so, we only have to perform $\hat{A} - \xi + 1$ times of calculation to obtain $Q_k^n(\widetilde{X}_k, 1)$ for all $\widetilde{X}_k \in \widetilde{\mathcal{X}}_k$.

The complexity of the low-complexity algorithm for the source reduces from $\mathcal{O}(\prod_{k\in\mathcal{K}} \hat{A}(\hat{A}-\xi+1)(\xi+1))$ to $\mathcal{O}(\sum_{k\in\mathcal{K}} \hat{A}(\hat{A}-\xi+1)(\xi+1))$. Moreover, as we mentioned that $\widetilde{X}_k$ can be considered local information for device k, we find that the computation is not restricted to perform on the source. Instead, it is possible to implement this computation on the device side, especially since many devices have the ability to perform fundamental calculations nowadays. This decentralized feature, which allows parallel computing, further reduces execution time. If the computation can be finished on each device, the source does not need to perform any calculation in this phase, and the complexity for every single device is $\mathcal{O}(\hat{A}(\hat{A}-\xi+1)(\xi+1))$.

If we adopt the decentralized algorithm, the Q-factors are stored in each device. Therefore, devices shall report the results of Q-factors to the source. Then, the source computes the suboptimal policy based on the Q-factors in phase 2, Line 21. Since the structural property of the suboptimal policy exists, the source does not need to add Q-factors for all states to get the optimal action, as shown in Lines 14 to 19.

Remark 2: Although the original RVIA has high complexity, we emphasize the value of studying the optimal policy. The algorithm serves as a theoretical benchmark, providing insights into the performance limits of low-complexity methods. Moreover, it aids in designing low-complexity RVIA by providing theoretical support. In addition, the optimal policy reveals key

[4]In practice, we can set a reference extended state $\widetilde{X}_{k,\text{ref}} = ((\xi-1)\mathbb{1}(A_k = \xi)+1, A_k, \xi(1-\mathbb{1}(A_k=\xi)))$, because the only possible extended state is $(\xi,\xi,0)$ if $A_k = \xi$. Otherwise, we can select the reference extended state as $(1, A_k, \xi)$. Based on Propositions 1–3, we can verify the arrays are in the extended state space.

Algorithm 2: Decentralized Structure-Aware Low-Complexity RVIA.

PHASE 1: RVIA at each device k based on Δ_k: Before the source starts the transmission, each device k calculates its Q-factors $Q_k(\widetilde{X}_k, d)$ separately:

Initialize: $h_k^0(\widetilde{X}_k) = 0, \forall \widetilde{X}_k \in \widetilde{\mathcal{X}}_k$, a fixed state $o \in \widetilde{\mathcal{X}}_k$, $n = 0$ and tolerance $\epsilon > 0$
do
  for ***each*** $A_k \in \mathcal{A}$ **do**
    Select a reference extended state $\widetilde{X}_{k,\text{ref}} = (P_0, A_k, R_{k,0}) \in \widetilde{\mathcal{X}}_k$;
    $f_k^{n+1}[A_k] \leftarrow A_k + \mathbb{E}[h_k^n(\widetilde{X}_k')|\widetilde{X}_{k,\text{ref}}, d = 1] - h_k^n(o)$;
  for ***each*** $\widetilde{X}_k \in \widetilde{\mathcal{X}}_k$ **do**
    $Q_k^{n+1}(\widetilde{X}_k, 0) \leftarrow A_k + \mathbb{E}[h_k^n(\widetilde{X}_k')|\widetilde{X}_k, d = 0] - h_k^n(o)$;
    $Q_k^{n+1}(\widetilde{X}_k, 1) \leftarrow f_k^{n+1}[A_k]$;
    $h_k^{n+1}(\widetilde{X}_k) \leftarrow \min_{d\in\mathcal{D}} Q_k^{n+1}(\widetilde{X}_k, d)$;
  $n \leftarrow n + 1$
while $\max_{\widetilde{X}_k \in \widetilde{\mathcal{X}}_k} |h_k^n(\widetilde{X}_k) - h_k^{n-1}(\widetilde{X}_k)| > \epsilon$;
$Q_k(\widetilde{X}_k, d) \leftarrow Q_k^n(\widetilde{X}_k, d)\ \forall \widetilde{X}_k \in \widetilde{\mathcal{X}}_k, d \in \mathcal{D}$

PHASE 2: The source obtains the suboptimal policy: The source acquires the suboptimal policy based on reported Q-factors $Q_k(\widetilde{X}_k, d)$ from devices:

for ***each*** $X \in \mathcal{X}$ **do**
  if $\mathbf{R} = \boldsymbol{0}$ *or* $\mathbf{R} = \boldsymbol{\xi}$ **then**
    $\widetilde{\pi}^\star(X) \leftarrow 1$;
  else if *there exists a non-negative integer* c *and* K*-dimensional non-negative integers* $\boldsymbol{c}$ *such that* $\widetilde{\pi}^\star(P + c, \mathbf{A}, \mathbf{R} + \boldsymbol{c}) = 0$ **then**
    $\widetilde{\pi}^\star(X) \leftarrow 0$
  else if *there exists a non-negative integer* c *and* K*-dimensional non-negative integers* $\boldsymbol{c}$ *such that* $\widetilde{\pi}^\star(P - c, \mathbf{A}, \mathbf{R} - \boldsymbol{c}) = 1$ **then**
    $\widetilde{\pi}^\star(X) \leftarrow 1$
  else
    $\widetilde{\pi}^\star(X) \leftarrow \arg\min_{d\in\mathcal{D}} \sum_{k\in\mathcal{K}} Q_k(\widetilde{X}_k, d)$

Algorithm 3: Numerical Method for Searching Optimal FR Strategy Threshold.

Result: The optimal FR strategy threshold $P^\star$ such that the average AoI can be minimized.

minAoI $\leftarrow \infty$;
$P^\star \leftarrow 0$;
for $n \leftarrow \xi$ **to** $\hat{A}$ **do**
  sumAoI $\leftarrow 0$;
  for $k \in \mathcal{K}$ **do**
    sumAoI $\leftarrow$ sumAoI $+ V^{\pi_{\text{FR}}}(n, p_k)$;
  currAoI $\leftarrow \frac{\text{sumAoI}}{K}$;
  if *currAoI* $<$ *minAoI* **then**
    minAoI $\leftarrow$ currAoI ;
    $P^\star \leftarrow n$;
return $P^\star$;

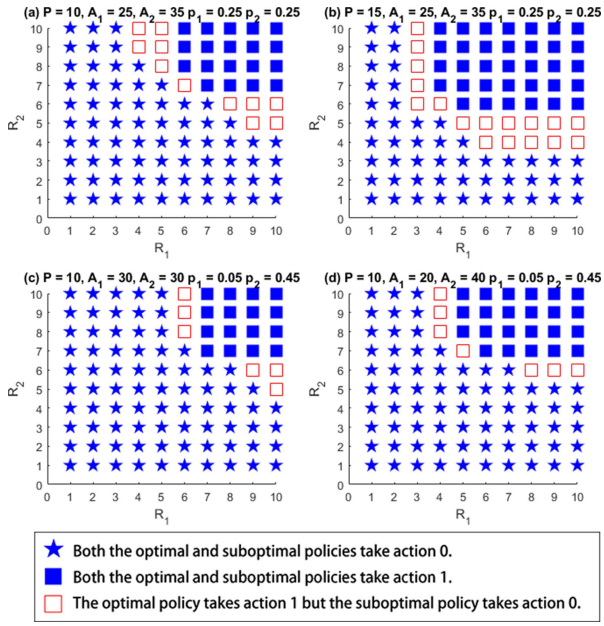


Fig. 4. Actions taken by the suboptimal policy and optimal policy under different states and system settings. We fix $\xi = 10$, $\hat{A} = 40$, and $K = 2$. (a) P = 10, $A_1 = 25$, $A_2 = 35$, $p_1 = 0.25$, $p_2 = 0.25$. (b) P = 15, $A_1 = 25$, $A_2 = 35$, $p_1 = 0.25$, $p_2 = 0.25$. (c) P = 10, $A_1 = 30$, $A_2 = 30$, $p_1 = 0.05$, $p_2 = 0.45$. (d) P = 10, $A_1 = 20$, $A_2 = 40$, $p_1 = 0.05$, $p_2 = 0.45$.

system dynamics, such as the impact of packet erasure rates on network performance, as we will demonstrate in Section VII-B.

VII. Performance Evaluation

In this section, we shall compare our proposed optimal and low-complexity suboptimal solution with other reference baselines. First, we investigate the difference between the optimal and the proposed low-complexity suboptimal policies under four states and system settings. Afterward, we simulate the performance of AoI under different policies concerning packet erasure rates and the number of devices.

A. Performance Evaluation Methodologies

The simulations are conducted with MATLAB. For Section VII-B, we perform Algorithms 1 and 2 to obtain the actions under different states. Since the Fig. 4(a)–(d) have different packet erasure rates, the algorithms are performed twice.

In Section VII-C, we compare the AoI performance with respect to different combinations of packet erasure rates and different numbers of users. All of the simulations in the section have the system parameters $\hat{A} = 40$, $\xi = 10$, and $T = 50000$. Other parameter settings are listed in Table II. The results are averaged over 100 runs. Except for the optimal and low-complexity suboptimal solutions, we further compare their performance with three available baseline policies.

The optimal fixed redundancy (Optimal FR) strategy does not require device feedback. The system calculates how many packets it should transmit based on the static packet erasure

TABLE II
PARAMETER SETTINGS FOR SECTION VII-C

Figure	5	6	7	8	9	10
Number of devices, K	2		2, 20	[2,100][a]		
Packet erasure rate of device k, p_k	$p_1 = 0.2$ $p_2 = [0.1, 0.6]$	$p_1 = 0.5$ $p_2 = [0.1, 0.6]$	[0.1,0.6], $\forall k \in \mathcal{K}$	0.25, $\forall k \in \mathcal{K}$	0.05, if $1 \leq k \leq \frac{K}{2}$ 0.45, if $\frac{K}{2} < k \leq K$	$\frac{1}{2}\frac{k-1}{K-1}$

[a] $[a, b]$ means the value ranges from a to b.

rates. If the number of currently transmitted packets is larger or equal to the threshold, it will start to send a new update in the next time slot. In the study introducing the FR strategy [14], the researchers detailed the average AoI for a specific device experiencing a packet erasure rate of δ when employing the FR strategy. This is represented as follows:

$$V^{\pi_{\text{FR}}}(n, \delta) = \frac{n}{F_\xi(n)} - \frac{n}{2} + \frac{\xi F_{\xi+1}(n+1)}{(1-\delta)F_\xi(n)} \tag{34}$$

where n is the FR strategy threshold, ξ is the rateless threshold, $F_\xi(n)$ is the cumulative distribution function (CDF) of a negative binomial (NB) $(\xi, 1-\delta)$ random variable. The optimal FR strategy will find the optimal FR strategy threshold, denoted by $P^\star$, to minimize the average AoI for all devices. Since the possible threshold values range from ξ to $\hat{A}$, and the status update for all users is independent and does not influence one another, the optimal FR strategy threshold $P^\star$ can be searched using a numerical method as described in Algorithm 3.

On the other hand, the IIR strategy will not transmit a new update until all devices successfully receive more or equal to ξ packets. This strategy can also be regarded as the persistent policy in [43]. The optimal fixed incremental redundancy (optimal FIR) strategy combines the optimal FR strategy and the IIR strategy. In summary, we can write the three strategies precisely as follows:

$$\pi_{\text{FR}}(X) = \begin{cases} 1, & P \geq P^\star \\ 0, & P < P^\star \end{cases}. \tag{35}$$

$$\pi_{\text{IIR}}(X) = \begin{cases} 1, & \mathbf{R} = \mathbf{0} \\ 0, & \text{otherwise} \end{cases}. \tag{36}$$

$$\pi_{\text{FIR}}(X) = \begin{cases} 1, & \mathbf{R} = \mathbf{0} \text{ or } P \geq P^\star \\ 0, & \text{otherwise} \end{cases}. \tag{37}$$

B. Observations From the Actions Obtained by the Optimal and Suboptimal Policies

Fig. 4 illustrates the actions of the optimal policy and suboptimal policy under different states and system settings. The red and hollow symbols represent the states that the suboptimal solution will select a different action from the optimal one. First, we observe that the suboptimal policy has the same action as the optimal one in most states. This proves the validity of the low-complexity suboptimal solution. It is worth noting that all the different actions stem from the suboptimal policy continuing to transmit the last status update. In contrast, the optimal policy decides to generate a new update. This implies the suboptimal policy tends to overestimate the Q-factor under action 0.

Next, we focus on the structural properties–all four subfigures in Fig. 4 indicate the structural property on $\mathbf{R}$ for both the optimal and suboptimal policies. If a strategy selects action 1 for a state, all actions to the top right of that state will also be 1. For the structural property on P, we shall focus on Fig. 4(a) and (b). If the strategy has chosen to generate a new update on the state in Fig. 4(a), i.e., a smaller P, it will keep the same decision at the same position in Fig. 4(b). It means that if one state with a lower P chooses to generate a new update, the other state will also follow suit, verifying the structural properties in Theorem 2 and Corollary 1.

Finally, we have some observations from the figure. Without the preemption, the source will wait for all devices to decode a status update so that it will not select to take action 1 unless $\mathbf{R} = \mathbf{0}$. However, from the number of square symbols in the figure, preemption is essential for minimizing time-average AoI. In addition, we can infer some simple regularity from the actions. From Fig. 4(a) and (b), when p_k are the same, but A_k are different, the source will deprioritize the device with lower A_k. For instance, the optimal policy will take action 1 when $\mathbf{R} = [4, 10]$ in Fig. 4(a), but will take action 0 when $\mathbf{R} = [10, 4]$. The same phenomenon can be seen in Fig. 4(c), where the states have the same A_k, whereas the packet erasure rates p_k are distinct. The source will sacrifice the performance of a device with a lower packet erasure rate. In Fig. 4(d), device 1 has a lower age and packet erasure rate, making the phenomenon more significant. Consequently, the source shall deprioritize the device with lower AoI and better channel conditions to minimize the overall time-average AoI.

We conclude the main properties of the optimal and suboptimal policies: First, the suboptimal policy closely resembles the optimal policy in most states, validating the effectiveness of the low-complexity suboptimal solution. Second, the structural properties of both policies are evident in all parts of Fig. 4. Finally, the optimal policy may prioritize certain devices over others with lower AoI and better channel conditions to optimize the overall time-average AoI.

C. Simulations for Time-Average AoI

The first three figures fix the number of devices in the system while the packet erasure rate changes. Figs. 5 and 6 illustrate the variation of AoI when we fix the packet erasure rate of one device, accompanied by the increase of the packet erasure rate of another device. The performance of our proposed suboptimal policy is almost the same as that of the optimal policy in these

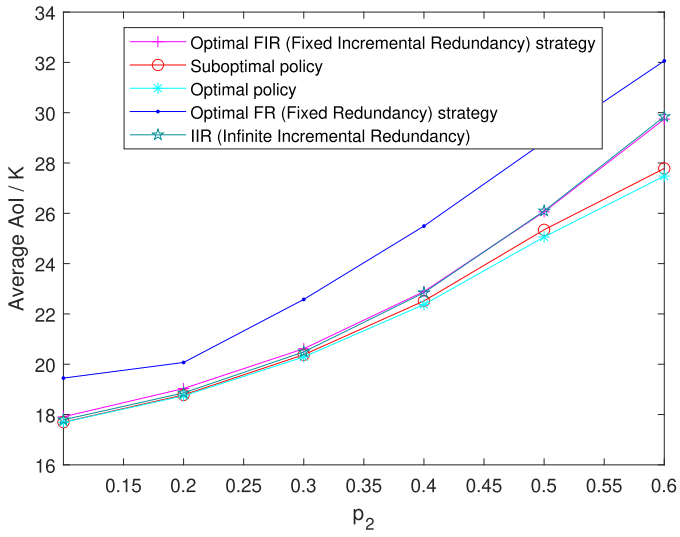


Fig. 5. Average AoI per-device with respect to p_2 when $K = 2$ and $p_1 = 0.2$.

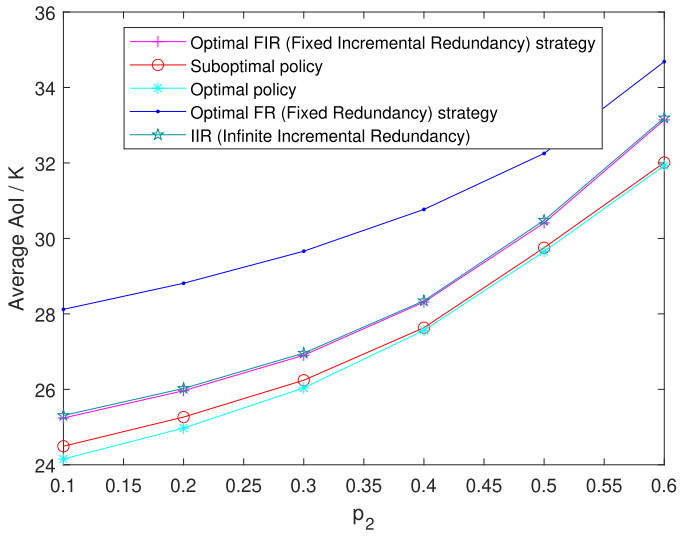


Fig. 6. Average AoI per-device with respect to p_2 when $K = 2$ and $p_1 = 0.5$.

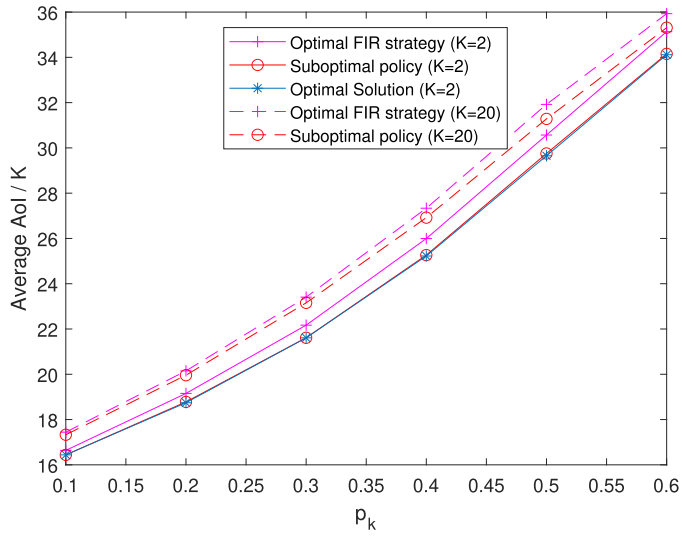


Fig. 7. Average AoI per-device with respect to different packet erasure rates $p = p_k$ for all $k \in \mathcal{K}$. The dashed lines show age with $K = 20$ and the solid lines show age with $K = 2$.

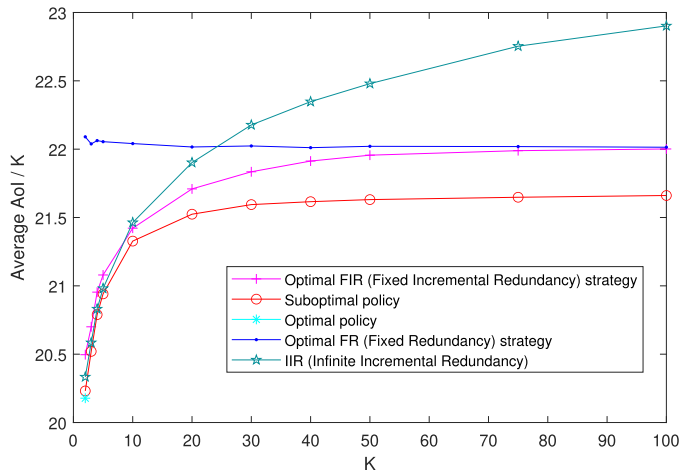


Fig. 8. Average AoI per-device with respect to the number of devices K with $p_k = 0.25$ for all $k \in \mathcal{K}$.

figures. Moreover, these two policies have a gap with other baseline policies, especially when the packet erasure rate increases. Among the baseline policies, the performance of the optimal FIR and IIR strategies is close, outperforming the optimal FR strategy.

Fig. 7 assumes all devices have the same packet erasure rates and focus on two and twenty device cases, respectively. When there are two devices, the suboptimal policy exhibits almost the same AoI performance as the optimal policy, and both are better than the optimal FIR strategy. The suboptimal policy still outperforms the optimal FIR strategy when the number of devices reaches twenty. We conclude that our proposed low-complexity suboptimal policy almost overlaps the curve of the optimal policy if the packet erasure rates are close, and they both surpass the baseline strategies if we vary the packet erasure rates.

Figs. 8–10 investigate the average AoI when the number of devices changes with the same pattern of packet erasure rates. The mean of packet erasure rates is set to be 0.25. Due to the curse of dimensionality, the optimal solution is no longer available when the number of devices increases, so we only simulate it under the two-device case.

Fig. 8 specifies all devices' packet erasure rates to be 0.25. Our proposed low-complexity suboptimal policy still outperforms other baseline policies. With the exception of the optimal FR strategy, AoI tends to increase as the number of devices increases. In addition, except for the IIR strategy, the AoI growth rate of other strategies decreases significantly when the number of devices is large. Fig. 9 divides devices into two groups, one with a high packet erasure rate of 0.45, whereas the other group only has a low erasure rate of 0.05. A similar phenomenon can be observed in this graph compared with that in Fig. 8, while the

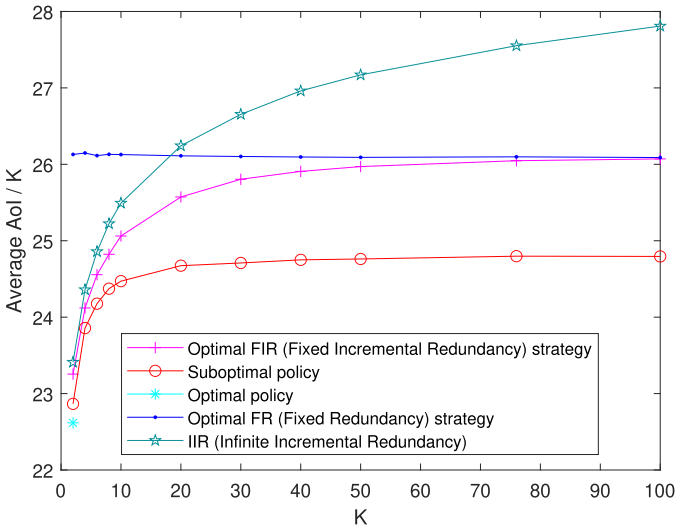


Fig. 9. Average AoI per-device with respect to the number of devices K with $p_k = 0.05$ if $1 \le k \le K/2$ and $p_k = 0.45$ if $K/2 < k \le K$. (All simulated K in the above table are even numbers.)

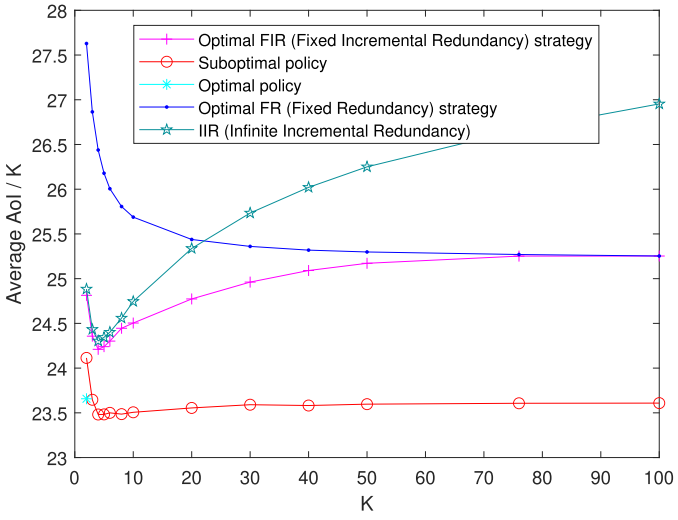


Fig. 10. Average AoI per-device with respect to the number of devices K with $p_k = 1/2k - 1/K - 1$ for all $k \in \mathcal{K}$.

gap between our proposed solution and the best baseline policy becomes greater. Furthermore, time-average AoI increases in all policies, implying the average AoI will increase if the packet erasure rates are not uniform. This shows that despite the same number of devices and packet erasure rates' mean, the time-average AoI can still be heavily influenced by the distribution of packet erasure rates.

Fig. 10 has the packet erasure rates linearly increase from 0 to 0.5 for each device, with the difference of $0.5/K - 1$. Again, the performance of the suboptimal policy surpasses other baseline strategies. As the number of devices increases, the average AoI of the suboptimal strategy remains stable, and its gap with the optimal FIR strategy is the largest among the three cases. Interestingly, except for the optimal FR strategy, when the number of devices is small, the average AoI drops to a low

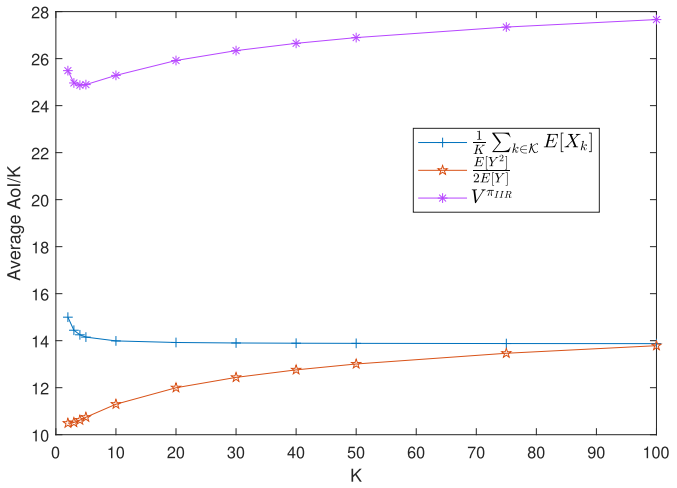


Fig. 11. Theoretical average AoI per-device under IIR strategy with respect to the number of devices K with $p_k = 1/2k - 1/K - 1$ for all $k \in \mathcal{K}$.

point and slowly rises. Since it is hard to find the closed-form average AoI value of the suboptimal solution, we will explain it by the IIR strategy. It shows a similar trend to the suboptimal policy and has a closed-form solution. Based on [14, Eq. (11)], we can derive the theoretical average AoI performance of IIR strategies $V^{\pi_{\text{IIR}}}$ without considering the upper limit of AoI. It can be written as follows:

$$V^{\pi_{\text{IIR}}} = \frac{1}{K}\sum_{k\in\mathcal{K}} \mathbb{E}[X_k] + \frac{\mathbb{E}[Y^2]}{2\mathbb{E}[Y]} \tag{38}$$

where $X_1, X_2, \ldots$ are independent NB $(\xi, 1 - p_k)$ random variables, representing the number of transmissions required for each device to successfully decode an update. $Y = \max(X_1, X_2, \ldots, X_K)$ is the maximum of all X_k. The first term $1/K \sum_{k\in\mathcal{K}} \mathbb{E}[X_k]$ is the average expected number of packets for each device to receive ξ packets. The second term $\mathbb{E}[Y^2]/2\mathbb{E}[Y]$ relates to Y. We expect Y to be positively correlated with K, as we have to wait longer if more devices have to receive an update. In Fig. 11, we perform a simulation to explain the phenomenon further. The first term declines as the number of devices expands. By contrast, the second term increases gradually with the number of devices. We can observe that the first term shrinks significantly when the number of devices is small. This causes the total age to fall initially and goes up subsequently due to the increase in the second term.[5]

Figs. 8–10 also reveal an interesting phenomenon: When the number of devices is low, the IIR strategy exhibits superior

[5]Since we have $\mathbb{E}[X_k] = \xi/1 - p_k$, the first term is equivalent to the multiplication of ξ and the reciprocal of the harmonic mean of all $1 - p_k$. Moreover, because $X_1, X_2, \ldots$ are independent, the CDF of Y can be denoted as $F_Y(y) = \prod_{k\in\mathcal{K}} F_k(y)$, where $F_k(x) = \sum_{i=\xi}^{x} \binom{i-1}{\xi-1}(1-p_k)^{\xi} p_k^{i-\xi}$ is the CDF of X_k. By utilizing $F_Y(y)$, the moments $\mathbb{E}[Y^2]$ and $\mathbb{E}[Y]$ can be deduced. Therefore, the simulation of Fig. 11 is deterministic. Besides, $V^{\pi_{\text{IIR}}}$ will be slightly higher than the simulated result in Fig. 10. The reason is that $\hat{A}$ is infinite in Fig. 11, whereas we truncate the excessive age to a finite $\hat{A}$ in Fig. 10.

performance in accurately capturing the retransmission timing. However, as the number of devices increases (with a threshold of 20 devices identified in our simulation), the IIR strategy, which waits for all devices to complete an update, shows significant shortcomings. In such scenarios, employing the optimal FR strategy, which transmits a fixed number of packets without considering the reception status, efficiently addresses this issue. The optimal FIR policy, integrating the advantages of both approaches, generally outperforms the former two under most conditions. Moreover, in scenarios where full feedback is available, the controller can further leverage this information to refine the transmission policy, achieving a superior average AoI compared with the initial three strategies, as demonstrated by our suboptimal policy.

In summary, the average AoI of our proposed suboptimal policy can approach the optimal one when the number of devices is low. The optimal solution is unavailable if the number of devices rises, but the suboptimal policy can still be used. In all settings, the suboptimal policy consistently outperforms other baseline policies.

VIII. Conclusion

In future networks demanding real-time information and reduced bandwidth, multicast networks with age-aware features are emerging as a promising solution. For instance, in smart factories, a server multicasts environmental data to robots, and in wildfire detection, sensors adjust placements in real-time based on multicast directives. We explored minimizing the time-average AoI in a multicast WSN using rateless codes, which eliminate retransmissions and save wireless resources. Using MDP, we identified unique state transition properties and confirmed the MDP's communicating property, making the Bellman equation's solution feasible for an age-optimal policy. We also introduced a structure-aware RVIA for efficient policy solutions. For large-scale networks, we proposed a decentralized, low-complexity approach for a suboptimal policy.

Through numerical analysis, we demonstrated the effectiveness of both optimal and suboptimal policies and their structural properties. These policies often align in most states. Following the proposed optimal scheduling policy, it is recommended that the source deprioritizes devices with lower AoI and better channel conditions to reduce the overall time-average AoI. In simulations, our two policies consistently surpass baseline ones, with the suboptimal nearing the optimal performance. Despite identical average packet erasure rates among devices, the distribution still impacts AoI performance. AoI does not necessarily rise with more devices. In essence, our proposed solutions address the issue efficiently.

Appendix A
Proof of Lemma 1

We define the terms *reach* and *randomized stationary policy*.

Definition 4: We say a state X can *reach* another state X' if there exists a stationary policy f such that for some t, we have the following:

$$Pr\left[X(t) = X' | X(0) = X, f\right] > 0. \tag{39}$$

Definition 5: A *randomized* and stationary policy defined as $\hat{\pi}$ is a mapping from a set of states to a distribution on the action space $\mathcal{D} = \{0, 1\}$. That is, $\hat{\pi} : \mathcal{X} \to \mathscr{P}(\mathcal{D})$, where $\mathscr{P}(\mathcal{D})$ is a collection of probability distributions on subsets of $\mathcal{D}$.

If for every two states $X, X' \in \mathcal{X}$, we have X can reach X', then the MDP is communicating. Note that from [46, Prop. 8.3.1], the stationary policy f is not required to be a deterministic policy. Therefore, we let f be a stationary randomized policy of the MDP that chooses $D(t) = 0$ and $D(t) = 1$ with equal probability for all t, implying we may take action 0 or action 1 for every state. Before we continue to our proof, we first derive two useful lemmas by the AoI model.

Lemma 3: Given two states $X, X' \in \mathcal{X}$ where $X = (P, (A_k, R_k)_{k\in\mathcal{K}})$ and $X' = (1, [A_k + 1]^+_{\hat{A}}, R'_k)_{k\in\mathcal{K}})$, X can reach X'.

Proof: The result is evident if we take action 1 on X, by (1) and (2). □

Lemma 4: Given two states $X, X' \in \mathcal{X}$ where $X = (1, (A_k, R_k)_{k\in\mathcal{K}})$ and $X' = (P', (A'_k, R'_k)_{k\in\mathcal{K}})$ with $P' > 1$. If $A'_k = [A_k + P' - 1]^+_{\hat{A}}$ for those k with $A'_k \neq P'$, and $R_k = \xi - 1 + \mathbb{1}(R'_k = \xi)\ \forall k \in \mathcal{K}$, then X can reach X'.

Proof: The following discussion focuses on the assumption the source takes action 0 starting from state X for $P' - 1$ times. Therefore, the limitation on A_k and R_k are both sufficient conditions of the lemma.

Under such assumption, the state $X' = (P', (A'_k, R'_k)_{k\in\mathcal{K}})$ will have two possible values of A'_k for each k. If a device k has successfully decoded the update, it will have $A'_k = P'$ by (5), and the initial value of A_k is no longer important. On the other hand, if the device k has not successfully decoded the update, its age will increase by $P' - 1$ and thus $A'_k = [A_k + P' - 1]^+_{\hat{A}}$. Therefore, for the first constraint on A'_k, we know if $A'_k \neq P'$, then $A'_k = [A_k + P' - 1]^+_{\hat{A}}$ is the age at device k after $P' - 1$ times of transmission.

If $R'_k = \xi$, it indicates the device k has never succeeded in receiving any packet, including the first transmission. Therefore, we should set $R_k = \xi$. Otherwise, the device has at least succeeded in receiving a packet once, and we can assume it occurs at the first transmission and let $R_k = \xi - 1$. Then, during the $P' - 1$ times of transmission, we can choose $\xi - 1 - R'_k$ times of transmission to be successful because of the fact that $P' - 1 \geq \xi - 1 - R'_k$ from Proposition 1 and $1 - p_k > 0$.

Consequently, such constraint of A_k and R_k ensures that X can reach X' by $P' - 1$ times of transmission with action 0 starting from state X.

Combining the Lemmas 3 and 4, we will have the following proposition.

Proposition 5: Given two states $X, X' \in \mathcal{X}$ where $X = (P, (A_k, R_k)_{k\in\mathcal{K}})$ and $X' = (P', (A'_k, R'_k)_{k\in\mathcal{K}})$. If $A'_k = [A_k + P']^+_{\hat{A}}$ for those k with $A'_k \neq P'$, X can reach X'.

Now, we come back to the original proof. Our target is to show for every state $X, X' \in \mathcal{X}$, X can reach X'. Notice we have the observation that R'_k will not influence the accessibility

from Proposition 5. Therefore, we shall only focus on P' and A'_k. For convenience, we denote P' as A'_0. Without loss of generality, we assume $A'_0 \le A'_1 \le A'_2 \le \cdots \le A'_K$. Besides, we define κ as the smallest index such that $A'_k = \hat{A}$, where

$$\kappa = \begin{cases} \min\{k | A'_k = \hat{A}, k \in \mathcal{K}\}, & \text{if } A'_K = \hat{A} \\ \infty, & \text{otherwise.} \end{cases} \tag{40}$$

Then, we define the difference between each term as e_k for all $1 \le k \le K$, where

$$e_k = \begin{cases} \max\{A'_k - A'_{k-1}, \xi\}, & \text{if } k = \kappa \\ A'_k - A'_{k-1}, & \text{otherwise.} \end{cases} \tag{41}$$

Hereafter, the state X' can be rewritten as $X' = (P', ([P' + \sum_{j=1}^{k} e_j]^+_{\hat{A}}, R'_k)_{k\in\mathcal{K}})$. From Proposition 1, we notice that $e_k = 0$ or $e_k \ge \xi$ if none of A'_k equals to $\hat{A}$. By specifying the value of e_κ, we can guarantee there is no $0 < e_k < \xi$. We define $\mathcal{I}_\xi$ as the set of index such that $e_k \ge \xi$, where $\mathcal{I}_\xi = \{i | e_i \ge \xi, i \in \mathcal{K}\}$, $\bar{i} = \max\ \mathcal{I}_\xi$, and $\underline{i} = \min\ \mathcal{I}_\xi$. The biggest index that is smaller than $i \in \mathcal{I}_\xi \setminus \{\underline{i}\}$ is defined as $i^\star = \max\{j | j < i, j \in \mathcal{I}_\xi\}$.

Furthermore, for all $i \in \mathcal{I}_\xi$, we define its corresponding state $X_i = (P_i, (A_{k,i}, R_{k,i})_{k\in\mathcal{K}}) = (e_i, (X_{k,i})_{k\in\mathcal{K}})$, where

$$A_{k,i} = \begin{cases} e_i, & \text{if } k < i \\ \sum_{j=i}^{k} e_j, & \text{if } k \ge i \\ \left[\sum_{j=i}^{k} e_j\right]^+_{\hat{A}}, & \text{if } i = \underline{i}, k = \kappa \end{cases}$$

$$R_{k,i} = \xi \cdot \mathbb{1}(A_{k,i} \ne e_i). \tag{42}$$

We design these states so that we can prove X_i can reach $X_{i^\star}$ iteratively. It is worth mentioning that only $A_{\kappa,\underline{i}}$ may take the minimum with $\hat{A}$, as we will show in (43). Because for all $k < \kappa$, $P' + \sum_{j=1}^{k} e_j < \hat{A}$. Besides, for $k = \kappa$, e_κ may compensate at most ξ for the difference. Note that if $\kappa \ne \infty$, $\bar{i} = \kappa$. By the facts that $e_{\underline{i}} \ge \xi$ and $P' > 0$, we have the following:

$$\begin{aligned} & P' + \sum_{j=1}^{\kappa} e_j - \hat{A} = P' + \sum_{j\in\mathcal{I}_\xi} e_j - \hat{A} \le \xi \\ \Leftrightarrow & \sum_{j\in\mathcal{I}_\xi\setminus\{\underline{i}\}} e_j - \hat{A} \le \xi - P' - e_{\underline{i}} < 0 \\ \Leftrightarrow & \sum_{j\in\mathcal{I}_\xi\setminus\{\underline{i}\}} e_j < \hat{A} \Leftrightarrow A_{\kappa,i} < \hat{A} \quad \forall i \in \mathcal{I}_\xi \setminus \{\underline{i}\}. \end{aligned} \tag{43}$$

Now, we will prove the three following things subsequently:

1) *X can reach $X_{\bar{i}}$:* Since $e_k = 0$ for all $k > \bar{i}$ by the definition of $\bar{i}$, we have $X_{\bar{i}} = (e_{\bar{i}}, (e_{\bar{i}}, 0)_{k\in\mathcal{K}})$, which implying $A_{k,\bar{i}} = P_{\bar{i}}$ for all k. Afterward, from Proposition 5, every state can reach $X_{\bar{i}}$.
2) *$X_{\bar{i}}$ can reach $X_{\underline{i}}$:* We will show that for all $i \in \mathcal{I}_\xi \setminus \{\underline{i}\}$, X_i can reach $X_{i^\star}$. From the definition of $X_{i^\star}$, we can find that for device $k < i$, $A_{k,i^\star} = P_{i^\star}$, so we do not have to concern the value of $A_{k,i}$. Then, if $k \ge i$, since $A_{k,i} = \sum_{j=i}^{k} e_j$ if $i \in \mathcal{I}_\xi \setminus \{\underline{i}\}$, $A_{k,i^\star} = [\sum_{j=i^\star}^{k} e_j]^+_{\hat{A}} = [\sum_{j=i}^{k} e_j + \sum_{j=i^\star}^{i-1} e_j]^+_{\hat{A}} = [A_{k,i} + e_{i^\star}]^+_{\hat{A}} = [A_{k,i} + P_{i^\star}]^+_{\hat{A}}$. Therefore, by the Proposition 5, X_i can reach $X_{i^\star}$. By iteratively applying this result, we have $X_{\bar{i}}$ can reach $X_{\underline{i}}$.
3) *$X_{\underline{i}}$ can reach X':* Note that $X' = (P', ([P' + \sum_{j=1}^{k} e_j]^+_{\hat{A}}, R'_k)_{k\in\mathcal{K}})$. For those $k \ge \underline{i}$, we have $A'_k > P'$. Then, $A'_k = [P' + \sum_{j=\underline{i}}^{k} e_j]^+_{\hat{A}} = [P' + [\sum_{j=\underline{i}}^{k} e_j]^+_{\hat{A}}]^+_{\hat{A}} = [P' + A_{k,\underline{i}}]^+_{\hat{A}}$. If $k < \underline{i}$, $A'_k = P'$. Therefore, by Proposition 5, $X_{\underline{i}}$ can reach X'.

If $|\mathcal{I}_\xi| = 1$, we shall skip the step 2 since $\bar{i} = \underline{i}$. If $\mathcal{I}_\xi = \phi$, it means $P' = A'_k \forall k \in \mathcal{K}$, so every state can reach X' by Proposition 5. In summary, we can guarantee X can always reach X', and the proof is completed.

Appendix B
Proof of Lemma 2

At the beginning, if $h_n(X^1) \le h_n(X^2)$ holds for all n, then by (21), we can take limit on both sides and thus $h(X^1) \le h(X^2)$. Therefore, we shall focus on the proof of $h_n(X^1) \le h_n(X^2)$ for all n in the remaining part.

We will use the mathematical induction to prove the statement. Because we have initialized all $h_0(X) = 0$, so it holds when $n = 0$. Assume it holds for $n = n_0$ and by the condition $\mathbf{A}^1 \preceq \mathbf{A}^2$, $h_{n_0+1}(X^1)$ has the following relationship:

$$\begin{aligned} & h_{n_0+1}(X^1) \\ & = \sum_{k\in\mathcal{K}} A^1_k + \min_{d\in\mathcal{D}} \sum_{X^{1'}\in\mathcal{X}} Pr[X^{1'}|X^1, d] h_{n_0}(X^{1'}) \\ & = \sum_{k\in\mathcal{K}} A^1_k + \sum_{X^{1'}\in\mathcal{X}} Pr[X^{1'}|X^1, \pi^\star(X^1)] h_{n_0}(X^{1'}) \\ & \le \sum_{k\in\mathcal{K}} A^1_k + \sum_{X^{1'}\in\mathcal{X}} Pr[X^{1'}|X^1, \pi^\star(X^2)] h_{n_0}(X^{1'}) \\ & \le \sum_{k\in\mathcal{K}} A^2_k + \sum_{X^{1'}\in\mathcal{X}} Pr[X^{1'}|X^1, \pi^\star(X^2)] h_{n_0}(X^{1'}). \end{aligned} \tag{44}$$

The first inequality holds because $\pi^\star(X^1)$ is the optimal action for X^1, but $\pi^\star(X^2)$ is not necessarily. Similarly, $h_{n_0+1}(X^2)$ can be expressed as follows:

$$\begin{aligned} & h_{n_0+1}(X^2) \\ & = \sum_{k\in\mathcal{K}} A^2_k + \sum_{X^{2'}\in\mathcal{X}} Pr\left[X^{2'}|X^2, \pi^\star(X^2)\right] h_{n_0}(X^{2'}). \end{aligned} \tag{45}$$

Therefore, to prove $h_{n_0+1}(X^1) \le h_{n_0+1}(X^2)$, it is sufficient to prove $\sum_{X^{1'}\in\mathcal{X}} Pr[X^{1'}|X^1, \pi^\star(X^2)] h_{n_0}(X^{1'}) \le \sum_{X^{2'}\in\mathcal{X}} Pr[X^{2'}|X^2, \pi^\star(X^2)] h_{n_0}(X^{2'})$. If $\pi^\star(X^2) = 0$, then from (4) and (5), we can verify that $P^{1'} \le P^{2'}$, $\mathbf{A}^{1'} \preceq \mathbf{A}^{2'}$, and $\mathbf{R}^{1'} \preceq \mathbf{R}^{2'}$ for all possible transitions.[6] On the other hand, if $\pi^\star(X^2) = 1$, we have $P^{1'} = P^{2'}$, $\mathbf{A}^{1'} \preceq \mathbf{A}^{2'}$, and $\mathbf{R}^{1'} = \mathbf{R}^{2'}$; as a result, regardless of $\pi^\star(X^2)$, we have $h_{n_0}(X^{1'}) \le h_{n_0}(X^{2'})$

[6] Since there are K devices and each device may either succeed or fail to receive a packet, we have 2^K possible transitions after a decision.

according to the hypothesis of the induction. Hence, we conclude that $\sum_{X^{1'}\in\mathcal{X}} Pr[X^{1'}|X^1,\pi^\star(X^2)]h_{n_0}(X^{1'}) \leq \sum_{X^{2'}\in\mathcal{X}} Pr[X^{2'}|X^2,\pi^\star(X^2)]h_{n_0}(X^{2'})$ holds, and it implies $h_{n_0+1}(X^1) \leq h_{n_0+1}(X^2)$. Consequently, by the mathematical induction, $h_n(X^1) \leq h_n(X^2)$ holds for all n, and the proof of Lemma 2 is completed.

Appendix C
Proof of Theorem 1

We will start from the part of taking action 1. According to (19), it suffices to prove that the term $\sum_{X^{1'}\in\mathcal{X}} Pr[X^{1'}|X^1,1]h_n(X^{1'})$ is equal to $\sum_{X^{2'}\in\mathcal{X}} Pr[X^{2'}|X^2,1]h_n(X^{2'})$ since $\mathbf{A}^1=\mathbf{A}^2$. According to the AoI model (2) and (3), the two states will have the same transition for each device k if their ages are the same. Therefore, $h_n(X^{1'})=h_n(X^{2'})$ holds for all possible transitions, and the proof is completed.

For the part of taking action 0, by (19), it is equivalent to prove the inequality $\sum_{X^{1'}\in\mathcal{X}} Pr[X^{1'}|X^1,0]h_n(X^{1'}) \leq \sum_{X^{2'}\in\mathcal{X}} Pr[X^{2'}|X^2,0]h_n(X^{2'})$ instead. Under the premise $P^1\leq P^2$, $\mathbf{A}^1=\mathbf{A}^2$, and $\mathbf{R}^1\preceq\mathbf{R}^{2'}$, one can check that from (4) and (5), $P^{1'}\leq P^{2'}$, $\mathbf{A}^{1'}\preceq\mathbf{A}^{2'}$, and $\mathbf{R}^{1'}\preceq\mathbf{R}^{2'}$ holds for all possible transitions if action 0 is taken. The age vector $\mathbf{A}^{1'}$ may become smaller because there may exist some k who successfully received the packet, causing its R_k^1 to become 0, but R_k^2 does not. Thus, $A_k^{1'}$ decreases to $P^{1'}$ by (5). Following Lemma 2, we deduce that $h_n(X^{1'})\leq h_n(X^{2'})$, and the proof is completed immediately.

Appendix D
Proof of Proposition 4

The proof will be divided into two parts. First, we will prove that if a state has $\mathbf{R}=\boldsymbol{\xi}$, then $\pi^\star(X)=1$. According to the definition of the optimal action for a state in (18), our goal is to identify the action that minimizes the expected cost after its execution. For a state X with $\mathbf{R}=\boldsymbol{\xi}$, we denote the new state after taking action 0 as $X^{0'}$ and 1 as $X^{1'}$. We have $P^{0'}=[P+1]^+_{\hat{A}} > P^{1'}=1$ by (4) and (1). Because $\mathbf{R}=\boldsymbol{\xi}$, none of the devices can successfully decode an update and reduce its age at the next time slot, implying $\mathbf{A}^{0'}=\mathbf{A}^{1'}$ and $\mathbf{R}^{0'}=\mathbf{R}^{1'}$. After applying Lemma 2, we have $h(X^{0'})\geq h(X^{1'})$. Combining (18), we immediately obtain $\pi^\star(X)=1$.

Let us move to the proof that if a state with $\mathbf{R}=\mathbf{0}$, then $\pi^\star(X)=1$. We will again use mathematical induction to prove the statement. To make the proof more concise, we rewrite the narrative of the proof as the following lemma.

Lemma 5: For any $X^1=(P^1,\mathbf{A}^1,\mathbf{R}^1), X^2=(P^2,\mathbf{A}^2,\mathbf{R}^2)\in\mathcal{X}$ with $\mathbf{A}^1=\mathbf{A}^2$ and $\mathbf{R}^1=\mathbf{0}$, we have $h_n(X^1)\geq h_n(X^2)$ and $\pi_n^\star(X^1)=1$ for all n. Hence, $h(X^1)\geq h(X^2)$ and $\pi^\star(X^1)=1$.

Proof: In the beginning, we define the optimal policy based on the relative value function at iteration n, $h_n(\cdot)$, as follows:

$$\begin{aligned}\pi_n^\star(X) &= \arg\min_{d\in\mathcal{D}} Q_n(X,d)\\ &= \arg\min_{d\in\mathcal{D}} \sum_{X'\in\mathcal{X}} Pr[X'|X,d]h_n(X'). \qquad (46)\end{aligned}$$

Because we have initialized all $h_0(X)=0$, both statements are true for $n=0$. Assume $h_{n0}(X^1)\geq h_{n0}(X^2)$ is true for $n=n_0$. We first prove $\pi_{n0}^\star(X^1)=1$ by discussing the transited states after taking different actions on X^1.

We denote the transited state after taking action d for state X^1 as $X_d^{1'}$. If $d=0$, since we have $\mathbf{R}^1=\mathbf{0}$, it will follow (5), and thus $X_0^{1'}=([P^1+1]^+_{\hat{A}},([A_k^1+1]^+_{\hat{A}},0)_{k\in\mathcal{K}})$. On the other hand, if $d=1$, the state $X_1^{1'}$ becomes $(1,([A_k^1+1]^+_{\hat{A}},R_k)_{k\in\mathcal{K}})$, where each R_k is either ξ or $\xi-1$ by (3). Note that $\mathbf{A}_0^{1'}=\mathbf{A}_1^{1'}$ and $\mathbf{R}_0^{1'}=\mathbf{0}$ under all possible transitions. Therefore, $h_{n0}(X_0^{1'})\geq h_{n0}(X_1^{1'})$ for all possible transition, and thus $\sum_{X_0^{1'}\in\mathcal{X}} Pr[X_0^{1'}|X^1,0]h_{n_0}(X_0^{1'}) \geq \sum_{X_1^{1'}\in\mathcal{X}} Pr[X_1^{1'}|X^1,1]h_{n_0}(X_1^{1'})$, implying $\pi_{n0}^\star(X^1)=1$ by the definition of (46).

Then, we will prove $h_{n_0+1}(X^1)\geq h_{n_0+1}(X^2)$ is true under the assumption $h_{n_0}(X^1)\geq h_{n_0}(X^2)$. By observing (19) and (20), since $\mathbf{A}^1=\mathbf{A}^2$ and $h_n(o)$ is prescribed, we only have to prove $\sum_{X^{2'}\in\mathcal{X}} Pr[X^{2'}|X^2,\pi_{n_0}^\star(X^2)]h_{n_0}(X^{2'}) \leq \sum_{X^{1'}\in\mathcal{X}} Pr[X^{1'}|X^1,\pi_{n_0}^\star(X^1)]h_{n_0}(X^{1'})$:

$$\begin{aligned}&\sum_{X^{2'}\in\mathcal{X}} Pr[X^{2'}|X^2,\pi_{n_0}^\star(X^2)]h_{n_0}(X^{2'})\\ \leq &\sum_{X^{2'}\in\mathcal{X}} Pr[X^{2'}|X^2,1]h_{n_0}(X^{2'})\\ = &\sum_{X^{1'}\in\mathcal{X}} Pr[X^{1'}|X^1,1]h_{n_0}(X^{1'})\\ = &\sum_{X^{1'}\in\mathcal{X}} Pr[X^{1'}|X^1,\pi_{n_0}^\star(X^1)]h_{n_0}(X^{1'}). \qquad (47)\end{aligned}$$

The inequality holds because $\pi^\star(X^2)$ is the optimal action for X^2. Thus, selecting action $\pi_{n_0}^\star(X^2)$ is always at least as good as selecting 1. Furthermore, since X^1 and X^2 both have $\mathbf{A}^1=\mathbf{A}^2$, the first equality is validated by Theorem 1. Finally, given that $\mathbf{R}^1=\mathbf{0}$, the second equality is confirmed by the proof from the previous part. As a result, we have proven that $h_{n_0+1}(X^1)\geq h_{n_0+1}(X^2)$ as indicated in (47). By mathematical induction the statement $h_n(X^1)\geq h_n(X^2)$ holds for every natural number n. It follows $\pi_n^\star(X^1)=1$ holds for every natural number n, and thus $\pi^\star(X^1)=1$.

In Lemma 5, if $X^2=X^1$, $\pi^\star(X^1)=1$ still holds. Accordingly, we conclude $\pi^\star(X)=1$ if $\mathbf{R}=\boldsymbol{\xi}$ or $\mathbf{R}=\mathbf{0}$, by the first paragraph and Lemma 5, respectively.

References

[1] N. Chukhno, O. Chukhno, S. Pizzi, A. Molinaro, A. Iera, and G. Araniti, "Approaching 6G use case requirements with multicasting," *IEEE Commun. Mag.*, vol. 61, no. 5, pp. 144–150, May 2023.

[2] S. Kaul, M. Gruteser, V. Rai, and J. Kenney, "Minimizing age of information in vehicular networks," in *Proc. 8th Annu. IEEE Commun. Soc. Conf. Sensor, Mesh Ad Hoc Commun. Netw.*, 2011, pp. 350–358.

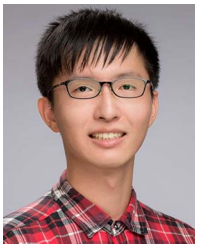
[3] S. Kaul, R. Yates, and M. Gruteser, "On piggybacking in vehicular networks," in *Proc. IEEE Glob. Telecommun. Conf.*, 2011, pp. 1–5.

[4] S. Kaul, R. Yates, and M. Gruteser, "Real-time status: How often should one update?," in *Proc. IEEE INFOCOM*, 2012, pp. 2731–2735.
[5] 3GPP, "Evolved universal terrestrial radio access (E-UTRA); radio resource control (RRC); protocol specification," *3rd Gener. Partnership Project (3GPP), Sophia Antipolis, France, Tech. Specification (TS) 26.346, version 17.0.0*, Aug. 2022.
[6] 3GPP, "Architectural enhancements for 5G multicast-broadcast services," *3rd Gener. Partnership Project (3GPP), Sophia Antipolis, France, Tech. Specification (TS) 23.247, version 17.6.0*, Apr. 2023.
[7] 3GPP, "Study on mission critical services over 5G multicast broadcast system," *3rd Gener. Partnership Project (3GPP), Sophia Antipolis, France, Tech. Rep. (TR) 23.774, version 17.0.0*, Mar. 2020.
[8] 3GPP, "5G multicast-broadcast services; user service architecture," *3rd Gener. Partnership Project (3GPP), Sophia Antipolis, France, Tech. Specification (TS) 26.502, version 17.4.0*, Apr. 2023.
[9] A. Rajanna and M. Haenggi, "Enhanced cellular coverage and throughput using rateless codes," *IEEE Trans. Commun.*, vol. 65, no. 5, pp. 1899–1912, May 2017.
[10] M. Luby, "LT codes," in *Proc. 43rd Annu. IEEE Symp. Found. Comput. Sci.*, 2002, pp. 271–280.
[11] A. Shokrollahi, "Raptor codes," *IEEE Trans. Inf. Theory*, vol. 52, no. 6, pp. 2551–2567, Jun. 2006.
[12] L. Minder, A. Shokrollahi, M. Watson, M. Luby, and T. Stockhammer, "RaptorQ forward error correction scheme for object delivery," RFC 6330, Aug. 2011. [Online]. Available: https://www.rfc-editor.org/info/rfc6330
[13] A. Shokrollahi and M. Luby, "Raptor codes," *Found. Trends Commun. Inf. Theory*, vol. 6, no. 3–4, pp. 213–322, 2011. [Online]. Available: http://dx.doi.org/10.1561/0100000060
[14] R. D. Yates, E. Najm, E. Soljanin, and J. Zhong, "Timely updates over an erasure channel," in *Proc. IEEE Int. Symp. Inf. Theory*, 2017, pp. 316–320.
[15] R. D. Yates, Y. Sun, D. R. Brown, S. K. Kaul, E. Modiano, and S. Ulukus, "Age of information: An introduction and survey," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 5, pp. 1183–1210, May 2021.
[16] J. Östman, R. Devassy, G. Durisi, and E. Uysal, "Peak-age violation guarantees for the transmission of short packets over fading channels," in *Proc. IEEE INFOCOM IEEE Conf. Comput. Commun. Workshops*, 2019, pp. 109–114.
[17] V. Tripathi, R. Talak, and E. Modiano, "Information freshness in multi-hop wireless networks," *IEEE/ACM Trans. Netw.*, vol. 31, no. 2, pp. 784–799, Apr. 2023.
[18] R. D. Yates and S. K. Kaul, "Status updates over unreliable multiaccess channels," in *Proc. IEEE Int. Symp. Inf. Theory*, 2017, pp. 331–335.
[19] M. Zhang, A. Arafa, J. Huang, and H. V. Poor, "Pricing fresh data," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 5, pp. 1211–1225, May 2021.
[20] S. Zhang, J. Li, H. Luo, J. Gao, L. Zhao, and X. S. Shen, "Low-latency and fresh content provision in information-centric vehicular networks," *IEEE Trans. Mobile Comput.*, vol. 21, no. 5, pp. 1723–1738, May 2022.
[21] I. Kadota, A. Sinha, E. U.-Biyikoglu, R. Singh, and E. Modiano, "Scheduling policies for minimizing age of information in broadcast wireless networks," *IEEE/ACM Trans. Netw.*, vol. 26, no. 6, pp. 2637–2650, Dec. 2018.
[22] Y.-P. Hsu, E. Modiano, and L. Duan, "Scheduling algorithms for minimizing age of information in wireless broadcast networks with random arrivals," *IEEE Trans. Mobile Comput.*, vol. 19, no. 12, pp. 2903–2915, Dec. 2020.
[23] I. Kadota and E. Modiano, "Minimizing the age of information in wireless networks with stochastic arrivals," *IEEE Trans. Mobile Comput.*, vol. 20, no. 3, pp. 1173–1185, Mar. 2021.
[24] E. T. Ceran, D. Gündüz, and A. György, "A reinforcement learning approach to age of information in multi-user networks with HARQ," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 5, pp. 1412–1426, May 2021.
[25] T. Zhang et al., "AoI and PAoI in the IoT-based multisource status update system: Violation probabilities and optimal arrival rate allocation," *IEEE Internet Things J.*, vol. 10, no. 23, pp. 20 617–20 632, Dec. 2023.
[26] Z. Chen et al., "Analysis of age of information in dual updating systems," *IEEE Trans. Wireless Commun.*, vol. 22, no. 11, pp. 8003–8019, Nov. 2023.
[27] K. Lang, Z. Chen, N. Pappas, H. H. Yang, Y. Jia, and T. Q. S. Quek, "The effect of device redundancy in timeliness of information," in *Proc. IEEE Int. Conf. Commun.*, 2023, pp. 1694–1699.
[28] Z. Tang, N. Yang, P. Sadeghi, and X. Zhou, "Age of information in downlink systems: Broadcast or unicast transmission?," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 7, pp. 2057–2070, Jul. 2023.
[29] J. Zhong, E. Soljanin, and R. D. Yates, "Status updates through multicast networks," in *Proc. IEEE 55th Annu. Allerton Conf. Commun., Control, Comput.*, 2017, pp. 463–469.
[30] J. Zhong, R. D. Yates, and E. Soljanin, "Multicast with prioritized delivery: How fresh is your data?," in *Proc. IEEE 19th Int. Workshop Signal Process. Adv. Wirel. Commun.*, 2018, pp. 1–5.
[31] B. Buyukates, A. Soysal, and S. Ulukus, "Age of information in multihop multicast networks," *J. Commun. Netw.*, vol. 21, no. 3, pp. 256–267, 2019.
[32] Y. Chen, H. Tang, and J. Wang, "Optimizing age of information in multicast unilateral networks," in *Proc. IEEE Int. Symp. Broadband Multimedia Syst. Broadcast.*, 2020, pp. 1–4.
[33] J. Li, Y. Zhou, and H. Chen, "Age of information for multicast transmission with fixed and random deadlines in IoT systems," *IEEE Internet Things J.*, vol. 7, no. 9, pp. 8178–8191, Sep. 2020.
[34] Y.-P. Hsu, "Efficient broadcast for timely updates in mobile networks," *IEEE Commun. Lett.*, vol. 25, no. 6, pp. 1969–1973, Jun. 2021.
[35] S. Nath, J. Wu, and J. Yang, "Optimum energy efficiency and age-of-information tradeoff in multicast scheduling," in *Proc. IEEE Int. Conf. Commun.*, 2018, pp. 1–6.
[36] M. Xie, J. Gong, and X. Ma, "Age-energy tradeoff of short packet based transmissions in multicast networks with ARQ," in *Proc. IEEE 91st Veh. Technol. Conf.*, 2020, pp. 1–5.
[37] H. Wei and N. Deng, "On the age of information in wireless networks using rateless codes," *IEEE Access*, vol. 8, pp. 173147–173157, 2020.
[38] S. Meng, S. Wu, A. Li, J. Jiao, N. Zhang, and Q. Zhang, "Analysis and optimization of spinal codes over the BSC: From the AoI perspective," in *Proc. IEEE Int. Conf. Commun. Workshops*, 2021, pp. 1–6.
[39] B. Wang, S. Feng, and J. Yang, "When to preempt? age of information minimization under link capacity constraint," *J. Commun. Netw.*, vol. 21, no. 3, pp. 220–232, 2019.
[40] 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.
[41] B. Zhou and W. Saad, "Minimum age of information in the Internet of Things with non-uniform status packet sizes," *IEEE Trans. Wireless Commun.*, vol. 19, no. 3, pp. 1933–1947, Mar. 2020.
[42] X. Xie, H. Wang, L. Yu, and M. Weng, "Online algorithms for optimizing age of information in the IoT systems with multi-slot status delivery," *IEEE Wireless Commun. Lett.*, vol. 10, no. 5, pp. 971–975, May 2021.
[43] S. Feng and J. Yang, "Age-optimal transmission of rateless codes in an erasure channel," in *Proc. IEEE Int. Conf. Commun.*, 2019, pp. 1–6.
[44] E. Gindullina, L. Badia, and D. Gündüz, "Age-of-information with information source diversity in an energy harvesting system," *IEEE Trans. Green Commun. Netw.*, vol. 5, no. 3, pp. 1529–1540, Sep. 2021.
[45] S. Leng and A. Yener, "Age of information minimization for an energy harvesting cognitive radio," *IEEE Trans. Cogn. Commun. Netw.*, vol. 5, no. 2, pp. 427–439, Jun. 2019.
[46] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*, 1st ed. New York, NY, USA: Wiley, 1994.
[47] D. P. Bertsekas, *Dynamic Programming and Optimal Control*, 3rd ed. Belmont, MA, USA: Athena Scientific, 2007.
[48] W. B. Powell, *Approximate Dynamic Programming: Solving the Curses of Dimensionality*. Hoboken, NJ, USA: Wiley, 2007.
[49] A. M. Bedewy, Y. Sun, S. Kompella, and N. B. Shroff, "Optimal sampling and scheduling for timely status updates in multi-source networks," *IEEE Trans. Inf. Theory*, vol. 67, no. 6, pp. 4019–4034, Jun. 2021.

Hung-Chun Lin received the B.S. degree in electrical engineering in 2021 from National Taiwan University, Taipei, Taiwan, where he is currently working toward the M.S. degree in communication engineering.

His research interests include wireless networks, information freshness, reinforcement learning, and power-saving mechanisms.

Mr. Lin was the recipient of Academic Excellence Award from National Taiwan University, in 2020.



Kuang-Hsun Lin (Member, IEEE) received the B.S. degree in electrical engineering from National Taiwan University, Taipei, Taiwan, in 2015, and the Ph.D. degree in communication engineering from GICE, National Taiwan University, in 2022.

Since 2015, he has been working under the supervision of Prof. Hung-Yu Wei with the Wireless and Mobile Networking Laboratory. He completed summer internships with MediaTek in 2015 and 2018. Currently, he is a Postdoctoral Researcher with GICE, National Taiwan University, and also a Postdoctoral Research Assistant for Taiwan's NSTC 6G Program Office. His research interests include wireless mobile networks, power-saving mechanisms for devices and networks, mobility management, and MAC protocol design.

Hung-Yu Wei (Senior Member, IEEE) received the B.S. degree in electrical engineering from National Taiwan University, Taipei, Taiwan, in 1999. He received the M.S. and the Ph.D. degrees in electrical engineering from Columbia University, New York, NY, USA, in 2001 and 2005, respectively.

He was a Interim Department Chair and Associate Chair with Department of Electrical Engineering. In 2000 and 2001, he was a summer intern with Telcordia Applied Research. From 2003 to 2005, he was with NEC Labs America. In 2005, he joined Department of Electrical Engineering with the National Taiwan University, where he is currently a Professor with Department of Electrical Engineering and Graduate Institute of Communications Engineering. His research interests include next-generation wireless networks, IoT, fog/edge computing, cross-layer design for wireless multimedia, and game theoretic models for communications networks.

Dr. Wei was the recipient of NTU Excellent Teaching Award in 2008 and 2018, the "Recruiting Outstanding Young Scholar Award" from the Foundation for the Advancement of Outstanding Scholarship in 2006, K. T. Li Young Researcher Award from ACM Taipei/Taiwan Chapter and The Institute of Information and Computing Machinery in 2012, Excellent Young Engineer Award from the Chinese Institute of Electrical Engineering in 2014, Wu Ta You Memorial Award from MOST in 2015, and Outstanding Research Award from MOST in 2020. He has been actively participating in NGMN, IEEE 802.16, 3GPP, IEEE P1934, and IEEE P1935 standardization. He is the Secretary for IEEE ComSoC Fog/Edge Industry Community. He is Chair of IEEE P1935 working group for edge/fog management and orchestration standard and Chair of IEEE P1934 Working Group for edge/fog computing and networking architecture. He is an Associate Editor for *IEEE IoT Magazine*, and was an Associate Editor for IEEE INTERNET OF THINGS JOURNAL, and IEEE SYSTEMS JOURNAL. He is an IEEE certified Wireless Communications Professional. He was the Chair of IEEE VTS Taipei Chapter during 2016–2017.