A Reinforcement Learning Approach to Age of Information in Multi-User Networks with HARQ

Elif Tuğçe Ceran, Deniz Gündüz, and András György

***Abstract*—Scheduling the transmission of time-sensitive information from a source node to multiple users over error-prone communication channels is studied with the goal of minimizing the long-term average *age of information (AoI)* at the users. A long-term average resource constraint is imposed on the source, which limits the average number of transmissions. The source can transmit only to a single user at each time slot, and after each transmission, it receives an instantaneous ACK/NACK feedback from the intended receiver, and decides when and to which user to transmit the next update. Assuming the channel statistics are known, the optimal scheduling policy is studied for both the standard automatic repeat request (ARQ) and hybrid ARQ (HARQ) protocols. Then, a *reinforcement learning* (RL) approach is introduced to find a near-optimal policy, which does not assume any *a priori* information on the random processes governing the channel states. Different RL methods including average-cost SARSA with linear function approximation (LFA), upper confidence reinforcement learning (UCRL2), and deep Q-network (DQN) are applied and compared through numerical simulations.**

Index Terms: Age of information, hybrid automatic repeat request (HARQ), constrained Markov decision process, reinforcement learning, Whittle index.

I. Introduction

We consider a status update system, in which a source node wants to communicate the state of a time-varying process to multiple users. The timeliness of the information at each user is measured by the *age of information* (AoI), defined as the time elapsed since the most recent status update received by that user was generated at the source [2]–[4]. The goal of the source is to minimize the *average* AoI across the users. Most of the earlier work on AoI consider queue-based models, in which the status updates arrive at the source node randomly according to a Poisson process, and are stored in a buffer before being transmitted to the destination [3]–[6]. Instead, we consider the *generate-at-will* model, in which the source can generate a fresh status update at any time [2], [7]–[14].

We address the scheduling of status updates in a multi-user network under a transmission-rate constraint. This constraint is motivated by the limited energy supplies of most sensors (e.g., powered via energy harvesting [7], [14], [15]); hence, they cannot send an unlimited number of updates. We assume that the source can transmit to only a single user at each time slot, and the communication channels experience fading. While the source does not have channel state information, we assume the presence of a single bit perfect feedback link from each user to the source terminal, across which the corresponding receiver can send ACK/NACK feedback after each transmission. We consider both the standard ARQ and the hybrid ARQ (HARQ) protocols. Note that, in the former, the same transmission is repeated until it is successfully received; however, in a status update system no retransmission takes place, as it is always better to send a fresh status update. On the other hand, under HARQ, one may repeat previously sent packets as the probability of correct decoding increases with multiple transmissions. First, we assume that the success probability of each transmission attempt is known beforehand, in which case the source can judiciously decide when to transmit, or, in the case of HARQ, to retransmit or discard failed information and send a fresh update. Then, we consider scheduling status updates over unknown channels, in which case the success probabilities of transmission attempts are not known *a priori*, and must be learned in an online fashion using the ACK/NACK signals.

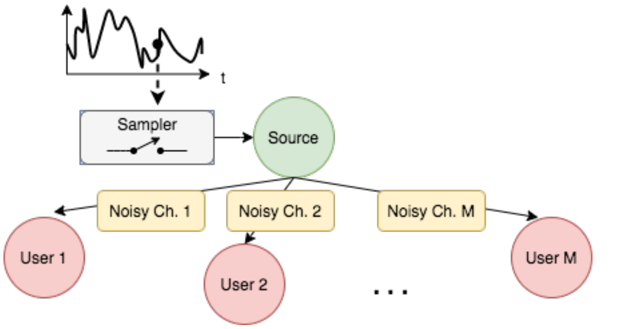

Figure 1. The system model of a multi-user status update system over error prone links.

AoI in multi-user networks has been studied in [9], [10], [13], [16]–[19]. It is shown in [16] that the scheduling problem, where a set of links that share a common channel and the transmitter at each link contains a given number of packets with time stamps from an information source, is NP-hard. Scheduling transmissions to multiple receivers is investigated in [9], focusing on a perfect transmission medium, and the optimal scheduling algorithm is shown to be of threshold-type on the AoI. Average AoI has also been studied when status updates are transmitted over unreliable multiple-access channels [17] or multi-cast networks [18]. A source node send-

Part of this work is was presented at the IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, Bologna, Italy, September 2018 [1]. This work was supported in part by the European Research Council (ERC) Starting Grant BEACON (grant agreement no. 677854) and by UK EPSRC (grant no. EP/T023600/1). E. T. Ceran was with Imperial College London, UK. She is currently with Middle East Technical University, Turkey, email: elifce@metu.edu.tr. D. Gündüz is with Imperial College London, UK (email: d.gunduz@imperial.ac.uk. A. György is with DeepMind, UK (email: agyorgy@google.com).

ing time-sensitive information to a number of users through unreliable channels is considered in [13], where the problem is formulated as a *restless multi-armed bandit* (RMAB), and a suboptimal Whittle Index (WI) policy is proposed.

In [5], [10], [20], AoI at a single user is studied when status updates are transmitted over an erasure channel with retransmissions. Two HARQ protocols are considered to combat erasures: infinite incremental redundancy (IIR) and fixed redundancy (FR) coding. The IIR protocol represents a system in which a status update packet is encoded with k_s symbols ratelessly, such that the transmission of an update continues until k_s symbols are received. An FR protocol represents an (n_s, k_s)-maximum distance separable (MDS) code, where each update is transmitted as an n_s symbol packet, and the packet can be decoded if at least k_s symbols are received. An information theoretic approach to the AoI problem is taken in [21], where the optimal average AoI is characterized when no feedback is assumed. In this paper, we instead consider scheduling of status updates to multiple users under a transmission rate constraint for general HARQ protocols, and we study standard ARQ and FR HARQ protocols as a special case of HARQ. In our earlier work, we studied a point-to-point status update system under a transmission-rate constraint [11], [12], and showed that the optimal policy is a randomized stationary policy with randomization at most one state. As opposed to the single user setting, in the multi-user scenario considered in this paper, the source has to decide not only when to transmit, but also to which user to transmit, significantly increasing both the state and action spaces of the underlying problem.

Most prior literature on AoI assume perfect statistical knowledge of the random processes governing the status update system. However, in most practical systems (e.g., sensors embedded in unknown or time-varying environments), the characteristics of the system are not known *a priori*, and must be learned. A limited number of recent works consider the unknown or time-varying characteristics of status update systems, and apply a learning-theoretic approach [1], [6], [9], [11], [12], [14], [22]–[24]. The scheduling decisions with multiple receivers over a perfect channel is investigated in [6], [9], where the goal is to learn data arrival statistics. Q-learning is used for a generate-at-will model in [9], while policy gradients and DQN methods are used for a queue-based multi-flow AoI-optimal scheduling problem in [6]. In [22], policy gradients and DQN methods are employed for AoI minimization in a wireless ad-hoc network, where nodes exchange status updates with one another over a shared spectrum. Average cost reinforcement learning (RL) algorithms are proposed in [1], [12] to learn the decoding error probabilities in a status update system with HARQ. The work in [14] exploits RL methods in order to learn both the decoding error probabilities and the energy harvesting characteristics.

To the best of our knowledge, the average AoI with HARQ is studied for the first time for a multi-user system under a long-term average resource constraint. Similarly, there is no prior work in the literature which compares the performances of the various RL methods exploited in this paper. The main contributions of this paper can be summarized as follows:

- Both retransmission and pre-emption following a failed transmission are considered, corresponding, respectively, to the HARQ and ARQ protocols, and the structure of the optimal policy is determined.
- The multi-user scheduling problem is shown to be indexable, and suboptimal WI policies are derived in closed-form for the standard ARQ and FR HARQ protocols.
- Lower bounds on the average AoI are proposed for the standard ARQ and the FR HARQ protocols under a resource constraint.
- We employ average-cost RL algorithms, in particular, *average-cost SARSA*, *upper confidence reinforcement learning* (UCRL2), *average-cost SARSA with softmax and linear function approximation* (LFA) and *deep reinforcement learning* (DRL) to learn the optimal scheduling decisions when the transmission probabilities are unknown.
- Extensive numerical simulations are conducted in order to analyze the effect of the resource constraint, the network size, and the ARQ or HARQ mechanisms on the freshness of information, and the effectiveness of the proposed RL algorithms.

II. System Model and Problem Formulation

We consider a slotted status update system, where a source terminal monitors a time-varying process and sends updates about the process' state to multiple users. In every time slot, the source terminal is able to generate an update at the beginning of the slot, and can transmit a status update to (at most) one of the M users. This can be either because of dedicated orthogonal links to the users, for example, in a wired network, or because the users are interested in distinct information. A transmission attempt of a status update to a single user takes constant time, which is assumed to be equal to the duration of one time slot.

We assume that the state of each of the channels changes randomly from one time slot to the next in an independent and identically distributed (i.i.d.) fashion, and the channel state information is available only at the corresponding receivers. We assume the availability of an instantaneous error-free single-bit ACK/NACK feedback from each user to the source. Successful reception of the status update at the end of time slot t is acknowledged by an ACK signal (denoted by $K_t = 1$), while a NACK signal is sent in case of a failure (denoted by $K_t = 0$). In the standard ARQ protocol, a packet is retransmitted after each NACK feedback, until it is successfully decoded. However, in the AoI framework there is no point in retransmitting a failed out-of-date status packet if it has the same error probability as a fresh status update. Hence, the source always removes a failed status signal, and transmits a fresh update. On the other hand, in HARQ, signals from previous transmission attempts are combined, and therefore the probability of error decreases with every retransmission [25].

In practice, the utility of status updates typically becomes zero beyond a certain age, hence we assume that the age is bounded; as such, we assume that the maximum age is $N < \infty$. Assuming that the most up-to-date packet received by the j^{th} user ($j \in [M] \triangleq \{1, \ldots, M\}$) before time slot t was generated in slot $U_j(t)$, the AoI at the receiver of user j at the beginning of time slot t is defined as $\delta_{j,t}^{rx} \triangleq \min\{t - U_j(t), N\} \in [N] \triangleq \{1, \dots, N\}$.

At each time slot t, the source node takes an action a_t from the set $\mathcal{A} = \{\mathrm{i}, \mathrm{n}_1, \mathrm{x}_1, \dots, \mathrm{n}_M, \mathrm{x}_M\}$: in particular, the source can i) remain idle ($a_t = \mathrm{i}$); ii) generate and transmit a new status update to the j^{th} user ($a_t = \mathrm{n}_j$, $j \in [M]$); or, iii) retransmit the most recent failed status update to the j^{th} user ($a_t = \mathrm{x}_j$, $j \in [M]$). Note that $|\mathcal{A}| = 2M + 1$. For the j^{th} user, the probability of error after r retransmissions, denoted by $g_j(r)$, depends on r and the particular HARQ scheme used [25]. In any reasonable HARQ strategy, $g_j(r)$ is non-increasing in r, i.e., $1 > g_j(r) \geq g_j(r') > 0$ for all $r \leq r'$. We will denote the maximum number of retransmissions by r_{max}. We note that standard HARQ methods only allow a finite number of retransmissions (e.g., $r_{max} = 3$ [26], [27]).

Let $\delta_{j,t}^{tx}$ denote the number of time slots elapsed since the generation of the most recently transmitted (successfully or not) packet to user j at the transmitter, while recall that $\delta_{j,t}^{rx}$ denote the AoI of the most recently received status update at the receiver of user j. $\delta_{j,t}^{tx}$ resets to 1 if a new status update is generated for user j at time slot $t - 1$, and increases by one (up to N) otherwise, i.e.,

$$\delta_{j,t+1}^{tx} = \begin{cases} 1 & \text{if } a_t = \mathrm{n}_j; \\ \min(\delta_{j,t}^{tx} + 1, N) & \text{otherwise.} \end{cases}$$

On the other hand, the AoI at the receiver side evolves as follows:

$$\delta_{j,t+1}^{rx} = \begin{cases} 1 & \text{if } a_t = \mathrm{n}_j \text{ and } K_t = 1; \\ \min(\delta_{j,t}^{tx} + 1, N) & \text{if } a_t = \mathrm{x}_j \text{ and } K_t = 1; \\ \min(\delta_{j,t}^{rx} + 1, N) & \text{otherwise.} \end{cases}$$

Note that once the AoI at the receiver is at least as large as at the transmitter, this relationship holds forever; thus it is enough to consider cases when $\delta_t^{rx} \geq \delta_t^{tx}$.

Therefore, $\delta_{j,t}^{rx}$ increases by 1 when the source chooses to transmit to another user, or if the transmission fails, while it decreases to 1, or, in the case of HARQ, to $\min(\delta_{j,t}^{tx} + 1, N)$, when a status update is successfully decoded. Also, $\delta_{j,t}^{tx}$ increases by 1 if the source chooses not to generate a new packet and transmit it to user j ($a_t \neq \mathrm{n}_j$).

For the j^{th} user, let $r_{j,t} \in \{0, \dots, r_{max}\}$ denote the number of previous transmission attempts of the most recent packet. Thus, the number of retransmissions is zero for a newly sensed status update and increases up to r_{max} as we keep retransmitting the same packet. Then, the state of the system can be described by the vector $s_t \triangleq (\delta_{1,t}^{rx}, \delta_{1,t}^{tx}, r_{1,t}, \dots, \delta_{M,t}^{rx}, \delta_{M,t}^{tx}, r_{M,t})$, where s_t belongs to the set of possible states $\mathcal{S} \subset ([N] \times [N] \times [r_{max}])^M$.

If no resource constraint is imposed, remaining idle is clearly a suboptimal action. However, in practice, continuous transmission is typically not possible due to energy or interference constraints. To model these situations, we impose a constraint on the average number of transmissions, denoted by $\lambda \in (0, 1]$. This leads to a constrained Markov decision proccess (CMDP) formulation, defined by the 5-tuple $\left(\mathcal{S}, \mathcal{A}, \mathcal{P}, c, d\right)$: The countable set of states $\mathcal{S}$ and the finite set of actions $\mathcal{A}$ have already been defined. $\mathcal{P}$ refers to the transition kernel and can be summarized as follows: $\mathcal{P}_{s,s'}(a) =$

$$\begin{cases} 1 & \text{if } a = \mathrm{i}, \delta_i^{rx'} = \min\{\delta_i^{rx} + 1, N\}, \\ & \ \delta_i^{tx'} = \min\{\delta_i^{tx} + 1, N\},\ r_i' = r_i,\ \forall i; \\ 1 - g_j(0) & \text{if } a = \mathrm{n}_j, \delta_j^{rx'} = 1, \delta_j^{tx'} = 1, \delta_i^{rx'} = \min\{\delta_i^{rx} + 1, N\}, \\ & \delta_i^{tx'} = \min\{\delta_i^{tx} + 1, N\}, r_j' = 0, r_i' = r_i, \forall i \neq j; \\ g_j(0) & \text{if } a = \mathrm{n}_j,\ \delta_j^{rx'} = \min\{\delta_j^{rx} + 1, N\},\ \delta_j^{tx'} = 1, \\ & r_j' = 1,\ r_i' = r_i,\ \delta_i^{rx'} = \min\{\delta_i^{rx} + 1, N\}; \\ & \ \delta_i^{tx'} = \min\{\delta_i^{tx} + 1, N\}, \forall i \neq j; \\ 1 - g_j(r_j) & \text{if } a = \mathrm{x}_j, \delta_j^{rx'} = \delta_j^{tx} + 1,\ r_j' = 0,\ r_i' = r_i, \\ & \delta_j^{tx'} = \min\{\delta_j^{tx} + 1, N\}, \delta_i^{rx'} = \min\{\delta_i^{rx} + 1, N\}, \\ & \delta_i^{tx'} = \min\{\delta_i^{tx} + 1, N\}, \forall i \neq j; \\ g_j(r_j) & \text{if } a = \mathrm{x}_j, \delta_j^{rx'} = \min\{\delta_j^{rx} + 1, N\}, r_i' = r_i, \\ & r_j' = \min\{r_j' + 1, r_{max}\}, \delta_i^{tx'} = \min\{\delta_i^{tx} + 1, N\}, \\ & \delta_j^{tx'} = \min\{\delta_j^{tx} + 1, N\}, \delta_i^{rx'} = \min\{\delta_i^{rx} + 1, N\}, \forall i \neq j; \\ 0 & \text{otherwise,} \end{cases} \tag{1}$$

where $\mathcal{P}_{s,s'}(a) = \Pr(s_{t+1} = s' \mid s_t = s, a_t = a)$ is the probability that action $a \in \mathcal{A}$ in state $s \in \mathcal{S}$ at time t leads to state $s' \in \mathcal{S}$ at time $t + 1$ (the components of state s' are denoted by a prime in the above equation). The instantaneous cost function $c : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is defined as the weighted sum of the AoIs at the multiple users, independently of a. Formally, $c(s, a) = \Delta \triangleq w_1 \delta_1^{rx} + \dots + w_M \delta_M^{rx}$, where the weight $w_j > 0$ represents priority of user j. The instantaneous transmission cost $d : \mathcal{A} \to \mathbb{R}$ is defined as $d(\mathrm{i}) = 0$ and $d(a) = 1$ if $a \neq \mathrm{i}$.

Naturally, as reflected by the system model, for every user we keep only the most recent status update packet: thus, the number of retransmissions is zero for a newly sensed and generated status update and increases up to r_{max} as we keep retransmitting the same packet. If a maximum of r_{max} retransmissions is reached, the packet can still be retransmitted; however, due to the protocol, only the last r_{max} retransmissions are used in the decoding, hence the retransmission count saturates at r_{max}. Figure 2 illustrates an example showing the actions and state transitions for a 2-user system.

A stationary *policy* $\pi : \mathcal{S} \times \mathcal{A} \to [0, 1]$ maps each state $s \in \mathcal{S}$ to an action $a \in \mathcal{A}$ with probability $\pi(a|s)$ ($\pi(\cdot|s)$ is a distribution over $\mathcal{A}$). We use $s_t^\pi = (\delta_{1,t}^{rx\,\pi}, \delta_{1,t}^{tx\,\pi}, r_{1,t}^\pi, \dots, \delta_{M,t}^{rx\ \pi}, \delta_{M,t}^{tx\ \pi}, r_{M,t}^\pi)$ and a_t^π to denote the sequences of states and actions, respectively, induced by policy π, while $\Delta_t^\pi \triangleq \sum_{j=1}^M w_j \delta_{j,t}^{rx\,\pi}$ denotes the instantaneous weighted cost.

The infinite horizon expected weighted average AoI for policy π starting from the initial state $s_0 \in \mathcal{S}$ is defined as

$$J^\pi(s_0) \triangleq \limsup_{T \to \infty} \frac{1}{T} \mathbb{E}\left[\sum_{t=1}^{T} \Delta_t^\pi \middle| s_0\right], \tag{2}$$

while the corresponding average number of transmissions is

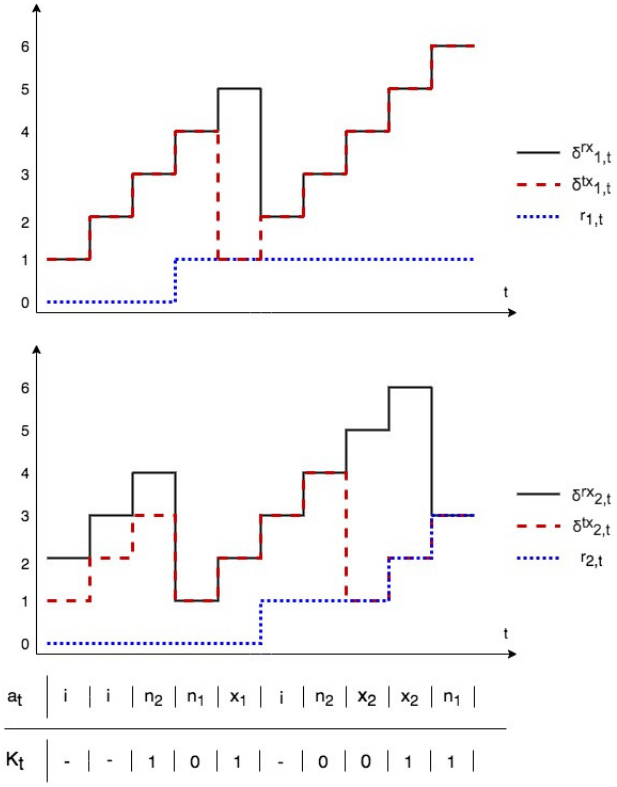

Figure 2. An example illustrating the AoIs and retransmission numbers for a 2-user network in the presence of ACK/NACK feedback

given by

$$C^{\pi}(s_0) \triangleq \limsup_{T\to\infty} \frac{1}{T}\mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}[a_t^{\pi} \neq \text{i}] \middle| s_0\right] . \tag{3}$$

We are interested in minimizing $J^{\pi}(s_0)$ given a constraint λ on the average number of transmissions $C^{\pi}(s_0)$, leading to the following CMDP optimization problem:

Problem 1. $\underset{\pi\in\Pi}{\text{Minimize}}$ $J^{\pi}(s_0)$ over $\pi \in \Pi$ such that $C^{\pi}(s_0) \leq \lambda$.

Without loss of generality, we assume that the state at the beginning of the problem is $s_0 = (1, 1, 0, 2, 1, 0, \ldots, M, 1, 0)$; and we omit s_0 from the notation for simplicity. A policy $\pi^* \in \Pi$ is called optimal if $J^* \triangleq J^{\pi^*} \leq J^{\pi}$ for all $\pi \in \Pi$ and we are interested in finding optimal policies.

III. Lagrangian Relaxation and the Structure of the Optimal Policy

A detailed treatment of finite-state finite-action discounted MDPs is considered in [28], but here we need more general results that apply to MDPs and CMDPs with average expected cost [28], [29]. Below we follow [29] and [30] to characterize the optimal policy.

We will need two well-known concepts for MDPs [28], [29]: An MDP is *communicating* if for any two states s, s' there exists a deterministic policy π such that s' is reachable from s' with positive probability following π. A stronger concept is the *unichain* property, which we define for the more general class of CMDPs: a finite CMDP is unichain if any feasible policy (i.e., a policy that satisfies the resource constraint) induces a finite-state Markov chain that contains a single recurrent class and possibly, some transient states. We will show below that our MDP is communicating (cf. Theorem 1) and that it is unichain under the ARQ protocol (cf. Theorem 2).

To solve the constrained MDP, we rewrite Problem 1 in its Lagrangian form. Average Lagrangian cost of policy π with Lagrange multiplier $\eta \geq 0$ is defined as

$$L_{\eta}^{\pi} = \lim_{T\to\infty} \frac{1}{T}\left(\mathbb{E}\left[\sum_{t=1}^{T}\Delta_t^{\pi}\right] + \eta\mathbb{E}\left[\sum_{t=1}^{T}\mathbb{1}[a_t^{\pi} \neq \text{i}]\right]\right) \tag{4}$$

and, for any η, the optimal achievable cost is defined as $L_{\eta}^{*} \triangleq \inf_{\pi} L_{\eta}^{\pi}$. This formulation is equivalent to an unconstrained countable-state average-cost MDP with instantaneous (overall) cost $\Delta_t^{\pi} + \eta\mathbb{1}[a_t^{\pi} \neq \text{i}]$.

If $\lambda = 1$, a transmission (new update or retransmission) is allowed in every time slot, and instead of a CMDP we have a finite-state MDP with bounded cost. Then it follows from Theorem 8.4.3 and Theorem 8.4.5 of [28] that if the MDP is unichain (which holds for the ARQ protocol as shown in Theorem 2), there exists an optimal deterministic policy that satisfies the Bellman equations. In this section, we focus on the more interesting constrained problem. The constraint on the transmission cost is less than or equal to one (i.e., $\lambda \leq 1$), then we have $\eta \geq 0$, which will be assumed throughout the paper. A policy π is called η-optimal if it achieves L_{η}^{*}.

Theorem 1. *An optimal stationary policy π_n^* minimizing* (4) *(and hence achieving L_{η}^{*}) exists for the unconstrained MDP with Lagrangian parameter η.*

Proof. First, we show that the unconstrained MDP is communicating, that is, for every pair of $(s, s') \in \mathcal{S}$, there exists a deterministic policy under which s' is accessible from s. It is easy to see that there exists a policy which induces a recurrent Markov chain: Consider the policy which always transmits to the user with the smallest index such that the corresponding AoI at the user is less than N or the retransmission count is less than r_{max}, sending a new packet if the retransmission count is 0 and retransmitting if it is not. This policy gets to the state $(N, N, r_{max}, \ldots, N, N, r_{max})$ from any other state with at least a fixed positive probability in at most $M \max\{N, r_{max}\}$ steps, hence it induces a recurrent Markov chain. It follows than from Proposition 8.3.1 of [28] that the MDP is communicating. Then, by Theorem 8.3.2 of [28], an optimal stationary policy satisfying (5) exists. □

On the other hand, if the MDP is unichain, we can obtain stronger results specifying the structure of an optimal policy. In this case, there exists a function $h_{\eta}(s)$, called the *differential cost function*, satisfying the *Bellman optimality* equations

$$h_{\eta}(s) + L_{\eta}^{*} = \min_{a\in\mathcal{A}} \left(\Delta + \eta \cdot \mathbb{1}[a \neq \text{i}] + \mathbb{E}\left[h_{\eta}(s')|s, a\right]\right), \tag{5}$$

$\forall s \in \mathcal{S}$, where $s' \in \mathcal{S}$ is the next state obtained after taking action a [28]. The *state-action cost function* is defined as

$$Q_{\eta}(s, a) \triangleq \Delta + \eta \cdot \mathbb{1}[a \neq \text{i}] + \mathbb{E}\left[h_{\eta}(s')|s, a\right], \tag{6}$$

$\forall s \in \mathcal{S}, a \in \mathcal{A}$. Then, at each state the optimal deterministic policy takes the action achieving the minimum in (6):

$$\pi_\eta^*(s) \in \underset{a\in\{\mathrm{i,n,x}\}}{\arg\min}\, Q_\eta(s,a) \ . \tag{7}$$

For a single-user point-to-point status update system, [12] characterizes the structure of the optimal policy, and shows that there exists a stationary policy which randomizes in at most one state. Next we extend this result to multi-user status update systems for the ARQ protocol.

If we assume that the system adopts the standard ARQ protocol, that is, failed transmissions are discarded at the destination, then the state space reduces to $(\delta_1^{rx}, \delta_2^{rx}, \dots, \delta_M^{rx})$ as $r_{j,t} = 0,\ \forall j, t$, and the action space to $\mathcal{A} = \{\mathrm{i}, \mathrm{n}_1, \dots, \mathrm{n}_M\}$. The probability of error of each status update is $p_j \triangleq g_j(0)$ for user j. State transitions in (1) and the Bellman optimality equations can all be modified accordingly. Then we can extend Theorem 1 of [12] to multi-user systems.

Theorem 2. *There exists an optimal stationary policy for Problem 1 under standard ARQ, which is optimal for the unconstrained problem considered in* (4) *for some* $\eta = \eta^*$*, and randomizes in at most one state. This policy can be expressed as a mixture of two deterministic policies* $\pi^*_{\eta^*,1}$ *and* $\pi^*_{\eta^*,2}$ *that differ in at most a single state* $\hat{s}$*, and are both optimal for the Lagrangian problem* (4) *with* $\eta = \eta^*$*. More precisely, there exist two deterministic policies* $\pi^*_{\eta^*,1}$*,* $\pi^*_{\eta^*,2}$ *as described above and* $\mu \in [0,1]$*, such that the mixture policy* $\pi^*_{\eta^*}$*, which selects, in state* $\hat{s}$*,* $\pi^*_{\eta^*,1}(\hat{s})$ *with probability* μ *and* $\pi^*_{\eta^*,2}(\hat{s})$ *with probability* $1-\mu$*, and otherwise follows these two policies (which agree in all other states) is optimal for Problem 1, and the constraint in* (3) *is satisfied with equality.*

Proof. Since the state $(N, N, \dots, N)$ is visited under every stationary policy with at least a fixed positive probability in at most NM steps from every other state under the ARQ protocol (if NM transmissions fail), the CMDP is unichain. Then, by Theorem 4.4 of [29], since Problem 1 is feasible (i.e., there exists at least one policy which satisfies the constraint (3)), there exists an optimal stationary policy that is a mixture of two deterministic policies that differ in at most a single state with $\mu \in [0,1]$. From Section 4.4, Theorem 3.6 and Theorem 4.4 of [29], the mixture policy $\pi^*_{\eta^*}$, for any $\mu \in [0,1]$, also satisfies (5), and is optimal for the unconstrained problem in (4) with $\eta = \eta^*$. This completes the proof of the theorem. □

Some other results in [29], [30] will be useful in determining $\pi^*_{\eta^*}$. For any $\eta > 0$, let C_η and J_η denote the average number of transmissions and average AoI, respectively, for the optimal policy π^*_η. Note that, C_η and J_η can be computed directly by finding the stationary distribution of the chain, or estimated empirically by running the MDP with policy π^*_η.

To determine the optimal policy, one needs to find η^*, and the policies $\pi^*_{\eta^*,1}$ and $\pi^*_{\eta^*,2}$, In fact, [30] shows that η^* is defined as

$$\eta^* \triangleq \inf\{\eta > 0 : C_\eta \leq \lambda\}, \tag{8}$$

where the inequality $C_\eta \leq \lambda$ is satisfied if it is satisfied for at least one of $C^{\pi^*_{\eta^*,i}}$ for $i = 1$ or $i = 2$. By Lemma 3.3 of [30], η^* is finite, and $\eta^* > 0$ if $\lambda < 1$.

Algorithm 1 Relative Value Iteration (RVI)

```
Input: Lagrange parameter $\eta$, error probability $g(r)$, $(\delta^{ref}, r^{ref})$
    /* choose an arbitrary but fixed reference state */
$h_0^{N\times r_{max}} \leftarrow \mathbf{0}$        /* initialization */
for episodes $n = 0, 1, 2, \dots$ do
    for state $s \in \mathcal{S}$ do
        for action $a \in \mathcal{A}$ do
            $Q_{n+1}(s,a) \leftarrow \Delta + \eta \cdot \mathbb{1}[a^\pi \neq \mathrm{i}] + \mathbb{E}\left[h_n(s')\right]$
        end for
        $V_{n+1}(\delta, r) \leftarrow \min_a(Q_{n+1}(\delta, r, a))$
        $h_{n+1}(\delta, r) \leftarrow V_{n+1}(\delta, r) - V_{n+1}(\delta^{ref}, r^{ref})$
    end for
    if $|h_{n+1} - h_n| \leq \epsilon$ then
    /* convergence */
        for $s \in \mathcal{S}$ do    /* compute the optimal policy */
            $\pi_\eta^*(s) \leftarrow \arg\min_a(Q(s,a))$
        end for
        Return $\pi^*$
    end if
end for
```

Theorem 2 and the discussion above describe the general structure of the optimal policy. A detailed discussion on finding both η^* and the policies $\pi^*_{\eta^*,1}$ and $\pi^*_{\eta^*,2}$ are given in [12], which is not the focus of this paper. In Section IV, for practical implementation, an iterative heuristic algorithm, also is employed based on the discussion in this section.

IV. An Iterative Algorithm to Minimize AoI

For a practical solution to our problem, we can employ the *relative value iteration* (RVI) [28] algorithm to solve (5) for any given η; and hence, find (an approximation of) the optimal policy π^*_η. To simplify the notation, the dependence on η is suppressed in the algorithm for h and Q. Note that a finite approximation is needed for the practical implementation of the RVI algorithm since each iteration of RVI requires the computation of the value function for each state-action pair.

The RVI algorithm essentially computes the optimal state value function through improving the estimates of state and state action values, $h(s)$ and $Q(s,a)$, respectively. The algorithm starts with a random initialization of $h_0(s), \forall s$, and sets an arbitrary but fixed reference state s^{ref}. It then continuously updates the $Q(s,a)$ and $h(s)$ values until convergence. A single iteration of the RVI algorithm is given as follows:

$$Q_{n+1}(s,a) \leftarrow \Delta + \eta \cdot \mathbb{1}[a^\pi \neq \mathrm{i}] + \mathbb{E}\left[h_n(s')|a\right], \tag{9}$$

$$h_{n+1}(s) \leftarrow \min_a(Q_{n+1}(s,a)) - \min_a(Q_{n+1}(s^{ref}, a)), \tag{10}$$

where $Q_n(s,a)$ and $h_n(s)$ denote the state action value function and differential value function for iteration n, respectively.

After presenting an algorithm that can compute the optimal deterministic policy π^*_η for any given η (more precisely, an arbitrarily close approximation thereof for the finite-state MDP), we need to find the particular Lagrange multiplier η^* as defined by (8). A heuristic method to find a single η value with $C_\eta \approx \lambda$ is as follows: We start with an initial parameter η^0, and run an iterative algorithm updating η as $\eta^{m+1} = \eta^m + \alpha_m(C_{\eta^m} - \lambda)$ for a step size parameter α_m[1].

[1] α_m is a positive decreasing sequence and satisfies the following conditions: $\sum_m \alpha_m = \infty$ and $\sum_m \alpha_m^2 < \infty$ from the theory of stochastic approximation [31].

We continue this iteration until $|C_{\eta^m} - \lambda|$ becomes smaller than a given threshold, and denote the resulting value by η^*. A more detailed discussion on an iterative algorithm minimizing AoI is also given in [11], [12].

V. AoI with Standard ARQ Protocol

In this section, we assume that the system adopts the standard ARQ protocol and the state space reduces to $(\delta_1^{rx}, \delta_2^{rx}, \dots, \delta_M^{rx})$ as $r_{j,t} = 0$, $\forall j, t$, and the action space to $\mathcal{A} = \{\mathrm{i}, \mathrm{n}_1, \dots, \mathrm{n}_M\}$. The probability of error of each status update is $p_j \triangleq g_j(0)$ for user j. State transitions in (1), the Bellman optimality equations, and the RVI algorithm can all be simplified accordingly. Thanks to these simplifications, we are able to derive a low-complexity policy based on Whittle's approach [32] by modelling the problem as a RMAB [33]. Although the RVI algorithm in Section IV provides an optimal solution to Problem 1, its computational complexity grows quickly with the network size. The WI policy in Section V-A provides a suboptimal yet computationally efficient alternative, which performs very well in practice. We also derive a lower bound for the constrained MDP in Section V-B.

A. WI Policy

Multi-armed bandits (MAB) [33] constitute a class of RL problems with a single state. In the restless MAB (RMAB) problem [32], each arm is associated with a state that evolves over time, and the reward distribution of the arm depends on its state (in contrast, in the basic stochastic MAB problems, rewards are i.i.d.). The multi-user AoI minimization problem with ARQ can be formulated as a RMAB with $M+1$ arms: choosing arm j is associated with transmitting to user j, while arm $M+1$ represents the action of staying idle ($a = \mathrm{i}$). RMAB problems are known to be PSPACE-hard in general [33]; however, a low-complexity heuristic policy can be found for certain problems by relaxing the constraint that in every round only a single arm can be selected, and instead introducing a bound on the expected number of arms chosen [32]. The resulting policy, known as the WI policy, is a sub-optimal policy, but it is known to perform close to optimal in many settings [32].

Following Whittle's approach, we decouple our problem into M sub-problems each corresponding to a single user, and treat these problems independently. The cost of transmitting to a user (called *subsidy for passivity* [32]) is denoted by C, which will be later used to derive the index policy. Writing the Bellman equation (6) for each subproblem, we obtain the optimality equations for the single user AoI minimization problem with the standard ARQ protocol where the action space is $\{\mathrm{i}, \mathrm{n}_j\}$

$$h_C(\delta_j^{rx}) + L_j^* = \min\left\{Q(\delta_j^{rx}, \mathrm{n}_j), Q(\delta_j^{rx}, \mathrm{i})\right\}, \tag{11}$$

and the optimal policy to each subproblem is given

$$\pi_C^*(\delta_j^{rx}) \in \underset{a \in \{\mathrm{i}, \mathrm{n}_j\}}{\arg\min} \left\{Q(\delta_j^{rx}, a)\right\}, \text{ where} \tag{12}$$

$$Q(\delta_j^{rx}, \mathrm{n}_j) \triangleq w_j \delta_j^{rx} + C + p_j h_C(\delta_j^{rx}+1) + (1-p_j) h_C(1),$$
$$Q(\delta_j^{rx}, \mathrm{i}) \triangleq w_j \delta_j^{rx} + h_C(\delta_j^{rx}+1).$$

Given (11) and (12), let $S_j^{\mathrm{n}_j}(C)$ represent the set of states the optimal action is equal to n_j for a given C, that is, $S_j^{\mathrm{n}_j}(C) = \{s : \pi_C^*(\delta_j^{rx}) = \mathrm{n}_j\}$. Then, we define indexability as follows.

Definition 1. *An arm is indexable if the set $S_j^{\mathrm{n}_j}(C)$ as a function of C is monotonically decreasing for $C \in \mathbb{R}$, and $\lim_{C\to\infty} S_j^{\mathrm{n}_j}(C) = \varnothing$ and $\lim_{C\to-\infty} S_j^{\mathrm{n}_j}(C) = \mathcal{S}$ [32], [33]. The problem is indexable if every arm is indexable.*

Note that if a problem is indexable as defined in Definition 1, $S_j^a(C_1) \subset S_j^a(C_2)$ for $C_1 \geq C_2$, and there exists a C such that both actions are *equally desirable*, that is, $Q(\delta_j^{rx}, \mathrm{i}) = Q(\delta_j^{rx}, \mathrm{n}_j)$ for all δ_j^{rx}. The WI for our problem is defined as follows.

Definition 2. *The WI for user j at state δ_j^{rx}, denoted by $I_j(\delta_j^{rx})$, is defined as the cost C that makes both actions n_j and i equally desirable.*

Next, we derive the WI for our problem:

Proposition 1. *Problem 1 with standard ARQ is indexable and the WI for each user j and state δ_j^{rx} can be computed as*

$$I_j(\delta_j^{rx}) = \frac{1}{2} w_j \delta_j^{rx} (1-p_j) \left(\delta_j^{rx} + \frac{1+p_j}{1-p_j}\right), \ \forall j \in [M], \tag{13}$$

where the WI for the idle action is $I_{M+1} = \eta$.

Proof. The proof is given in Appendix A. □

The WI policy is defined as follows: in state $(\delta_1^{rx}, \delta_2^{rx}, \dots, \delta_M^{rx})$, compare the highest index with the Lagrange parameter η, and if η is smaller, then the source transmits to the user with the highest index, otherwise the source remains idle. The WI policy, defined below, tends to transmit to the user with a high weight (w_j), low error probability (p_j) and high AoI (δ_j^{rx}). Formally,

$$\pi(\delta_1^{rx}, \dots, \delta_M^{rx}) = \begin{cases} \mathrm{n}_{\arg\max_j (I_j(\delta_j^{rx}))} & \text{if } \max_j I_j(\delta_j^{rx}) \geq \eta, \\ \mathrm{i} & \text{otherwise.} \end{cases} \tag{14}$$

The effectiveness of the WI policy is demonstrated in Section VIII. The WI policy, which corresponds to a suboptimal policy, can easily be shown to be optimal for our problem with standard ARQ if all the users are identical, i.e, $p_j = p$ and $w_j = w$, $\forall j$, and $\lambda = 1$.

B. Lower Bound under a Resource Constraint

In this section, we derive a closed-form lower bound for the constrained MDP:

Theorem 3. *For Problem 1 with the standard ARQ protocol, we have $J_{LB} \leq J^\pi$, $\forall \pi \in \Pi$, where*

$$J_{LB} = \frac{1}{2\lambda}\left(\sum_{j=1}^M \sqrt{\frac{w_j}{1-p_j}}\right)^2 + \frac{\lambda w_{j^*} p_{j^*}}{2(1-p_{j^*})} + \frac{1}{2}\sum_{j=1}^M w_j,$$

$$\text{and } j^* \triangleq \arg\min_j \frac{w_j p_j}{2(1-p_j)}.$$

Proof. The proof is provided in Appendix B. □

Previously, [13] proposed a lower bound on the average AoI for a source node sending time-sensitive information to multiple users through unreliable channels, without any resource constraint (i.e. $\lambda = 1$). The lower bound in Theorem 3 shows the effect of the constraint λ, and even for $\lambda = 1$, it is tighter than the one provided in [13].

VI. AoI with Fixed Redundancy (FR) HARQ Protocol

In this section, FR HARQ protocol [5], [10] is investigated. We assume that a generated status update contains k_s information symbols and encoding of a status update is performed using an (n_s, k_s)-MDS code [5], [10]. The transmission continues until n_s symbols are transmitted either successfully or not. The receiver starts decoding after n_s symbols are transmitted, and the AoI drops to n_s if at least k_s transmissions are successful, otherwise increases by n_s. In this case, each information symbol is considered as a packet transmitted in one time slot; that is, the minimum achievable age is n_s.

We note that other HARQ schemes can also be studied, e.g., chase combining when the base station retransmits the same packet and the receiver aggregates the energy from repeated transmissions to increase the signal to noise ratio (SNR), or incremental redundancy (IR) HARQ, which transmits additional redundancy bits in each retransmission and constantly adapts coding rate until successful decoding [34]. We consider a general HARQ model that can be adapted to both chase combining and IR. The particular HARQ protocol, i.e. FR HARQ with MDS coding, is chosen due to simplicity of computation and tractability of error probabilities. FR HARQ can be adapted to the general HARQ error probabilities with $r_{max} = n_s - 1$: $g(r) = 1 \ \forall r = \{0, \ldots, n_s - 2\}$ and $g(n_s - 1) = p_j^{FR} \triangleq$ Pr(less than k_s symbols are received among n_s transmissions) $=$ $\sum_{k=0}^{k_s-1} \binom{n_s}{k} p_j^{n_s-k}(1-p_j)^k$, $\forall j \in [M]$.

The problem for FR HARQ can be formulated as a RMAB problem, and whenever an arm (user) is chosen for transmission, a new update is generated and an encoded packet is transmitted for n_s time slots to that user. If the idle action is chosen, the source stays idle for a single time slot.

Similarly to Section V, low complexity heuristics based on WI and a lower bound on the average AoI are presented for FR HARQ protocol.

Proposition 2. *Problem 1 with the FR HARQ protocol is indexable and the WI for each user can be computed in closed form. We have*

$$I_j^{FR}(\delta_j^{rx}) = \frac{w_j}{2\left(\frac{n_s}{1-p_j^{FR}}\right)} \cdot \left(\left(\delta_j^{rx} + \frac{n_s p_j^{FR}}{1-p_j^{FR}} \right)^2 + \delta_j^{rx} + \frac{n_s p_j^{FR}}{1-\ p_j^{FR}} - \frac{n_s^2 p_j^{FR}}{(1-p_j^{FR})^2} \right), \tag{15}$$

where

$$p_j^{FR} \triangleq Pr(\textit{less than } k_s \textit{ symb. recv.}) = \sum_{k=0}^{k_s-1} \binom{n_s}{k} p_j^{n_s-k}(1-p_j)^k. \tag{16}$$

Proof. The proof is given in Appendix C. □

Following the WI policy presented in Proposition 2, the source tends to transmit to a user more frequently as the age, the weight, and the error probability of the user increases.

Theorem 4. *For Problem 1 with the FR HARQ Protocol, we have $J_{LB} \leq J^{\pi}$, $\forall \pi \in \Pi$, where*

$$J_{LB} = \frac{n_s}{2\lambda} \left(\sum_{j=1}^{M} \sqrt{\frac{w_j}{1-p_j^{FR}}} \right)^2 + \frac{\lambda n_s w_{j^*} p_{j^*}^{FR}}{2(1-p_{j^*}^{FR})} + \sum_{j=1}^{M} w_j \left(n_s - \frac{1}{2} \right),$$

$$\text{and } j^* \triangleq \arg\min_j \frac{w_j p_j^{FR}}{(1-p_j^{FR})}.$$

Proof. The proof is provided in Appendix D. □

For any given network with (w_j, p_j, $\forall j$, and λ) and FR HARQ (n_s, k_s) protocol, the average AoI that can be obtained under any casual policy is higher than the closed-form lower bound provided in Theorem 4. The expression in Theorem 4 provides an intuition on how the weights (w_j), the error probabilities (p_j), the average transmission constraint (λ), the number of users (M) and the design of MDS coding (n_s,k_s) affect the performance of the system in terms of average AoI.

Note that the results obtained for FR HARQ are identical to the ones obtained for standard ARQ protocol when (n_s, k_s) = $(1, 1)$. If (n_s, k_s) is different than $(1, 1)$, the average AoI result of Theorem 4 is equivalent to that of Theorem 3 scaled by n_s, where p_js are replaced by p_j^{FR} defined in (16).

VII. Learning in an unknown environment

In sections IV-VI, it is assumed that the channel statistics change very slowly and the same transmission environment has been used for a long time before the time of deployment, i.e., the statistics regarding the error probabilities are available. In most practical wireless settings, however, the channel error probabilities for retransmissions may not be known at the time of deployment, or may change over time. We employ online learning algorithms to learn the error probabilities over time without degrading the performance significantly. In our previous work [11], [12], [14], we proposed a simple *average-cost SARSA* algorithm to minimize the average AoI for a single user. Due to the large state space of the multi-user network considered in this paper, different learning algorithms are considered.

A. UCRL2 with HARQ

The upper confidence RL (UCRL2) algorithm [35] is a well-known RL algorithm for finite state and action MDP problems, with strong theoretical performance guarantees. However, the computational complexity of the algorithm scales quadratically with the size of the state space, which makes it unsuitable for large state spaces. UCRL2 has been initially proposed for generic MDPs with unknown rewards and transition probabilities; which need to be learned for each state-action pair. For the average AoI problem, the rewards are known (i.e., AoIs) while the transition probabilities are unknown. Moreover, the number of parameters to be learned can be reduced to the number of transmission error probabilities to each user; thus, the computational complexity can be reduced significantly.

For a generic tabular MDP, UCRL2 keeps track of the possible MDP models (transition probabilities and expected immediate rewards) in a high-probability sense and finds a policy that has the best performance in the best possible MDP. To achieve this in our case, it is enough to optimistically estimate the error probabilities $g_j(r)$, and find a policy that is optimal for the resulting optimistic MDP. This is possible since the performance corresponding to a fixed sequence of transmission decisions improves if the error probabilities decrease. The average transmission constraint at the source requires additional modifications to UCRL2. We will guarantee this constraint by updating the Lagrange multiplier according to the empirical resource consumption. The details of the algorithm are given in Algorithm 2.

UCRL2 exploits the optimistic MDP characterized by the optimistic estimation of error probabilities within a certain confidence interval, where $\hat{g}_j(r)$ and $\tilde{g}_j(r)$ represent the empirical and the optimistic estimates of the error probability for user j after r retransmissions. In each episode, we keep track of a value η resulting in a transmission cost close to λ, and then find and apply a policy that is optimal for the optimistic MDP (i.e., the MDP with the smallest total cost from among all plausible ones given the observations so far) with Lagrangian cost. In contrast to the original UCRL2 algorithm, finding the optimistic MDP in our case is easy (choosing lower estimates of the error probabilities), and we can use standard value iteration (VI) to compute the optimal policy (instead of the much more complex extended VI used in UCRL2). Thus, the computational complexity, which is the main drawback of UCRL2 algorithm, reduces significantly for the average AoI problem. UCRL2 is employed for Problem 1 in this paper since it is an online algorithm (i.e., it does not need any previous training) and it enjoys strong theoretical guarantees for $\lambda = 1$. The resulting algorithm will be called UCRL2-VI.

Algorithm 2 UCRL2-VI

```
Input: A confidence parameter $\rho \in (0,1)$, an update parameter $\alpha$, $\lambda$,
    confidence bound constant $U$, $|\mathcal{S}|$, $|\mathcal{A}|$
 1: $\eta = 0$, $t = 1$ and observe the initial state $s_1$.
 2: for episodes $k = 1, 2, \ldots$ do set $t_k \triangleq t$.
 3:   for $j \in [M]$, $r \in [r_{max}]$ do
 4:     $N_k(j,r) \triangleq |\{\tau < t_k : a_\tau = \mathrm{x}_j, r_{j,\tau} = r\}|$, $N_k(j,0) \triangleq |\{\tau < t_k : a_\tau = \mathrm{n}_j\}|$.
 5:     $E_k(j,r) \triangleq |\{\tau < t_k : a_\tau = \mathrm{x}_j, r_{j,\tau} = r, NACK\}|$, $E_k(j,0) \triangleq |\{\tau < t_k : a_\tau = \mathrm{n}_j, NACK\}|$.
 6:     $\hat{g}_j(r) \triangleq \frac{E_k(j,r)}{\max\{N_k(j,r),1\}}$.
 7:   end for
 8:   $C_k \triangleq |\{\tau < t_k : a_\tau \neq \mathrm{i}\}|$.
 9:   $\eta \leftarrow \eta + \alpha(C_k/t_k - \lambda)$.
10:   Compute optimistic error probability estimates: $\tilde{g}_j(r) \triangleq \max\left\{0, \hat{g}_j(r) - \sqrt{\frac{U \log(|\mathcal{S}||\mathcal{A}|t_k/\rho)}{max\{1, N_k(j,r)\}}}\right\}$.
11:   Use $\tilde{g}_j(r)$ and VI to find a policy $\tilde{\pi}_k$.
12:   Set $v_k(j,r) \leftarrow 0$, $\forall j, r$.
13:   while $v_k(j,r) < N_k(j,r)$ do /* run policy $\tilde{\pi}_k$ */
14:     Choose an action $a_t = \tilde{\pi}_k(s_t)$, and if $a_t \neq \mathrm{i}$, set $j_t$ the target user, otherwise set $j_t = 0$.
15:     Obtain cost $\sum_{j=1}^{M} w_j \delta_j^{rx} + \eta \mathbb{1}[a_t \neq \mathrm{i}]$ and observe $s_{t+1}$.
16:     Update $v_k(j_t, r) = v_k(j_t, r) + 1$ and set $t \leftarrow t + 1$.
17:   end while
18: end for
```

Algorithm 3 UCRL2 for the average AoI with ARQ.

```
Input: A confidence parameter $\rho \in (0,1)$, an update parameter $\alpha$, $\lambda$,
    confidence bound constant $U$, $|\mathcal{S}|$, $|\mathcal{A}|$
 1: $\eta = 0$, $t = 1$ and observe the initial state $s_1$.
 2: for episodes $k = 1, 2, \ldots$ do set $t_k \triangleq t$,
 3:   $N_k(j) \triangleq |\{\tau < t_k : a_\tau = \mathrm{n}_j\}|$, $E_k(j) \triangleq |\{\tau < t_k : a_\tau = \mathrm{n}_j, NACK\}|$
 4:   $\hat{p}(j) \triangleq \frac{E_k(j)}{\max\{N_k(j),1\}}$, $C_k \triangleq |\{\tau < t_k : a_\tau \neq \mathrm{i}\}|$,
 5:   $\eta \leftarrow \eta + \alpha(C_k/t_k - \lambda)$.
 6:   Compute the optimistic error probabilities: $\tilde{p}(j) \triangleq \max\{0, \hat{p}(j) - \sqrt{\frac{U \log(|\mathcal{S}||\mathcal{A}|t_k/\rho)}{max\{1, N_k(j)\}}}\}$
 7:   Use $\tilde{p}(j)$ to find a policy $\tilde{\pi}_k$ and execute policy $\tilde{\pi}_k$
 8:   while $v_k(j) < N_k(j)$ do
 9:     Choose an action $a_t = \tilde{\pi}_k(s_t)$,
10:     Obtain cost $\sum_{j=1}^{M} w_j \delta_j^{rx} + \eta * \mathbb{1}[a_t \neq \mathrm{i}]$ and observe $s_{t+1}$
11:     Update $v_k(j) = v_k(j) + 1$, set $t \leftarrow t + 1$;
12:   end while
13: end for
```

B. A Heuristic Version of the UCRL2 for Standard ARQ

In this section, we consider the standard ARQ protocol with unknown error probabilities $p_j = g_j(0)$. The estimation procedure of UCRL2-VI can be immediately simplified accordingly, as it only needs to estimate M parameters. In order to reduce the computational complexity, we can replace the costly VI in the algorithm to find the $\tilde{\pi}_k$ with the suboptimal WI policy given in Section V-A. The resulting algorithm, called *UCRL2-Whittle*, selects policy $\tilde{\pi}_k$ in step 16 following the WI policy in Section V. The details of the algorithm are given in Algorithm 3, where $\hat{p}(j)$ and $\tilde{p}(j)$ denote the empirical and the optimistic estimate of the error probability for user j.

C. Average-Cost SARSA with LFA

In [11], the average-cost SARSA algorithm is employed with *Boltzmann* (*softmax*) exploration for the average AoI problem with a single user. For the problem with multiple users, the cardinality of the state-action space is large and it is difficult to even store a matrix that has the size of the state-action space. Hence, average-cost SARSA with LFA is employed, where a linear function of features can be used to approximate the Q-function in SARSA [28]. Average-cost SARSA with LFA is an online algorithm similar to average-cost SARSA and UCRL2 algorithms. It improves the performance of average-cost SARSA by improving the convergence rate significantly for multi-user systems and its application is much simpler than the UCRL2 algorithm.

We approximate the Q function with a linear function Q_θ defined as: $Q_\theta(s,a) \triangleq \theta^T \phi(s,a)$, where $\phi(s,a) \triangleq (\phi_1(s,a), \ldots, \phi_d(s,a))^T$ is a given feature associated with the pair (s,a). In our experiments, we set $\{\phi_i(s,a)\}_{i=1}^{M}$ as the weighted age at the receiver of each user ($w_j \delta_j^{rx}$), $\{\phi_i(s,a)\}_{i=M+1}^{2M}$ as the age at the transmitter of each user (δ_j^{tx}) and $\{\phi_i(s,a)\}_{i=2M+1}^{3M}$ as the retransmission number of each user (r_j) given an action $a \in \mathcal{A}$ is chosen in state $s \in \mathcal{S}$:

$$
\begin{aligned}
Q_\theta(s,a) = {} & \theta_{(0,a)} + \theta_{(1,a)} w_1 \delta_1^{rx} + \ldots + \theta_{(M,a)} w_M \delta_M^{rx} + \theta_{(M+1,a)} \\
& w_1 \delta_1^{rx} + \ldots + \theta_{(2M,a)} w_M \delta_M^{rx} + \theta_{(2M+1,a)} r_1 + \ldots + \theta_{(3M,a)} r_M,
\end{aligned}
\tag{17}
$$

Algorithm 4 Average-cost SARSA with LFA

```
Input: Lagrange parameter $\eta$, update parameters $\alpha$, $\beta$, $\gamma$, $\mathcal{A}$, and set $t \leftarrow 1$ ,
    $\theta \leftarrow 0,\ J_\eta \leftarrow 0$
for $t = 1, 2, \ldots$ do
    Find the parameterized policies with Boltzmann exploration: $\pi(a|s_t) = \frac{\exp(-\theta^T\phi(s_t,a))}{\sum_{a'\in\mathcal{A}}\exp(-\theta^T\phi(s_t,a'))}$.
    Sample and execute action $a_t$ from $\pi(a|s_t)$.
    Observe the next state $s_{t+1}$ and cost $\sum_{j=1}^{M}\delta_j^{rx} + \eta * \mathbb{1}[a_t \neq \text{i}]$.
    $\pi(a|s_{t+1}) = \frac{\exp(-\theta^T\phi(s_{t+1},a))}{\sum_{a'_{t+1}\in\mathcal{A}}\exp(-\theta^T\phi(s_{t+1},a'))}$
    Sample $a_{t+1}$ from $\pi(a|s_{t+1})$
    Compute $C_\eta$
    Update linear coefficients: $\theta \leftarrow \theta + \alpha_t[\Delta + \eta \cdot \mathbb{1}[a_t \neq \text{i}] - J_\eta + \theta^T\phi(s_{t+1}, a_{t+1}) - \theta^T\phi(s_t, a_t)]\phi(s_t, a_t)$,
    Update gain: $J_\eta \leftarrow J_\eta + \beta_t[\Delta + \eta \cdot \mathbb{1}[a_t \neq \text{i}] - J_\eta]$,
    Update Lagrange multiplier: $\eta \leftarrow \eta + \gamma_t(C_\eta - \lambda)$
end for
```

where $\theta_{(0,a)}$ denotes the constant variable. The dimension of θ is $d = (3M + 1)|\mathcal{A}|$. The outline of the algorithm is given in Algorithm 4.

The performance of average cost SARSA with LFA is demonstrated in Section VIII. We note that linear approximators are not always effective, and the performance can be improved in general by using a non-linear approximator. However; the performance also depends on the availability of data, i.e., the linear approximator may perform better if the available data set is limited.)

D. Deep Q-Network (DQN)

A DQN uses a multi-layered neural network in order to estimate the values of $Q(s, a)$; that is, for a given state s, DQN outputs a vector of state-action values, $Q_\theta(s, a)$, where θ denotes the parameters of the network. That is, the neural network is a function from $2M$ inputs to $|\mathcal{A}|$ outputs which are the estimates of the Q-function $Q_\theta(s, a)$. We apply the DQN algorithm of [36] to learn a scheduling policy. We create a fairly simple feed-forward neural network of 3 layers, one of which is the hidden layer with 24 neurons. We also use *Huber loss* [37] and the *Adam* algorithm [38] to conduct stochastic gradient descent to update the weights of the neural network.

We exploit two important features of DQNs as proposed in [36]: *experience replay* and a *fixed target network*, both of which provide algorithm stability. For *experience replay*, instead of training the neural network with a single observation $<s, a, s', c(s, a)>$ at the end of each step, many experiences (i.e., (state, action, next state, cost) quadruplets) can be stored in the replay memory for batch training, and a minibatch of observations randomly sampled at each step can be used. The DQN uses two neural networks: a target network and an online network. The *target network*, with parameters θ^-, is the same as the online network except that its parameters are updated with the parameters θ of the online network after every T steps, and θ^- is kept fixed in other time slots. For a minibatch of of observations for training, temporal difference estimation error e for a single observation can be calculated as

$$e = Q_\theta(s, a) - (-c(s, a) + \gamma Q_{\theta^-}(s', \arg\max Q_\theta(s', a))). \tag{18}$$

Huber loss is defined by the squared error term for small estimation errors, and a linear error term for high estimation errors, allowing less dramatic changes in the value functions and further improving the stability. For a given estimation error e and loss parameter d, the Huber loss function, denoted by $\mathrm{L}^d(e)$, and the average loss over the minibatch, denoted by $\mathcal{B}$, are computed as

$$\mathrm{L}^d(e) = \begin{cases} e^2 & \text{if } e \leq d \\ d(|e| - \frac{1}{2}d)) & \text{if } e > d, \end{cases} \quad \text{and } \mathrm{L}_{\mathcal{B}} = \frac{1}{|\mathcal{B}|} \sum_{<s,a,s',c(s,a)>\in\mathcal{B}} \mathrm{L}^d(e).$$

Table I
HYPERPARAMETERS OF DQN ALGORITHM USED IN THE PAPER

Parameter	Value	Parameter	Value
discount factor γ	0.99	optimizer	Adam
minibatch size	32	loss function	Huber loss
replay memory length	2000	exploration coefficient ϵ_0	1
activation function	ReLU	learning rate α	10^{-4}
hidden size	24	ϵ decay rate β	0.9
episode length T	1000	ϵ_{min}	0.01

Table II
A SUMMARY OF RL ALGORITHMS PRESENTED IN THIS PAPER

RL Method	Advantages	Disadvantages
RVI [28]	simple, converges to optimal for MDPs	requires apriori information on system characteristics
Average cost SARSA (tabular) [12]	simple, fully online	does not perform well for large state spaces, requires an approximation to finite state spaces
Average cost SARSA with LFA	converges faster than Average cost SARSA applicable to infinite state spaces	convergences slower than UCRL2 and DQN, stability issues for average cost problems
UCRL2-VI	theoretical convergence guarantee	large computational complexity due to VI
UCRL2-Whittle	low computational complexity	based on the computation of WI
DQN [36]	performs well and applicable to infinite or large state spaces	requires pre-training

We apply the ϵ-greedy policy to balance exploration and exploitation, i.e., with probability ϵ the source randomly selects an action, and with probability $1 - \epsilon$ it chooses the action with the minimum Q value. We let ϵ decay gradually from ϵ_0 to ϵ_{min}; in other words, the source explores more at the beginning of training and exploits more at the end. The hyperparameters of the DQN algorithm are tuned for our problem experimentally, and are given in Table I.

VIII. NUMERICAL RESULTS

In this section, we provide numerical results for the proposed learning algorithms, and compare the achieved average performances. First, we analyze the average AoI with the standard ARQ protocol. The asymptotic average AoI as a function of the resource constraint λ is shown in Figure 3 for a 3-user system with error probabilities $p = g(0) = [0.5\ 0.2\ 0.1]$. It can be seen from Figure 3 that both UCRL2-VI and UCRL2-Whittle perform very close to the lower bound, particularly when λ is small, i.e., the system is more constrained. Although UCRL2-Whittle has a significantly lower computational complexity, it performs very close to UCRL2-VI for all λ values.

Figure 4 illustrates the mean and variance of the average AoI with standard ARQ with respect to the size of the network

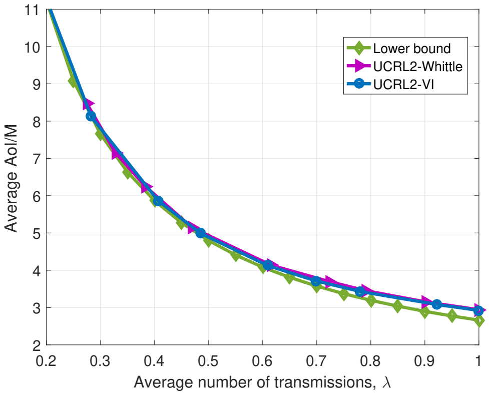

Figure 3. Average AoI with respect to λ for a 3-user network under the standard ARQ protocol, with error probabilities $p = [0.5\ 0.2\ 0.1]$, and $w_j = 1,\ \forall j$. Time horizon is set to $T = 10^5$, and the results are averaged over 100 runs.

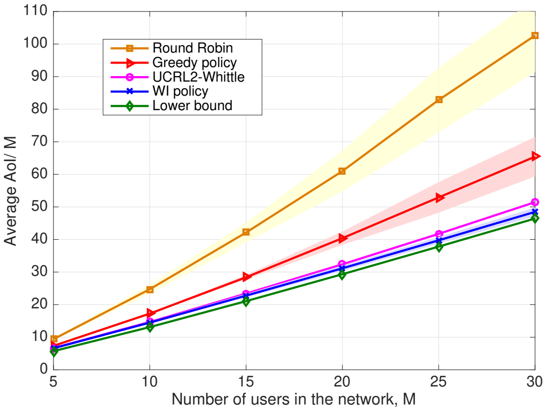

Figure 4. Average AoI under the standard ARQ protocol for networks with different sizes, where $p_j = (j-1)/M$, $\lambda = 1$, and $w_j = 1,\ \forall j$. The results are obtained after 10^4 time steps and averaged over 100 runs (both the mean and the variance are shown).

when there is no constraint on the average number of transmissions (i.e. $\lambda = 1$) and the performance of the UCRL2-Whittle is compared with the lower bound (UCRL2-VI is omitted since its performance is very similar to UCLR2-Whittle and has a much higher computational complexity, especially for large M). The performance of UCRL2-Whittle is close to the lower bound and is very similar to that of the WI policy, which requires a priori knowledge of the error probabilities. Moreover, our algorithm outperforms the benchmark *greedy policy*, which always transmits to the user with the highest age (i.e., $a = \mathrm{n}_j$, such that $j = argmax\delta_j^{rx}, \forall j \in [M]$), as well as the *round robin policy*, which transmits to each user in turns.

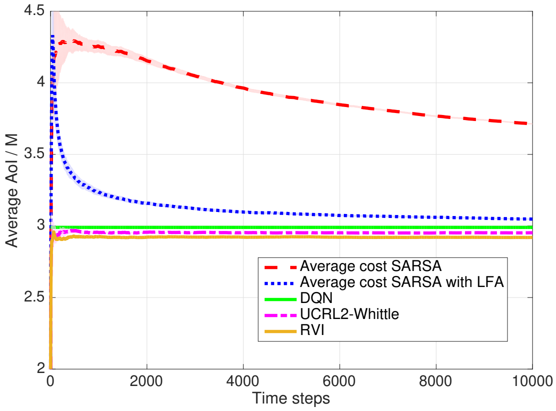

Figure 5. Average AoI for a 3-user ARQ network with error probabilities $p = [0.5\ 0.2\ 0.1]$, $\lambda = 1$, and $w_j = 1,\ \forall j$. The results are obtained after 10^4 time steps and averaged over 100 runs (both the mean and the variance are shown).

The performance of the proposed RL algorithms (the mean and the variance of the average AoI) is compared with the performance of average cost SARSA, proposed in [11] for a point-to-point status update system (which is used as a benchmark policy in this paper), in Figure 5. The DQN algorithm in the figure is configured as in Table I and trained for 500 episodes. The average AoI for DQN is obtained after 10^5 time steps and averaged over 100 runs. UCRL2-Whittle and average cost SARSA with LFA converge much faster compared to the standard average-cost SARSA, and they perform very close to the transmission scheduling computed by RVI with known error probabilities. Although DQN and UCRL2-Whittle perform better than average cost SARSA with LFA, DQN requires a training time before running the simulation.

Figure 6 shows the performance of the learning algorithms for the HARQ protocol (the mean and the variance of the average AoI) for a 2-user scenario. Similarly to Figure 5, DQN is trained for 500 episodes with configuration in Table I. It is worth noting that although UCRL2-VI converges to the optimal policy in fewer iterations than average-cost SARSA and average-cost SARSA with LFA, iterations in UCRL2-VI are computationally more demanding since the algorithm uses VI in each epoch. Therefore, UCRL2-VI is not practical for problems with large state spaces, in our case for large M. On the other hand, UCRL2-Whittle can handle a large number of users since it is based on a simple index policy instead of VI. As illustrated in both Figures 5 and 6 that LFA significantly improves the performance of average cost SARSA and DQN with neural network estimator, and UCRL2-Whittle improves the performance of RL even more.

The performance of FR HARQ protocol as described in Section VI for packets MDS-coded with $(n_s, k_s) = (5, 3)$ is shown in Figure 7. The probability that an MDS-coded packet is not correctly decoded is given in (16) where the symbol transmission error probability is set to $p_{s,j} = (j-1)/2M$ for user j. As Figure 7 illustrates, average AoI per user increases linearly with the number of users in the network and the WI policy performs very close to the lower bound for both $\lambda = 0.6$ and $\lambda = 1$.

Figure 8 shows the evolution of average AoI across 10 users with DQN after different number of training episodes, where each training episode consists of 1000 time steps. Following [26], [27], r_{max} is set to 3 and general HARQ protocol is considered with $g_j(r_j) = (j-1)/M \cdot 2^{-r_j}$ motivated by

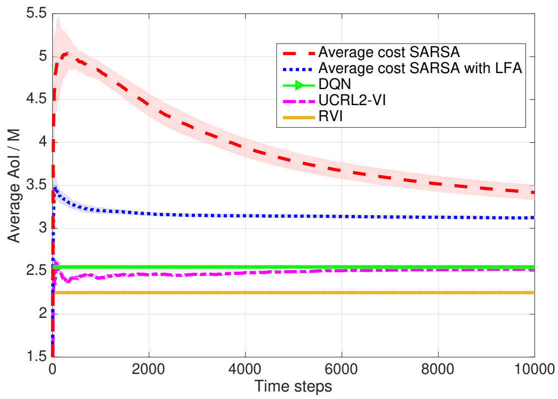

Figure 6. Average AoI for a 2-user HARQ network with error probabilities $g_1(r_1) = 0.5{\cdot}2^{-r_1}$ and $g_2(r_2) = 0.2{\cdot}2^{-r_2}$, where $\lambda = 1$ and $w_j = 1,\ \forall j$. The simulation results are averaged over 100 runs (both the mean and the variance are shown).

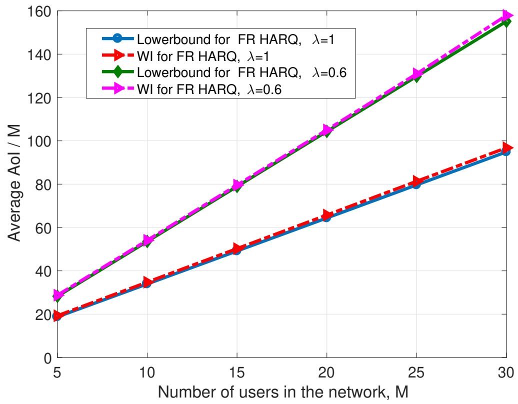

Figure 7. Average AoI for networks with different sizes and constraints, where $p_{s,j} = (j-1)/2M$, $(n_s, k_s) = (5, 3)$, and $w_j = 1,\ \forall j$. The results are obtained after 10^4 time steps and averaged over 100 runs.

the exponentially decreasing error profile of HARQ protocols studied in [25], [34]. Figure 8 illustrates that average AoI achieves its minimum after about 150 episodes of training. We note that we did not run UCRL2-VI and average-cost SARSA algorithms for a 10-user HARQ problem since state space is large and convergence takes too long compared to the DQN algorithm.

A summary of the employed RL algorithms together with their strengths and weakness are given in Table II. We concluded that the choice of the learning algorithm to be adopted depends on the scenario and system characteristics. It has been shown that average-cost SARSA is not effective considering the large state space of the multi-user problem. Different state-of-the-art RL methods are presented including SARSA with LFA, UCRL2, and DQN. The performance of UCRL2-VI algorithm is close to optimal for small networks, i.e. consisting of 1-5 users, and enjoys theoretical guarantees. However, UCRL2-VI is not favorable for large networks due to its computational complexity resulting from value iteration, and UCRL2-Whittle is preferable. On the other hand, UCRL2-Whittle cannot be employed for a general HARQ multi-user system. Similarly, SARSA with LFA has decreased the average AoI significantly for small-size networks with HARQ; however, is not effective for large networks and SARSA with LFA lacks stability. A non-linear approximation with DQN performs well for large networks, while it is not fully online and it requires a training time before running the algorithm.

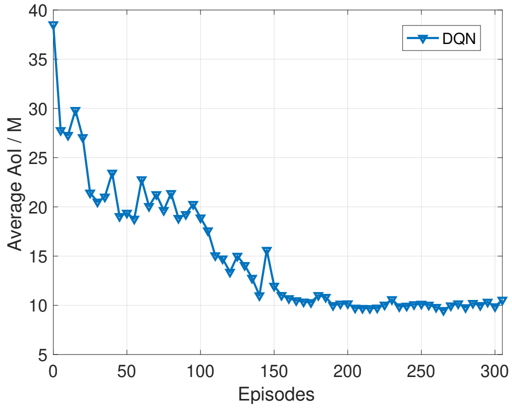

Figure 8. Average AoI obtained by DQN algorithm with respect to the number of training episodes for a 10-user HARQ network with error probabilities $g_j(r_j) = (j-1)/M \cdot 2^{-r_j}$, $j \in [M]$ and $r_j \in [r_{max}]$, $\lambda = 1$, $r_{max} = 3$ and $w_j = 1,\ \forall j$. Each episode consists of 1000 time steps and the results are obtained for a single run.

IX. Conclusion

We considered scheduling the transmission of status updates to multiple users with the weighted average AoI as the performance measure. Under a resource constraint at the source node, the problem is modeled as a CMDP and the structure of the optimal policy is established. Lower bounds on the average AoI are derived for special cases of the problem. RL algorithms were presented for scenarios where the error probabilities may not be known in advance, and were numerically shown to provide near-optimal performance in simple scenarios. It has been demonstrated that the optimal choice of the learning algorithm to be adopted depends on the scenario and system characteristics. The algorithms presented in this paper are also relevant to other multi-user systems concerning the timeliness of information. The AoI for multi-user systems without feedback, or under imperfect and delayed ACK/NACK feedback will be studied as a future work.

Appendix

A. *Proof of Proposition 1*

We note that each sub-problem (11) coincides with the Lagrangian formulation of the single-user problem which we previously studied in [11], [12]. According to Lemma 1 of [12], the policy which solves the Bellman optimality equations is a threshold policy, such that $a = \mathrm{n}_j$ if and only if $\delta_j^{rx} \geq \Gamma_j$ for an appropriate threshold Γ_j. As a consequence, the cost function (the sum of the average AoI and the average transmission cost for that user), given a threshold Γ_j and cost of transmission C, can also be obtained in closed form as

$$L_C^{\Gamma_j} = \frac{1}{\Gamma_j + \frac{p_j}{1-p_j}} \left(\frac{(\Gamma_j - 1)\Gamma_j}{2} + \frac{C + \Gamma_j}{1 - p_j} + \frac{p_j}{(1-p_j)^2} \right). \tag{19}$$

Using Definition 2 and (19), we can compute the WI in closed form: By the definition of the threshold policy, we can find a C such that both choices of thresholds $\Gamma_j = \delta_j^{rx}$ and $\Gamma_j = \delta_j^{rx} + 1$ result in the same average cost, i.e., the average cost $L_C^{\delta_j^{rx}}$ should be equal to $L_C^{\delta_j^{rx}+1}$, which can be computed using (19):

$$I_j(\delta_j^{rx}) = \frac{1}{2} w_j \delta_j^{rx} (1 - p_j)(\delta_j^{rx} + \frac{1 + p_j}{1 - p_j}). \tag{20}$$

The $(M+1)^{th}$ arm stands for the idle action. The Lagrange multiplier η represents the cost of transmission and C represents the cost of staying idle. If η^* is equal to C then both actions are equally desirable; that is, the WI for the $M + 1^{th}$ arm is $I_{M+1} = \eta^*$.

Note that an optimal threshold Γ_j^* for a given C, which minimizes (19), can be computed for a given C as follows:

$$\Gamma_j^* \in \left\{ \left\lfloor \frac{\sqrt{2C(1-p_j)+p_j}-p_j}{1-p_j} \right\rfloor, \left\lceil \frac{\sqrt{2C(1-p_j)+p_j}-p_j}{1-p_j} \right\rceil \right\}.$$

As C increases from 0 to ∞, Γ_j^* monotonically increases from 0 to ∞, and $S_j^{\mathrm{n}_j}(C)$ monotonically decreases from the entire state space $\mathcal{S}$ to an empty set. Thus, the problem is indexable. □

B. Proof of Theorem 3

Proof. The system model and the definition of action $a_t \in \mathcal{A}$ implies the following constraints in addition to the average number of transmissions constraint in Problem 1: (i) updates occur in discrete time slots, and (ii) collisions are not allowed, i.e., no more than one user can be updated in a slot. In order to derive the lower bound, we relax the constraints (i) and (ii).

First, we relax constraint (ii) and decouple the model to M point-to-point status update systems each with a single user to serve. Let J_j^π and C_j^π denote respectively, the expected average AoI and the expected average number of transmissions for user j if we follow policy π. Assume that each user j has an average number of transmissions constraint of λ_j is imposed on user j, and we have:

$$J^* \geq \sum_{j=1}^{M} w_j J_j^* \geq \sum_{j=1}^{M} w_j J_{j,LB}, \quad \text{given that } \sum_{j}^{M} \lambda_j = \lambda, \tag{21}$$

where $J_j^* \leq J_j^\pi$, $\forall \pi$, denotes the minimum expected average AoI for user j given $C_j^\pi \leq \lambda_j$ and $J_{j,LB}$ denotes a lower bound on the average AoI for user j.

The first inequality in (21) results from the relaxation of (ii) and decoupling the users. Then, we minimize the average AoI for a single user j under a constraint $C_j \leq \lambda_j$, which reduces Problem 1 to a single user problem which we previously studied in Section V of [12]. The second inequality in (21) is due to relaxing the discrete time assumption in (i) in order to find a lower bound $J_{j,LB}$ on the average AoI for user j by using closed form average AoI and resource consumption expressions also obtained in [12].

According to Theorem 2 of [12], the policy that solves the Bellman optimality equations is a threshold policy. Then, for a given threshold, the expected average AoI and average number of transmissions can be computed in closed form similarly to [12]:

$$J_j^{\Gamma_j} = \frac{(\Gamma_j(1-p_j) + p_j)^2 + p_j}{2(1-p_j)(\Gamma_j(1-p_j)+p_j)} + \frac{1}{2}, \; C_j^{\Gamma_j} = \frac{1}{\Gamma_j(1-p_j)+p_j}. \tag{22}$$

The lower bound $J_{j,LB}$ on the average AoI for a single user can be computed by substituting $C_j^{\Gamma_j}$ into $J_j^{\Gamma_j}$ and using the constraint $C_j^{\Gamma_j} \leq \lambda_j$:

$$J_j^{\Gamma_j} = \frac{1/C_j^2 + p_j}{2(1-p_j)/C_j} + \frac{1}{2} \geq \frac{1/\lambda_j^2 + p_j}{2(1-p_j)/\lambda_j} + \frac{1}{2} \triangleq J_{j,LB}, \tag{23}$$

and so $J_j^* \geq J_{j,LB}$ since $J_j^{\Gamma_j} \geq J_{j,LB}$ for all Γ_j values satisfying $C_j^{\Gamma_j} \leq \lambda_j$. By inserting (23) to (21), we obtain

$$J^* \geq \sum_{j=1}^{M} w_j \left(\frac{1/\lambda_j^2 + p_j}{2(1-p_j)/\lambda_j} + \frac{1}{2} \right) \quad \text{given that } \sum_{j}^{M} \lambda_j = \lambda. \tag{24}$$

Note that the right hand side (RHS) of (24) is a convex function of $\lambda_1, \ldots, \lambda_M$. Then, the optimal λ_j to minimize the RHS of (24) can be found numerically when p_j and λ are given. However, in order to obtain a closed form solution which can be easily computed and compared to the state-of-art bounds in the literature, we approximate the RHS and obtain a slightly looser bound on the performance in closed form:

$$J^* \geq \min_{\lambda_1,\ldots,\lambda_M : \sum_{i=1}^{M} \lambda_i = \lambda} \sum_{j=1}^{M} \frac{w_j}{2\lambda_j(1-p_j)} \tag{25a}$$

$$+ \min_{\lambda_1,\ldots,\lambda_M : \sum_{i=1}^{M} \lambda_i = \lambda} \sum_{j=1}^{M} \frac{w_j \lambda_j p_j}{2(1-p_j)} + \frac{1}{2} \sum_{j=1}^{M} w_j \tag{25b}$$

$$= \frac{1}{2\lambda} \left(\sum_{j=1}^{M} \sqrt{\frac{w_j}{1-p_j}} \right)^2 + \frac{\lambda}{2} \min_j \left(\frac{w_j p_j}{1-p_j} \right) + \frac{1}{2} \sum_{j=1}^{M} w_j. \tag{25c}$$

Here inequality (25b) results from the fact that independently minimizing the terms of a sum is smaller than the minimization of the sum; and the first term in (25c) is equal to the minimum of the first term in (25b) with the constraint of $\sum_{j=1}^{M} \lambda_j = \lambda$, where a solution for λ_j is found using the Lagrangian relaxation and the Karush–Kuhn–Tucker conditions [39], leading to

$$\lambda_j^* = \frac{\sqrt{w_j/(1-p_j)}}{\sum_{i=1}^{M} \sqrt{w_i/(1-p_i)}}. \tag{26}$$

The second term in (25c) is the minimum of the second term in (25b) under the constraint of $\sum_{j=1}^{M} \lambda_j = \lambda$, which proves the theorem. □

C. Proof of Proposition 2

Following similar steps to Appendix A, the problem can be approximated by decoupling the system into M point-to-point status update systems with FR HARQ, where the cost of a single transmission is C. The state-action cost function and optimality equations are given.

$$h_c(\delta,0)=\min\left(Q(\delta,0,\mathrm{n}_j),Q(\delta,0,\mathrm{i})\right), \tag{27}$$

$$Q(\delta,0,\mathrm{n}_j)=\left(\sum_{n=0}^{n_s-1}\delta+n+C-L_j^C\right)+p_e h_C(\delta+n_s,0)+(1-p_e)h_C(n_s,0), \tag{28}$$

$$Q(\delta,0,\mathrm{i})=\left(\sum_{n=0}^{n_s-1}\delta+n-L_j^C\right)+h_c(\delta+n_s,0). \tag{29}$$

Next we investigate the optimal policy in the subsystem for a single user (user j), treated independently from the other decisions, which can be shown to be of threshold type:

Lemma 1. *The decision to start transmitting to user j ($a_j=\mathrm{n}_j$) is monotone with respect to the age δ_j^{rx}, that is if $a_j^*(\delta^1,0)=\mathrm{n}_j$, then $a_j^*(\delta^2,0)=\mathrm{n}_j$ for all $\delta^2\geq\delta^1$.*

Proof. A monotone threshold policy is optimal if $Q(\delta_j^{rx},0,a_j)$ has a *sub-modular* structure in (δ_j^{rx},a_j) [40], that is,

$$Q(\delta^1,\mathrm{n}_j)-Q(\delta^1,\mathrm{i})\geq Q(\delta^2,\mathrm{n}_j)-Q(\delta^2,\mathrm{i}), \tag{30}$$

for any $\delta^2\geq\delta^1$. From (29) and (28), for any $\delta>0$, we have

$$Q(\delta,0,\mathrm{n}_j)-Q(\delta,0,\mathrm{i})=n_sC+(1-p_e)h_C(n_s,0)-(1-p_e)h_C(\delta+n_s). \tag{31}$$

We can see that (30) holds if and only if $h_C(\delta,0)$ is a non-decreasing function of the age. We compare the costs incurred by the systems starting in states δ^1 and δ^2 via coupling the stochastic processes governing the behavior of the system; that is, we assume that the realization of the channel behavior is the same for both systems over the time horizon (this is valid since channel states/errors are independent of the ages and the actions). Assume a sequence of actions $\{a_t^2\}_{t=1}^{\infty}$ corresponds to the optimal policy starting from age δ^2 for a particular realization of channel errors, and let $\{\delta_t^i\}$ denote the sequence of states obtained after following actions $\{a_t^2\}$ starting from state $\delta_1^{rx}=\delta^i$, $i=1,2$. Then, if $\delta^1\leq\delta^2$, clearly $\delta_t^1\leq\delta_2^{rxt}$ for all t. Furthermore, by the Bellman optimality equation (5),

$$h_C(\delta^1,0)\leq\mathbb{E}\left[\sum_{t=1}^{\infty}(\delta_t^1+C\cdot\mathbb{1}[a_t^2\neq\mathrm{i}]-L_j^*)\middle|\delta_1^1=\delta^1\right]$$
$$\leq\mathbb{E}\left[\sum_{t=1}^{\infty}(\delta_t^2+C\cdot\mathbb{1}[a_t^2\neq\mathrm{i}]-L_j^*)\middle|\delta_1^1=\delta^2\right]=h_C(\delta^2,0)\ .$$

This completes the proof of the lemma. □

Note that under a threshold policy with threshold Γ_j, the AoI process is a renewal process with i.i.d. renewal periods of $X\triangleq\Gamma_j-n_s+S_j$, where S_j is the random time between the start of transmission of a status update and successful decoding of that update at the receiver of user j similarly to [5], [10], [21]. Then, the average AoI can be be written as expectation of the area under the AoI graph divided by the expected value of X (denoted by $\mathbb{E}[X]$), which is given by:

$$J_j^{\Gamma_j}=\frac{\mathbb{E}\left[S^2\right]+(\Gamma_j-n_s)\mathbb{E}\left[S_j\right]}{2((\Gamma_j-n_s)+\mathbb{E}\left[S_j\right])}+\frac{\Gamma_j+n_s}{2}-\frac{1}{2}, \tag{32}$$

where the constant $(-1/2)$ results from the fact that we consider a stair-step function to represent the AoI. Similarly, the average number transmissions is given by:

$$C_j^{\Gamma_j}=\frac{\mathbb{E}\left[S_j\right]}{\mathbb{E}\left[X\right]}=\frac{\mathbb{E}\left[S_j\right]}{(\Gamma_j-n_s)+\mathbb{E}\left[S_j\right]}. \tag{33}$$

Thus, we have

$$L_C^{\Gamma_j}=\frac{\mathbb{E}\left[S_j^2\right]+(\Gamma_j-n_s)\mathbb{E}\left[S_j\right]}{2((\Gamma_j-n_s)+\mathbb{E}\left[S_j\right])}+\frac{\Gamma_j+n_s}{2}-\frac{1}{2}+C\frac{\mathbb{E}\left[S_j\right]}{(\Gamma_j-n_s)+\mathbb{E}\left[S_j\right]}, \tag{34}$$

where $\mathbb{E}\left[S_j\right]=\frac{n_s}{1-p_j^{FR}}$, and $\mathbb{E}\left[S_j^2\right]=\frac{n_s^2(1+p_j^{FR})}{(1-p_j^{FR})^2}$ for FR protocol [5] and p_j^{FR} is given as in (16).

Next steps for the derivation of WI is similar to Appendix A. By the definition of threshold policy, we solve (34) and try to find a $I_j^{FR}(\delta_j^{rx})\triangleq C$ such that actions of staying idle and starting the transmission to user j are equally desirable, that is, $L_C^{\delta_j^{rx}}$ and $L_C^{\delta_j^{rx}+1}$ give the same result. □

D. Proof of Theorem 4

The proof is similar to that of Theorem 3 in Appendix B. First, constraints of (i) and (ii) are relaxed and we obtain (21). J_j^* for FR HARQ protocol can be computed using (32) and (33). Note that both $J_j^{\Gamma_j}$ and $C_j^{\Gamma_j}$ are convex functions of Γ_j and $J_j^{\Gamma_j}$ can be written in terms of $C_j^{\Gamma_j}$

$$J_j^{\Gamma_j}=\frac{\mathbb{E}\left[S_j\right]}{2C_j^{\Gamma_j}}+C_j^{\Gamma_j}\frac{\mathbb{V}\left[S_j\right]}{2\mathbb{E}\left[S_j\right]}+n_s-\frac{1}{2}, \tag{35}$$

where $\mathbb{E}\left[S_j\right]$ and $\mathbb{V}\left[S_j\right]$ denote the expected value and the variance of S_j. $J_j^{\Gamma_j}$ is a convex increasing function of $C_j^{\Gamma_j}$ and $C_j^*\leq\lambda_j$. Then, by inserting (35) into (21), similarly to Appendix A, the lower bound for FR HARQ can be computed in closed form as follows:

$$J^*\geq\min\left(\sum_{j=1}^{M}w_j\left(\frac{\mathbb{E}\left[S_j\right]}{2\lambda_j}+\lambda_j\frac{\mathbb{V}\left[S_j\right]}{2\mathbb{E}\left[S_j\right]}+n_s-\frac{1}{2}\right)\right) \tag{36a}$$

$$=\frac{1}{2\lambda}\left(\sum_{j=1}^{M}\sqrt{w_j\mathbb{E}\left[S_j\right]}\right)^2+\frac{\lambda}{2}\min_j\left(\frac{\mathbb{V}\left[S_j\right]}{\mathbb{E}\left[S_j\right]}\right)+\sum_{j=1}^{M}w_j\left(n_s-\frac{1}{2}\right), \tag{36b}$$

which is equal to the bound in Theorem 4 where $\mathbb{E}\left[S_j\right]=\frac{n_s}{1-p_j^{FR}}$, and $\mathbb{V}\left[S_j\right]=\mathbb{E}\left[S_j^2\right]-\mathbb{E}\left[S_j\right]^2=\frac{n_s^2p_j^{FR}}{(1-p_j^{FR})^2}$ for FR. □

References

[1] E. T. Ceran, D. Gündüz, and A. György, "Reinforcement learning approach to age of information in multi-user networks," in *IEEE Int. Symp. on Personal, Indoor, and Mobile Radio Comms. (PIMRC)*, 2018.

[2] E. Altman, R. E. Azouzi, D. S. Menasché, and Y. Xu, "Forever young: Aging control in DTNs," *CoRR, abs/1009.4733*, 2010.

[3] S. Kaul, M. Gruteser, V. Rai, and J. Kenney, "Minimizing age of information in vehicular networks," in *IEEE Coms. Society Conference on Sensor, Mesh and Ad Hoc Coms. and Nets.*, June 2011, pp. 350–358.

[4] S. Kaul, R. Yates, and M. Gruteser, “Real-time status: How often should one update?” in *Proc. IEEE INFOCOM,*, March 2012, pp. 2731–2735.

[5] E. Najm, R. Yates, and E. Soljanin, “Status updates through M/G/1/1 queues with HARQ,” in *IEEE International Symposium on Information Theory (ISIT)*, June 2017, pp. 131–135.

[6] H. B. Beytur and E. Uysal, “Age minimization of multiple flows using reinforcement learning,” in *2019 International Conf. on Computing, Networking and Comms. (ICNC)*, 2019, pp. 339–343.

[7] B. T. Bacinoglu, E. T. Ceran, and E. Uysal-Biyikoglu, “Age of information under energy replenishment constraints,” in *2015 Information Theory and Applications Workshop (ITA)*, Feb 2015, pp. 25–31.

[8] Y. Sun, E. Uysal-Biyikoglu, R. Yates, C. E. Koksal, and N. B. Shroff, “Update or wait: How to keep your data fresh,” in *IEEE Int. Conf. on Computer Comms. (INFOCOM)*, April 2016, pp. 1–9.

[9] Y. P. Hsu, E. Modiano, and L. Duan, “Age of information: Design and analysis of optimal scheduling algorithms,” in *IEEE International Symposium on Information Theory (ISIT)*, June 2017, pp. 561–565.

[10] R. D. Yates, E. Najm, E. Soljanin, and J. Zhong, “Timely updates over an erasure channel,” in *IEEE Int. Symp. on Inf. Theory*, 2017, pp. 316–320.

[11] E. T. Ceran, D. Gündüz, and A. György, “Average age of information with hybrid ARQ under a resource constraint,” in *IEEE Wireless Communications and Networking Conference (WCNC)*, 2018.

[12] E. T. Ceran, D. Gündüz, and A. György, “Average age of information with hybrid arq under a resource constraint,” *IEEE Transactions on Wireless Communications*, vol. 18, pp. 1900–1913, March 2019.

[13] I. Kadota, E. Uysal-Biyikoglu, R. Singh, and E. Modiano, “Scheduling policies for minimizing age of information in broadcast wireless networks,” *IEEE/ACM Trans. on Netw.*, vol. 26, pp. 2637–2650, 2018.

[14] E. T. Ceran, D. Gündüz, and A. György, “Reinforcement learning to minimize age of information with an energy harvesting sensor with HARQ and sensing cost,” in *IEEE Conf. on Computer Comms. Workshops (INFOCOM WKSHPS)*, April 2019.

[15] D. Gunduz, K. Stamatiou, N. Michelusi, and M. Zorzi, “Designing intelligent energy harvesting communication systems,” *IEEE Communications Magazine*, vol. 52, pp. 210–216, 2014.

[16] Q. He, D. Yuan, and A. Ephremides, “Optimal link scheduling for age minimization in wireless systems,” *IEEE Trans. on Inf. Theory*, 2017.

[17] R. D. Yates and S. K. Kaul, “Status updates over unreliable multiaccess channels,” in *IEEE Int. Symp. on Inf. Theory*, June 2017, pp. 331–335.

[18] J. Zhong, E. Soljanin, and R. D. Yates, “Status updates through multicast networks,” in *Allerton Conf. on Comm., Cont., and Comp.*, 2017, pp. 463–469.

[19] A. M. Bedewy, Y. Sun, S. Kompella, and N. B. Shroff, “Optimal sampling and scheduling for timely status updates in multi-source networks,” 2020.

[20] A. Arafa, K. Banawan, K. G. Seddik, and H. V. Poor, “On timely channel coding with hybrid ARQ,” *CoRR*, vol. abs/1905.03238, 2019.

[21] E. Najm, E. Telatar, and R. Nasser, “Optimal age over erasure channels,” in *IEEE Int. Symp. on Inf. Theory (ISIT)*, 2019, pp. 335–339.

[22] S. Leng and A. Yener, “Age of information minimization for wireless ad hoc networks: A deep reinforcement learning approach,” in *2019 IEEE Global Communications Conference (GLOBECOM)*, 2019, pp. 1–6.

[23] M. A. Abd-Elmagid, A. Ferdowsi, H. S. Dhillon, and W. Saad, “Deep reinforcement learning for minimizing age-of-information in uav-assisted networks,” in *IEEE Global Comms. Conf.(GLOBECOM)*, 2019, pp. 1–6.

[24] A. Elgabli, H. Khan, M. Krouka, and M. Bennis, “Reinforcement learning based scheduling algorithm for optimizing age of information in ultra reliable low latency networks,” 01 2019.

[25] V. Tripathi, E. Visotsky, R. Peterson, and M. Honig, “Reliability-based type ii hybrid ARQ schemes,” in *IEEE International Conference on Communications,*, vol. 4, May 2003, pp. 2899–2903 vol.4.

[26] “IEEE standard for local and metropolitan area networks part 16: Air interface for fixed and mobile broadband wireless access systems amendment 2 (incorporated into IEEE standard 802.16e-2005 and std 802.16-2004/cor1-2005),” 2006.

[27] [Online]. Available: "http://rfmw.em.keysight.com/wireless/helpfiles/n7625bpxb/Content/RT/UL-SCHSettings.htmMaximumNumberofRetransmissions"

[28] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. USA: John Wiley & Sons, 1994.

[29] E. Altman, *Constrained Markov Decision Processes*, ser. Stochastic modeling. Chapman & Hall/CRC, 1999.

[30] F. J. Beutler and K. W. Ross, “Optimal policies for controlled markov chains with a constraint,” *Journal of Mathematical Analysis and Applications*, vol. 112, no. 1, pp. 236 – 252, 1985.

[31] H. J. Kushner and G. G. Yin, *Stochastic Approximation Algorithms and Applications*. Orlando, FL, USA: New York: Springer-Verlag, 1997.

[32] P. Whittle, “Restless bandits: activity allocation in a changing world,” *Journal of App. Prob.*, vol. 25, pp. 287–298, 1988.

[33] J. Gittins, K. D. Glazebrook, and R. Weber, *Multi-Armed Bandit Allocation Indices*. London: Wiley-Blackwell, 2011.

[34] P. Frenger, S. Parkvall, and E. Dahlman, “Performance comparison of HARQ with chase combining and incremental redundancy for HSDPA,” in *Proc. IEEE Vehicular Tech. Conf.*, vol. 3, 2001, pp. 1829–1833.

[35] P. Auer, T. Jaksch, and R. Ortner, “Near-optimal regret bounds for reinforcement learning,” in *Advances in Neural Information Processing Systems 21*. Curran Associates, Inc., 2009, pp. 89–96.

[36] V. e. a. Mnih, “Human-level control through deep reinforcement learning,” *Nature*, vol. 518, pp. 529–533, Feb. 2015.

[37] P. J. Huber, “Robust estimation of a location parameter,” *The Annals of Math. Statistics*, vol. 35, pp. 73–101, 03 1964.

[38] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *CoRR*, vol. abs/1412.6980, 2014.

[39] S. Boyd and L. Vandenberghe, *Convex Optimization*. NY, USA: Cambridge University Press, 2004.

[40] D. M. Topkis, “Minimizing a submodular function on a lattice,” *Op. Research*, vol. 26, no. 2, pp. 305–321, Apr. 1978.

Elif Tuğçe Ceran received the B.S. and M.S. degrees in Electrical and Electronics Engineering from Middle East Technical University, Ankara, Turkey in 2012 and in 2014 respectively, and Ph.D. degree in electrical and electronics engineering from Imperial College London in 2019. She is currently a postdoctoral researcher at Communication Networks Research Group, Middle East Technical University. Her research interests include online learning, reinforcement learning, energy harvesting wireless networks, resource allocation, and modelling and performance evaluation of communication systems and networks.

Deniz Gündüz [S'03-M'08-SM'13] received the B.S. degree in electrical and electronics engineering from METU, Turkey in 2002, and the M.S. and Ph.D. degrees in electrical engineering from NYU Tandon School of Engineering (formerly Polytechnic University) in 2004 and 2007, respectively. After his PhD, he served as a postdoctoral research associate at Princeton University, and as a consulting assistant professor at Stanford University. From Sep. 2009 until Sep. 2012 he served as a research associate at CTTC in Barcelona, Spain. ln Sep. 2012, he joined the Electrical and Electronic Engineering Department of Imperial College London, UK, where he is currently a Professor of Information Processing. He is also a part-time faculty member at the University of Modena and Reggio Emilia, Italy, and has held visiting positions at University of Padova (2018-2020) and Princeton University (2009-2012).

His research interests lie in the areas of communications and information theory, machine learning, and privacy. Dr. Gündüz serves as an Area Editor for the IEEE Transactions on Communications and the IEEE Journal on Selected Areas in Communications (JSAC), and as an Editor of the IEEE Transactions on Wireless Communications. He is a Distinguished Lecturer for the IEEE Information Theory Society (2020-22). He is the recipient of the IEEE Communications Society - Communication Theory Technical Committee (CTTC) Early Achievement Award in 2017, a Starting Grant of the European Research Council (ERC) in 2016, IEEE Communications Society Best Young Researcher Award for the Europe, Middle East, and Africa Region in 2014, and various best paper awards. He served as a Symposium Co-Chair for the 2020 IEEE International Conference on Communications, and as a General Co-chair of the 2019 London Symposium on Information Theory and 2016 IEEE Information Theory Workshop.

András György received the M.Sc. (Eng.) degree (with distinction) in technical informatics from the Technical University of Budapest, in 1999, the M.Sc. (Eng.) degree in mathematics and engineering from Queen's University, Kingston, ON, Canada, in 2001, and the Ph.D. degree in technical informatics from the Budapest University of Technology and Economics in 2003.

He was a Visiting Research Scholar in the Department of Electrical and Computer Engineering, University of California, San Diego, USA, in the spring of 1998. In 2002-2011 he was with the Computer and Automation Research Institute of the Hungarian Academy of Sciences, where, from 2006, he was a Senior Researcher and Head of the Machine Learning Research Group. In 2003-2004 he was also a NATO Science Fellow in the Department of Mathematics and Statistics, Queen's University. He also held a part-time research position at GusGus Capital Llc., Budapest, Hungary, in 2006-2011. In 2012-2015, he was a researcher in the Department of Computing Science, University of Alberta, Edmonton, AB, Canada. In 2015-2019, he was a Senior Lecturer at the Department of Electrical and Electronic Engineering of Imperial College London, London, UK. Since 2018, he has been a Research Scientist at Deepmind, London, UK.

Dr. György's research interests include machine learning, statistical learning theory, online learning, adaptive systems, optimization, and information theory. He is an Associate Editor of the IEEE Transactions on Pattern Analysis and Machine Intelligence, and regularly serves as a senior program committee member or area chair of leading conferences in machine learning and information theory. He received a Best Paper Award at the 7th IEEE Global Conference on Signal and Information Processing (GlobalSIP 2019), the Gyula Farkas prize of the János Bolyai Mathematical Society in 2001 and the Academic Golden Ring of the President of the Hungarian Republic in 2003.