IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 25 August 2023, accepted 12 October 2023, date of publication 23 October 2023, date of current version 7 November 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3326748

RESEARCH ARTICLE

Reinforcement Learning Based Adaptive Blocklength and MCS for Optimizing Age Violation Probability

AYSENUR OZKAYA[1], AHSEN TOPBAS[2], AND ELIF TUGCE CERAN[2]
[1]Aselsan Inc., 06200 Ankara, Turkey
[2]Department of Electrical and Electronics Engineering, Middle East Technical University, 06800 Ankara, Turkey

Corresponding author: Aysenur Ozkaya (ayozkaya@aselsan.com.tr)

This work was supported in part by Aselsan Inc., and in part by the Middle East Technical University (METU) SRP-Scientific Research Projects (BAP) under Grant AGEP-301-2022-10974.

ABSTRACT As a measure of the freshness of data, Age of Information (AoI) has become an essential performance metric in status update applications with stringent timeliness constraints. This study employs adaptive strategies to minimize the novel, information freshness-based performance metric age violation probability (AVP), the probability of the instantaneous age exceeding a predefined constraint, in short packet communications (SPC). AVP can be considered one of the key performance indicators (KPIs) in 5G Ultra-Reliable Low Latency Communications (URLLC), and it is expected to gain more importance in 6G technologies, especially in extreme URLLC (xURLLC). Two distinct approaches are considered: the first focuses on adaptively selecting the blocklengths with either imperfect or missing channel state information exploiting finite blocklength theory approximations. The second involves dynamically choosing the modulation and coding scheme (MCS) to minimize the AVP under stringent timeliness constraints and non-asymptotic information theory bounds. In the context of adaptive blocklength selection, state-aggregated value iteration, Q-learning algorithms, and finite blocklength theory approximations are leveraged to adjust blocklengths to achieve low age violation probabilities adaptively. The simulation results highlight the effectiveness of these algorithms in minimizing age violation probabilities compared to the fixed blocklengths under varying channel conditions. Additionally, constructing a deep reinforcement learning (DRL) framework, we propose a deep Q-network policy for the dynamic selection of the modulation and coding scheme among the available MCSs defined for URLLC systems. Through comprehensive simulations, we demonstrate the superiority of the proposed adaptive methods over traditional benchmark methods.

INDEX TERMS Age of information, reinforcement learning, dynamic programming, finite blocklength, adaptive modulation and coding.

I. INTRODUCTION

Reliable and fast communication has become an urgent need for many applications with the rapid development of technology over the years. Ranging from factory automation and smart grids to remote surgery and autonomous driving, a vast number of applications rely on reliably and efficiently transmitting short status update packets from a source to a monitor. With these applications came the demand for timely delivery of information. In consequence, a measure of the timeliness of data called *Age of Information (AoI)* has emerged and become an important research topic. AoI is defined as the time elapsed since the last successfully delivered packet was generated [1]. It is a critical metric in status update systems where information is needed before it becomes stale or irrelevant, such as industrial automation,

The associate editor coordinating the review of this manuscript and approving it for publication was Md. Arafatur Rahman.

augmented reality, and traffic safety applications. While it is also regarded as an important metric in fifth-generation (5G) systems, AoI is expected to gain more prominence and be considered as a key performance indicator (KPI) in sixth-generation (6G) communications, especially in next-generation/extreme Ultra-Reliable Low Latency Communication (xURLLC) and massive Machine Type Communication (mMTC) systems. As the name implies, 5G URLLC focuses on stringent latency and reliability requirements; 1 ms or lower latency is targeted in addition to successful packet delivery rates up to $1 - 10^{-5}$ or even $1 - 10^{-9}$ in some cases [2]. With xURLLC, additional qualifications are introduced such as throughput, spectral efficiency, energy efficiency, and security, as well as AoI [3]. The significance of AoI is also apparent in semantic communications, where the meaning of the transmitted message is more important than the accurate transmission of bits [4]. AoI is considered one of the fundamental measures of the relevance of the information in semantic communications, as it determines whether the information is still fresh and valuable or out-of-date and irrelevant [5].

In age-aware xURLLC and mMTC systems, and status update applications such as augmented reality, smart sensors, and industrial automation, information packets generally consist of a small number of bits. Such communication systems are referred to as *short packet communications*. Unlike conventional communication networks with long packets, in short packet communications, the distortions caused by the thermal noise and the propagating channel are not averaged out. Thus, Shannon capacity cannot be used as a performance metric in short packet communications as it is based on infinite blocklength. Instead of classic information theory results, finite blocklength (FBL) theory approximations need to be utilized [6].

The main challenge in age-aware short packet communication systems is the selection of the appropriate blocklength for coding. If a large blocklength is used, implying that a larger number of redundancy bits is used, the probability of error is small. However, the transmission duration increases as a result of transmitting a larger number of bits; hence, age also increases. On the other hand, using a small blocklength results in a shorter transmission time but a higher error probability. Thus, a challenging trade-off exists when selecting the blocklength, and one of our purposes in this study is to overcome this trade-off and minimize the AoI by selecting the blocklength dynamically.

Another approach to the AoI minimization problem for short packet communications is adaptive modulation and coding (AMC). In communication systems, the modulation and coding scheme (MCS) determines the number of bits to be transmitted in one symbol and the coding rate. The selection of the MCS directly affects the age, similar to the blocklength. MCSs with high code rates and modulation order result in short transmission time, but higher error probability. Contrarily, MCSs with lower modulation order and coding rate guarantee a lower error probability, yet longer transmission time. Hence, the same trade-off exists in MCS selection for age optimization.

The majority of the studies on AoI are focused on the *average age* [7], [8], [9], [10], [11], [12] and *peak age* [7], [13], [14]. *Average age* is defined as the time-average AoI. Although useful, it is not a sufficient metric for fully assessing the timeliness of the information since it cannot account for extreme AoI events observed with low probabilities [15]. *Peak age* is another important AoI metric, indicating the value of age just before an update is correctly received. While peak age is a critical metric for ensuring the freshness of the received data, the timeliness of the whole process also needs to be assured. Also, numerous real-time applications have stringent timeliness constraints, and violation probabilities are prominent rather than averages in such systems.

In this study, we investigate the age violation probability (AVP); the probability that the instantaneous age exceeds a given threshold in short packet communications. We first utilize finite blocklength theory approximations to dynamically select the optimal blocklength that optimizes AVP with either imperfect or missing channel state information. Secondly, we focus on choosing the MCS adaptively to minimize the AVP under stringent timeliness constraints and non-asymptotic information theory bounds.

Related Work: There are a few works in the literature showing the existence of an optimal blocklength that minimizes the age-related metrics [7], [8], [9], [13]. In [7], [8], and [9], the optimal blocklength minimizing the average age is investigated taking into account retransmission techniques like automatic repeat request (ARQ) and/or hybrid ARQ (HARQ). On the other hand, in [13], the optimal blocklengths optimizing delay and peak age violation probabilities are studied using FBL information-theoretic bounds. Notably, the study in [13] showed that there may exist two distinct optimal blocklengths that result in same average age but different age violation probabilities. This highlights the critical importance of prioritizing age violation probabilities in addition to the average age while optimizing blocklengths.

Aside from showing the existence of an optimal blocklength, methods for finding the optimal blocklength have also been a topic of discussion [10], [11], [12], [14], [16]. In [10], [11], [14], and [16], blocklength selection in point-to-point wireless networks are considered for optimizing end-to-end delay [16] or age metrics [10], [11], [14]. The study in [12], solves the non-convex blocklength optimization for average age in a two-hop wireless relaying network. References [10] and [16] formulate the average delay [16] and average AoI minimization problems as Markov decision process (MDP) and proposes dynamic blocklength selection methods based on reinforcement learning (RL). Meanwhile, [11] maps the average AoI minimization problem under a power consumption constraint to a constrained Markov decision process (CMDP) and solves the problem by linear programming methods. Although motivated by them, our blocklength selection problem differs from the aforementioned ones as it focuses on the age violation probability and

proposes a dynamic blocklength selection methods based on RL and dynamic programming (DP). This allows our method to adapt to the varying channel conditions and imperfect channel state information, setting it apart from previous work.

Some works in the literature also use RL techniques for AMC to optimize traditional performance metrics such as throughput [17], [18] and spectral efficiency [19]. However, none of them consider dynamic MCS selection in AoI-aware systems. Both [17] and [19], use Q-Learning to map channel quality indicators to MCS options. Reference [19] aims to maximize spectral efficiency and maintain a low block error rate (BLER) while [17] optimizes the link throughput in orthogonal frequency-division multiplexing (OFDM) wireless systems. Reference [18] also maximizes the link-level throughput with MCS selection and power allocation by Deep Deterministic Policy Gradient (DDPG) agents in a distributed manner. MCS selection in age-aware systems has been considered only in [20], where an AoI-driven scheduler without any learning-based approach or any finite blocklength analysis is proposed to minimize the long-term average AoI.

A baseline technique for AMC is outer loop link adaptation (OLLA) [21]. It is an addition to inner loop link adaptation (ILLA), a fixed lookup table method that maps the channel quality indicator (CQI) to the highest MCS that satisfies the block error rate requirement. OLLA improves ILLA by adjusting the signal-to-noise ratio (SNR) according to the positive or negative acknowledgment (ACK/NACK) following a transmission; thus, the effects of delayed CQI or quantization errors are avoided.

To the best of our knowledge, our study is the first to propose an RL-based dynamic MCS selection method to minimize AVP in short packet transmissions and provide superior performance compared to baseline methods. Similarly, while there are some studies on optimal fixed blocklength in age-aware systems, we present a novel method of dynamically selecting the optimal blocklength according to channel conditions based on RL, and we consider not average age but the AVP. Also note that the RL algorithms proposed in this paper do not assume the knowledge of the underlying system characteristics such as channel distribution, packet arrival statistics, and finite blocklength error probabilities.

Objectives and Contributions: Our main objective is to minimize the age violation probability by an adaptive selection of the blocklength or modulation and coding scheme, and the main contributions of this study are as follows:

- We leverage finite blocklength theory approximations and formulate the AVP minimization problem as a discrete-time Markov decision process. We present a dynamic programming method that uses the known system characteristics to select the optimal blocklength for the current channel and AoI states.
- In the absence of apriori knowledge of the system characteristics and with either imperfect or missing channel estimation, we exploit an RL approach for obtaining an online policy that chooses the optimal blocklength adaptively.
- We propose a deep Q-network (DQN) algorithm that dynamically chooses the appropriate MCS among the available MCSs defined in 5G URLLC standards [22]. The adaptive selection of both the codelength and the modulation order is investigated under different scenarios where the channel state information is available or unavailable.
- Extensive simulation results show that the proposed algorithms achieve significantly lower AVP than the fixed blocklength schemes and benchmark link adaptation policies.

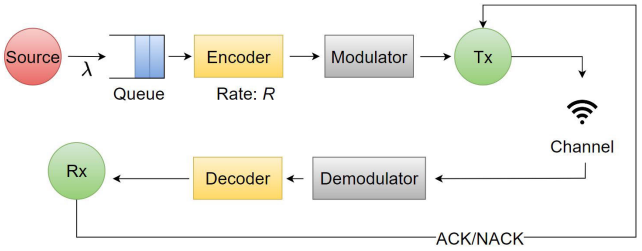

FIGURE 1. System model for the blocklength and MCS selection problems.

The structure of the paper is as follows: In Section II, we present the system model adopted in the blocklength and MCS selection problems. In Section III, we investigate DP and RL-based adaptive blocklength selection methods. In Section IV, we study AVP minimization with dynamic MCS selection and propose a deep RL-based solution. In Section V, we compare our RL-based policies' performances with the baseline methods. Section VI concludes the paper and discusses future work.

II. SYSTEM MODEL

We consider a discrete-time point-to-point communication link with stochastic arrivals of time-critical information packets. The source generates short status update packets according to a Bernoulli distribution, and $\lambda \in (0, 1)$ denotes the probability of a new packet arrival in one channel use (CU). The information packets are stored in a single-server queue with capacity 2, meaning that aside from the packet in service, there can be at most 1 packet in the queue. The queue follows a Last Come First Serve (LCFS) policy with preemption in the queue (LCFS-Q) as defined in [23]: If a new packet arrives when the queue is empty, it is sent to the server immediately. However, if the queue is not empty, the packet already waiting in the queue is replaced with the newly arrived packet. The LCFS-Q queueing policy has previously been shown to be more efficient than the First Come First Serve (FCFS) policy [24].

A. SHORT PACKET TRANSMISSION MODEL

The information packet generated by the source consists of k bits. The encoder maps the information packet to a codeword with blocklength n, and code rate k/n. After encoding and

modulation, the packet is transmitted through the wireless channel. The packet is demodulated and decoded on the receiving side, and a positive or negative acknowledgment is given. Figure 1 illustrates the main components of the system model studied in this paper. We assume a memoryless block-fading channel where the fading coefficient is constant for a block of symbols. Each transmitted packet is subject to independent and identically distributed (IID) fading coefficients and additive white Gaussian noise. The input-output relation of the channel is as follows:

$$y = x \cdot h + w, \tag{1}$$

where x and y denote the transmitted and received symbols, respectively. h is the corresponding fading coefficient and w denotes the additive noise. The fading coefficient h is assumed to be constant during the transmission of a block with length n. Let P denote the transmit power. Assuming additive white Gaussian noise (AWGN) with a standard normal distribution $\mathcal{N}(0, 1)$, instantaneous SNR can be expressed as

$$\gamma = P|h|^2. \tag{2}$$

This paper focuses on transmitting short packets within stringent timeliness constraints. With significantly reduced coding gain, short packet communications are error-prone due to AWGN and fading. The successful reception of a transmission block or a decoding error are assumed to be acknowledged by an error-free single-bit ACK/NACK feedback.

We first study adaptive blocklength selection schemes minimizing (16) and utilize non-asymptotic information theory results in order to derive the BLER for a chosen blocklength n, denoted by ϵ_n. In the well-known study of Polyanskiy et al. [25], the maximal coding rate, i.e., the rate at which an encoder/decoder pair with coded blocklength n and BLER lower than ϵ_n exists, is expressed as follows:

$$R^*(n, \epsilon_n) = C(\gamma) - \sqrt{\frac{V(\gamma)}{n}}\mathrm{Q}^{-1}(\epsilon_n) + \mathcal{O}\Big(\frac{\log n}{n}\Big), \tag{3}$$

where $C(\gamma)$ and $V(\gamma)$, defined as a function of the SNR γ, denote the *capacity* and *channel dispersion*, respectively.

$$C(\gamma) = \log_2(1 + \gamma) \tag{4}$$

$$V(\gamma) = \frac{\gamma(\gamma + 2)}{2(\gamma + 1)^2}\log_2^2(e) \tag{5}$$

Lastly, $\mathcal{O}(\log n/n)$ is the remainder term, and $\mathrm{Q}(\cdot)$ is the tail distribution function of the standard normal distribution:

$$\mathrm{Q}(x) = \frac{1}{\sqrt{2\pi}}\int_x^\infty e^{-t^2/2}dt. \tag{6}$$

Rewriting (3) in the following form allows us to formulate the block error rate ϵ_n given the number of information bits k, coded blocklength n, and SNR γ:

$$\epsilon_n \approx \mathrm{Q}\left(\frac{C(\gamma) - \frac{k}{n}}{\sqrt{\frac{V(\gamma)}{n}}}\right). \tag{7}$$

Then, as a more realistic and practical approach, we consider an MCS selection problem to choose the optimal blocklength and modulation order to minimize AVP in short packet communications. We leverage finite blocklength approximations to obtain BLER, denoted by $\epsilon_{n,M}$, for given blocklength n and modulation order M. In [25], an infinite constellation is assumed; thus, the expression for the maximal coding rate in (3) does not apply to practical modulation schemes with finite constellations such as M-ary quadrature amplitude modulation (M-QAM). In such cases, we can not use the capacity definition in (4). Instead, we can exploit the following mutual information bound in [26].

$$I(\gamma, M) = \log_2 M - \frac{1}{M\pi}\sum_{i=1}^{M}\Bigg[\int e^{-||y-\sqrt{\gamma}x_i||^2} \times \log_2\Bigg(\sum_{k=1}^{M} e^{-||y-\sqrt{\gamma}x_i||^2 - ||y-\sqrt{\gamma}x_k||^2}\Bigg)dy\Bigg] \tag{8}$$

Here, an M-QAM constellation with equiprobable symbols is assumed. γ is the SNR at the receiver, $x_i \in \mathcal{X}_M$ is the M-QAM constellation point from the symbol set $\mathcal{X}_M$, and y is the received signal. In [27], the authors provide the approximation for $I(\gamma, M)$, denoted by $I'(\gamma, M)$, based on multi-exponential decay curve fitting (M-EDCF):

$$I'(\gamma, M) \approx \log_2 M \times \Big(1 - \sum_{j=1}^{k_M}\varepsilon_j^{(M)}e^{-\vartheta_j^{(M)}\gamma}\Big). \tag{9}$$

The coefficients $\varepsilon_j^{(M)}$ and $\vartheta_j^{(M)}$ are provided in [27] and the approximation is shown to be in correspondence with the experimental results. To compute the maximum coding rate in an equiprobable M-QAM constellation, the capacity $C(\gamma)$ in (3) is replaced with $I'(\gamma, M)$ [26], with $V(\gamma)$ and $\mathrm{Q}(.)$ defined the same as in (5) and (6), respectively. Let us denote the block error rate in this case with $\epsilon_{n,M}$, then we can express the maximum coding rate as follows:

$$R^*(n, M, \epsilon_{n,M}) = I'(\gamma, M) - \sqrt{\frac{V(\gamma)}{n}}\mathrm{Q}^{-1}(\epsilon_{n,M}) + \mathcal{O}\Big(\frac{\log n}{n}\Big). \tag{10}$$

We can calculate the BLER by rewriting (10) in the following form:

$$\epsilon_{n,M} \approx \mathrm{Q}\left(\frac{I'(\gamma, M) - \frac{k}{n}}{\sqrt{\frac{V(\gamma)}{n}}}\right). \tag{11}$$

Thus, we use (7) in blocklength selection problem and (11) in MCS selection problem for calculating the block error rate. In addition, we can utilize MCS tables defined in the 5G standards [22], one of the tables lists MCSs with modulation up to 256QAM, and the other two tables define MCSs with 64QAM at most. In this work, we investigate the MCS indexes introduced for low spectral efficiency cases and URLLC applications at [22, Table 5.1.3.1-3]. Table 1

TABLE 1. Summary of MCS indexes [22, Table 5.1.3.1-3].

MCS Index	Modulation order	R x 1024	Spectral Efficiency
0	2	30	0.0586
5	2	99	0.1934
10	2	308	0.6016
15	4	340	1.3281
20	4	616	2.4063
25	6	616	3.6094

outlines some of the MCS indexes with the corresponding modulation orders M, code rates R, and spectral efficiencies. The blocklength used in each MCS, and in (11) for BLER calculation, can be found as in (12).

$$n^{(M)} = \frac{k}{R \cdot \log_2 M} \tag{12}$$

The adaptive MCS selection for AVP optimization can also be considered as adaptive blocklength n and modulation order M selection problem, where the set of available blocklengths is determined using (12).

We consider different scenarios to solve the adaptive block length and MCS selection problems. In the first one, the quantized channel state information (CSIT) is known and included in the state of the system. Channel quality indicator, CQI, stands as a measure of the channel condition depending on the SNR, described as in [19]:

$$CQI = \begin{cases} 0 & \text{if } \gamma \leq \gamma_{\min}; \\ (N_{cqi} - 1) & \text{if } \gamma \geq \gamma_{\max}; \\ \left\lfloor \dfrac{(\gamma - \gamma_{\min})(N_{cqi} - 1)}{\gamma_{\max} - \gamma_{\min}} \right\rfloor & \text{otherwise,} \end{cases} \tag{13}$$

where $\gamma_{\min}$ and $\gamma_{\max}$ are the minimum and maximum SNR values, respectively, and N_{cqi} is the total number of CQI states. $\lfloor . \rfloor$ corresponds to the floor function that takes a real number as input and gives the greatest integer less than or equal to this real number as output. Meanwhile, the second scenario is more practical and studied in this paper, assuming CSIT is unavailable, and CQI is excluded from the state.

B. AGE VIOLATION PROBABILITY (AVP)

Let $\Delta_r(t)$ denote the AoI at the receiver at time $t \in \{0, 1, 2, \ldots\}$, defined as the time elapsed since the generation of the most recent packet that was successfully delivered:

$$\Delta_r(t) = t - u(t), \tag{14}$$

where $u(t)$ is the packet's time stamp, similarly, $\Delta_q(t)$ denotes the AoI at the source queue at time t and represents the time elapsed since the arrival of the last packet in the queue. $\Delta_r(t)$ keeps increasing in the absence of a successful transmission; that is, a transmission error occurs, or there is no status update packet in the system. If a transmission error occurs, the previously transmitted packet is discarded, and the packet waiting in the queue gets transmitted. If a packet is correctly decoded, $\Delta_r(t)$ is set to $\Delta_q(t)$. Figure 2 shows the evolution of $\Delta_r(t)$ over time.

TABLE 2. Notation summary.

Symbol	Description
λ	Packet arrival rate
k	Number of information bits
n	Coded blocklength
γ	Instantaneous SNR
$C(\gamma)$	Capacity
$V(\gamma)$	Channel dispersion
ϵ_n	Block error rate for blocklength n
M	Modulation order
$\epsilon_{n,M}$	Block error rate for blocklength n & modulation order M
$I(\gamma, M)$	Mutual information bound
CQI	Channel Quality Indicator (CQI)
N_{cqi}	Total number of CQI states
$\gamma_{\min}$	Minimum SNR
$\gamma_{\max}$	Maximum SNR
$u(t)$	Time stamp of the most recently received packet
$\Delta_r(t)$	AoI at the receiver at time t
$\Delta_q(t)$	AoI at the queue at time t
$\Delta_{\max}$	Predetermined age threshold

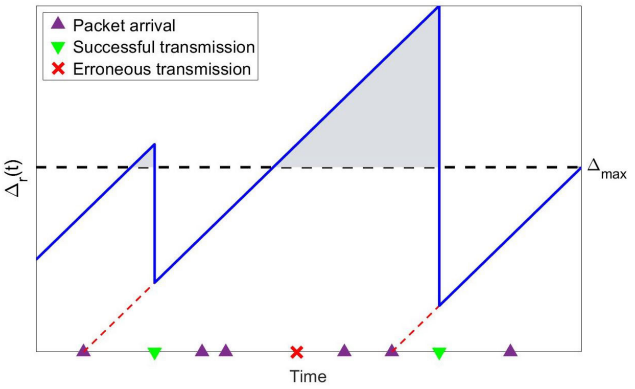

FIGURE 2. The evolution of $\Delta_r(t)$ in the presence of random packet arrivals with LCFS-Q, and transmission errors.

We aim to minimize the age violation probability, defined as the probability that $\Delta_r(t)$ exceeds a predetermined threshold $\Delta_{\max}$. Following the notations in [13] and [28], we can express the AVP as

$$\mathscr{P}_{av}(\Delta_{\max}) = \Pr\{\Delta_r(t) > \Delta_{\max}\}. \tag{15}$$

We consider a frame-based model where the transmitter chooses a finite blocklength n_l (and modulation order M_l for MCS selection) at frames denoted by $l = \{0, 1, 2, \ldots, L\}$. If there is a packet waiting at the source queue at the beginning of frame l, the transmitter transmits the most recent packet selecting a finite blocklength n_l (or modulation order M_l for MCS selection). Otherwise, the transmitter stays idle for one CU, which is assumed to be a frame with length one CU, i.e., $n_l = 1$. Let $t_l \in \mathbb{Z}_{\geq 0}$ and $t_{l+1} \in \mathbb{Z}_{\geq 0}$ denote the starting time of l^{th} frame $(l+1)^{th}$ frames, respectively, where $t_{l+1} = t_l + n_l$.

Using a simplified version of the reward function used in [28] and [29], we count the number of CUs in which the instantaneous age at the receiver exceeds the age threshold, i.e., when $\Delta_r(t) > \Delta_{\max}$, during each frame. We compute

the AVP by taking the ratio of time in which $\Delta_r(t)$ exceeds the threshold to the time passed during the total number of frames L [28]:

$$\mathscr{P}_{av}(\Delta_{\max}) = \lim_{L\to\infty} \frac{1}{L}\mathbb{E}\left[\sum_{l=0}^{L-1}\sum_{t=t_l}^{t_{l+1}-1} \mathbb{1}(\Delta_r(t) > \Delta_{\max})\right], \tag{16}$$

where $\mathbb{1}(\cdot)$ is the indicator function which is equal to 1 if there is an age violation, i.e, $\Delta_r(t) > \Delta_{\max}$; otherwise, it is equal to 0.

III. ADAPTIVE BLOCKLENGTH SELECTION FOR MINIMIZING AGE VIOLATION PROBABILITY

We consider the adaptive selection of coding rate to minimize AVP and address the tradeoff between smaller blocklengths with higher error probability and larger blocklengths with longer transmission delays. To effectively employ RL-based techniques, we formulate our problem as a countable-state discrete-time discounted MDP. This MDP is characterized by five-tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \Gamma\rangle$, where $\Gamma \in (0, 1)$ is the discount factor determining the importance given to future rewards. $\mathcal{S}$ represents the countable state space and is investigated for two different sets $\mathcal{S}^1$ and $\mathcal{S}^2$ corresponding to the scenarios CSIT is available and not, respectively. The first set includes CQI at frame l as a state variable and is formed by three components: $(\Delta_q(l), \Delta_r(l), CQI(l)) \in \mathcal{S}^1$. Meanwhile, the second set does not include CQI and thus $(\Delta_q(l), \Delta_r(l)) \in \mathcal{S}^2$ is formed by two components. With a slight abuse of notation $\Delta_q(l)$, $\Delta_r(l)$ and $CQI(l)$ denote the age of the packet at the queue, at the receiver and quantized channel state at the beginning of frame l, respectively. That is, $\Delta_q(l)$ and $\Delta_r(l)$ represent the AoI at time t_l, indicating that $\Delta_q(l) = \Delta_q(t_l)$ and $\Delta_r(l) = \Delta_r(t_l)$.

The action space, $\mathcal{A}$, represents the finite set of blocklengths we can select, plus *stay idle* action, that is, $n_l = 1$. The reward function $\mathcal{R} : \mathcal{S} \times \mathcal{A} \to \mathbb{Z}$ is defined as:

$$\begin{aligned}
&\mathcal{R}_l \triangleq \mathcal{R}(S_l = (\Delta_q(l), \Delta_r(l)), A_l = n_l)\\
&\mathcal{R}_l = -\sum_{t=t_l}^{t_l+n_l-1} \mathbb{1}(\Delta_r(t) > \Delta_{\max}),\\
&\quad = \begin{cases} -n_l & \text{if } \Delta_r(l) > \Delta_{\max};\\ 0 & \text{if } \Delta_r(l) < \Delta_{\max} - n_l;\\ -n_l + (\Delta_{\max} - \Delta_r(l)) & \text{otherwise,}\end{cases}
\end{aligned} \tag{17}$$

where $\Delta_r(l)$ is the component of S_l describing the AoI at the receiver and $A_l = n_l$ for all $n_l \in \mathcal{A}$. Besides that, we also need to consider the states in which the queue is empty, denoted by $\Delta_q(l) = -1$. There should be no blocklength selection in such states since there are no packets to transmit. The system should stay idle, i.e. $n_l = 1$, until a new packet arrives.

The state transition probabilities $\mathcal{P}_{ss'}^{n_l} = P(S_{l+1} = s'|S_l = s, A_l = n_l)$ is determined by the underlying statistics of error probabilities and random packet arrivals. Therefore, we first recognize all possible state transitions and calculate the following corresponding probabilities.

If the queue state is empty, i.e., $\Delta_q(l) = -1$, the transmitter stays idle for one CU and waits for a new packet arrival, that is, $n_l = 1$. The next queue state, i.e. $\Delta_q(l+1)$, depends on the packet arrival at one CU with probability $\lambda \in (0, 1)$ while $\Delta_r(l+1) = \Delta_r(l) + 1$ as there will not be any new packet arrival to the receiver. The transition probabilities are given as follows (omitting the parenthesis from the state variables (Δ_q, Δ_r)):

$$\begin{aligned}
\mathrm{P}(-1, \Delta_r + 1|\Delta_q, \Delta_r, 1) &= (1-\lambda),\\
\mathrm{P}(0, \Delta_r + 1|\Delta_q, \Delta_r, 1) &= \lambda,
\end{aligned} \tag{18}$$

where Δ_q and Δ_r stand for $\Delta_q(l)$ and $\Delta_r(l)$, respectively. When the queue is not empty at the beginning of frame l, i.e., $\Delta_q(l) \neq -1$, a packet is waiting to be transmitted. Then, the transmitter chooses a finite blocklength n_l from the available blocklengths, $n_l \in \mathcal{A}$. $\Delta_q(l+1)$ depends on the arrival time of the most recent packet in the queue during n_l CUs at frame l. $\Delta_q(l+1) = -1$ refers to the case of no packet arrivals throughout the n_l CUs. For a Bernoulli arrival rate of $\lambda \in (0, 1)$, $\Delta_q(l+1)$ can take the following values with the corresponding probabilities for all $j \in \{0, \ldots, n_l - 1\}$:

$$\Delta_q(l+1) = \begin{cases} -1, & \text{with prob. } (1-\lambda)^{n_l},\\ j, & \text{with prob. } \lambda(1-\lambda)^j.\end{cases} \tag{19}$$

The AoI at the queue in the next frame $l+1$, $\Delta_q(l+1)$, is determined by the arrival time of the most recent packet in the queue during the n_l CUs at previous frame l. The AoI at the receiver in next frame $l+1$, $\Delta_r(l+1)$, depends on the AoI in the queue at the beginning of frame l, $\Delta_q(l)$, and whether a block error occurred or not with probability ϵ_{n_l} defined in (7).

$$\begin{aligned}
\mathrm{P}(-1, \Delta_q + n_l|\Delta_q, \Delta_r, n_l) &= (1-\lambda)^{n_l}(1-\epsilon_{n_l})\\
\mathrm{P}(-1, \Delta_r + n_l|\Delta_q, \Delta_r, n_l) &= (1-\lambda)^{n_l}\epsilon_{n_l}\\
\mathrm{P}(j, \Delta_q + n_l|\Delta_q, \Delta_r, n_l) &= \lambda(1-\lambda)^j(1-\epsilon_{n_l})\\
\mathrm{P}(j, \Delta_r + n_l|\Delta_q, \Delta_r, n_l) &= \lambda(1-\lambda)^j\epsilon_{n_l}
\end{aligned} \tag{20}$$

Unlike $\Delta_q(l)$ and $\Delta_r(l)$, the change in the CQI state is completely independent of other states and the previous CQI state. We calculate the SNR as $\gamma = P|h|^2$ where the channel coefficient h is assumed to be a Rayleigh random variable for simplicity. Since the probability density function of the Rayleigh distribution is known, probabilities corresponding to the defined SNR, hence CQI, intervals can be calculated. In conclusion, using the packet arrival probabilities and state transitions expressed in (18) and (20), and CQI probabilities, we can obtain $\mathcal{P}_{ss'}^{n_l}$ for all states and all actions.

We remark that the formulated MDP has a countable-state space considering both $\Delta_q(l) \in \{0, 1, \ldots\}$ and $\Delta_r(l) \in \{1, 2, \ldots\}$ are unbounded by definition. However, since the reward given (17) is the same for all $\Delta_r(l) > \Delta_{\max}$, the problem can be reduced to a finite-state finite-action MDP where $\Delta_r(l), \Delta_q(l) \in [0, \Delta_{\max} + 1]$. Following the results at [30, Section 10.1.2] and [31, Section 5.1.2], under unichain

policies, Blackwell optimality holds for finite-state finite-action MDPs and the gain of the discounted MDP described in this section approaches to the AVP defined in (16) as discount rate increases, i.e., $\Gamma \to 1$.

We also adopt *state aggregation* method [32] when constructing the state space, i.e., by combining similar states into groups, we reduce the number of states, hence reducing the complexity of the problem. Although the time unit is one CU, $\Delta_q(l)$ and $\Delta_r(l)$ components of the state do not point to a single value, but a collection of values. Hence, the mapping from AoIs at the queue and the receiver to the states $\Delta_q(l)$ and $\Delta_r(l)$ is not one-to-one. With a much lower number of states, the complexities of the proposed algorithms are significantly reduced, and the convergence rate is accelerated.

Next, we present two novel solution methods for the blocklength selection problem. The first is based on the value iteration method [30], [31] exploiting the knowledge of system characteristics, while the second utilizes Q-learning [33] without apriori knowledge of system characteristics.

A. VALUE ITERATION BASED ADAPTIVE BLOCKLENGTH SELECTION

Value iteration is a dynamic programming method that requires full knowledge of the environment dynamics, i.e., state transition probabilities $\mathcal{P}_{ss'}^{n_l}$ in (18), (20) and reward function $\mathcal{R}(S_l, A_l)$ in (17). The purpose of value iteration is to maximize the *state-value function* denoted with $V(S_l)$, which is the expected discounted accumulation of the future rewards starting from the state S_l [30], [31]:

$$V(S_l) = \mathbb{E}\left[\sum_{m\geq 1} \Gamma^{m-1}\mathcal{R}(S_{l+m}, A_{l+m})\right]. \tag{21}$$

It is possible to obtain the optimum state-value function $V^*(s)$ recursively, using the knowledge of $\mathcal{P}_{ss'}^{n_l}$ and $\mathcal{R}_s^{n_l} = \mathcal{R}(S_l = s, A_l = n_l)$:

$$V^*(s) \leftarrow \max_{n_l} \mathcal{R}_s^{n_l} + \Gamma \sum_{s'\in\mathcal{S}^1} \mathcal{P}_{ss'}^{n_l} V(s'). \tag{22}$$

In the value iteration method, we exploit (22) to obtain the maximum state-value function. After the iteration converges, we obtain a deterministic policy denoted by π, where $\pi : \mathcal{S}^1 \to \mathcal{A}$:

$$\pi(s) = \arg\max_{n_l} \sum_{s'\in\mathcal{S}^1} \mathcal{P}_{ss'}^{n_l}[\mathcal{R}_s^{n_l} + \Gamma V(s')]. \tag{23}$$

Value iteration-based adaptive blocklength selection method (VI-ABM) is summarized in Algorithm 1.

Algorithm 1 VI-ABM

$V \leftarrow 0\ \forall s \in \mathcal{S}^1, \rho \ll 1$ /* initialization */
$\delta \leftarrow 1$
repeat
  for all $s = (\Delta_q(l), \Delta_r(l), CQI(l)) \in \mathcal{S}^1$ **do**
    $v \leftarrow V(s)$
    for all $n \in \mathcal{A}$ **do**
      $v(s, n_l) \leftarrow \sum_{s'\in\mathcal{S}} \mathcal{P}_{ss'}^{n_l}\left[\mathcal{R}_s^{n_l} + \Gamma V(s')\right]$
    end for
    $V(s) \leftarrow \max_{n_l} v(s, n_l)$
    $\delta \leftarrow \max(\delta, |v - V(s)|)$
  end for
until $\delta < \rho$ /* convergence */
for all $s = (\Delta_q(l), \Delta_r(l), CQI(l)) \in \mathcal{S}^1$ **do**
  $\pi(s) = \arg\max_{n_l} \sum_{s'} \mathcal{P}_{ss'}^{n_l}[\mathcal{R}_s^{n_l} + \Gamma V(s')]$
end for
return π

B. Q LEARNING BASED ADAPTIVE BLOCKLENGTH SELECTION

We propose two adaptive blocklength selection methods based on Q-learning, which assume no prior knowledge about environmental dynamics. The first Q-learning agent is assumed to know the quantized channel state information, so CQI is included in the state $S_l = (\Delta_q(l), \Delta_r(l), CQI(l)) \in \mathcal{S}^1$ of the system. Also, note that although the CQI knowledge is assumed, the channel state information is noisy and quantized with N_{cqi} as in (13). On the other hand, the second agent knows only the ages of the queue and receiver and assumes no CSIT. Hence, CQI is excluded from the state $S_l = (\Delta_q(l), \Delta_r(l)) \in \mathcal{S}^2$. Actions and rewards are the same for the two scenarios. $\Delta_q(l)$ denotes the age of the packet in the queue, and $\Delta_q(l) = -1$ if the queue is empty. $\Delta_r(l)$ denotes the age of the packet at the receiver.

Q-learning is an online reinforcement learning algorithm to find the optimal action-value function $Q(S_l, A_l)$, also known as *Q-function*. Q-function is the discounted accumulation of the future rewards given state S_l and action A_l:

$$Q(S_l, A_l) = \mathbb{E}\left[\sum_{m\geq 1} \Gamma^{m-1}\mathcal{R}(S_{l+m}, A_{l+m})|A_l\right]. \tag{24}$$

Q-learning is a model-free, off-policy temporal difference algorithm. The Q-learning agent learns entirely by trial and error, following a *behavior policy* that is different from the learned *target policy* to generate behavior [33]. The agent faces a trade-off between exploration and exploitation [34], i.e., choosing the action with the highest action-value estimate or a non-greedy action to improve its estimate. ε-greedy is a simple strategy to balance the exploration-exploitation trade-off: With probability ε, the agent chooses a random action, and with probability $1-\varepsilon$, it chooses a greedy action.

Firstly, we initialize the Q-functions $Q(S_l, A_l)$ to zero for all states $S_l \in \mathcal{S}$ and all actions $A_l \in \mathcal{A}$. We follow an ε-greedy policy with a decaying exploration rate: at each iteration, the exploration rate ε is multiplied by a decay rate ζ. The initial value is $\varepsilon = \varepsilon_{\max}$, and the minimum value is

limited to $\varepsilon_{\min}$. At each iteration, according to the observed state S_l, the agent has to select either to use a blocklength n_l if there is a packet waiting for service or to stay idle for one CU, i.e., $n_l = 1$. After the action is executed, the environment goes to the next state S_{l+1}, and returns reward $\mathcal{R}(S_l, A_l)$ defined in (17). We update the corresponding Q-table entry $Q(S_l, A_l)$ according to Bellman's rule:

$$Q(S_l, A_l) \leftarrow Q(S_l, A_l) + \alpha\big(\mathcal{R}(S_l, A_l) + \Gamma \max_{A_{l+1}} Q(S_{l+1}, A_{l+1}) - Q(S_l, A_l)\big), \quad (25)$$

where α, $0 < \alpha < 1$, is the *learning rate* or *step size*. With a higher learning rate, the changes in $Q(S_l, A_l)$ are more rapid. Similar to the exploration rate, we use a decaying learning rate: starting with $\alpha = \alpha_{\max}$, the learning rate is multiplied with the same decay rate ζ in each iteration, and the minimum value it can take is $\alpha_{\min}$. Assuming that all state-action pairs continue to be updated, and the parameters ε and α are set properly, $Q(S_l, A_l)$ converges to the optimal value $Q^*(s, a) = Q(S_l = s, A_l = a)$ for given frame l [33].

Algorithm 2 gives a detailed explanation of our Q-learning-based adaptive blocklength selection method (QL-ABM).

Algorithm 2 QL-ABM

$Q \leftarrow 0\ \forall s \in \mathcal{S}^{1,2}$ for QL-ABM-1,2 and $\forall a \in \mathcal{A}$
$l = 0$ /* initialize frame counter */
$s = (-1, 100, 0)$ for QL-ABM-1
/* initialize s with an empty queue, destination age equal to k (num. of info. bits), 0 dB SNR */
$s = (-1, 100)$ for QL-ABM-2
for $l = 1, 2, \ldots, L$ **do**
  Observe the current state s:
  $s = (\Delta_q(l), \Delta_r(l), CQI(l))$ for QL-ABM-1
  $s = (\Delta_q(l), \Delta_r(l))$ for Q ABM-2
  if $\Delta_q(l) = -1$ **then**
    $a \leftarrow 1$ /* choose stay idle */
  else
    $a \leftarrow n_l$ according to ε-greedy:
    Explore with probability ε /* choose a randomly */
    Exploit with probability $(1-\varepsilon)$ /* choose a that maximizes $Q(s', a)$ */
  end if
  if new packet arrives **then**
    Update $\Delta_q(l)$
  end if
  Observe the next state s' and reward r:
  $s' = (\Delta_q(l+1), \Delta_r(l+1), CQI(l+1))$ for QL-ABM-1
  $s' = (\Delta_q(l+1), \Delta_r(l+1))$ for QL-ABM-2
  & $r = -\sum_{t=t_l}^{t_l+n_l-1} \mathbb{1}(\Delta_r(t) > \Delta_{\max})$
  Update Q-table:
  $Q(s, a) \leftarrow Q(s, a) + \alpha(r + \Gamma \max_{a'} Q(s', a') - Q(s, a))$
  $s \leftarrow s'$
end for

IV. ADAPTIVE MCS SELECTION FOR MINIMIZING AGE VIOLATION PROBABILITY

In this section, we focus on adaptively selecting the modulation and coding schemes to minimize the age violation probability, and present our solution based on deep Q-networks.

A. DQN BASED ADAPTIVE MCS SELECTION

The modulation and coding scheme selection is a more complex problem than blocklength selection. This is because the number of actions and states is significantly larger, and it is impractical to use a tabular method like Q-learning where Q-functions $Q(S_l, A_l)$ for all states $S_l \in \mathcal{S}$ and actions $A_l \in \mathcal{A}$ are stored in a table. The required memory and computation resources are too high; thus, Q-learning fails to be a feasible solution, and we utilize deep reinforcement learning (DRL) methods instead [34]. It is a function approximation technique that uses deep neural networks (DNN). The Q-function $Q(S_l, A_l)$ is approximated by $Q(S_l, A_l; \theta)$, where θ is the vector consisting of the weights of the DNN mimicking the actual $Q(S_l, A_l)$. The network is also called a *deep Q-network (DQN)*. It consists of an input layer, H hidden layers, and an output layer. The network takes a state S_l as an input, and as outputs, it gives the Q-functions for state S_l and all possible actions.

Similar to Section III-B, we consider two DQN-based scenarios to solve the adaptive MCS selection problem. In the first one, the CQI information is known and included in the state S_l of the system. Meanwhile, the second scenario is more practical, assuming we know only the ages at the queue and receiver, and CQI is excluded from the state. Actions and rewards are the same for the two scenarios. Let $\mathcal{S}^1$ and $\mathcal{S}^2$ denote the state spaces for the first and second scenarios as $(\Delta_q(l), \Delta_r(l), CQI(l)) \in \mathcal{S}^1$ and $(\Delta_q(l), \Delta_r(l)) \in \mathcal{S}^2$, respectively. Similarly to Section III, $\Delta_q(l)$ denotes the age of the packet in the queue, and $\Delta_q(l) = -1$ if the queue is empty. $\Delta_r(l)$ denotes the age of the packet at the receiver. For the CQI state, instead of quantization as in Section III, here we obtain the CQI simply by rounding the SNR to the nearest integer.

Unlike the blocklength selection problem, we do not use the state aggregation method for $\Delta_q(l)$ and $\Delta_r(l)$. The evolutions of $\Delta_q(l)$ and $\Delta_r(l)$ in time are the same: The age of the packet at the queue is affected only by the new packet arrivals to the system. When a packet arrives at the queue, $\Delta_q(l)$ is reset to zero. Otherwise, it increases with the unit rate. The age at the receiver $\Delta_r(l)$, on the other hand, grows until the transmission is completed successfully. Let $n_l^{(M)}$ denote the blocklength used according to the chosen MCS index at frame l, and $n_l^{(M)} = 1$ implies the action of staying idle for one CU. Then, the changes in $\Delta_q(l)$ and $\Delta_r(l)$ after

$n_l^{(M)}$ CUs can be expressed as follows:

$$\Delta_q(l+1) = \begin{cases} -1, & \text{with prob. } (1-\lambda)^{n_l^{(M)}}, \\ j, & \text{with prob. } \lambda(1-\lambda)^j. \end{cases} \tag{26}$$

$$\Delta_r(l+1) = \begin{cases} \Delta_q(l) + n_l^{(M)} & \text{with prob. } (1-\epsilon_{n_l,M_l}) \\ & \text{if } \Delta_q(l) \neq -1, \\ \Delta_r(l) + n_l^{(M)} & \text{otherwise.} \end{cases} \tag{27}$$

Again, the CQI state after $n_l^{(M)}$ CUs does not depend on the previous or the other CQI states but changes randomly according to Rayleigh distribution. The finite action space $\mathcal{A}$ represents the MCSs in [22, Table 5.1.3.1-3], plus *stay idle* action. Also, we design a slightly different reward function $\mathcal{R}(S_l, A_l)$ than the one in Section III. We count the number of age violations in each iteration because of the selected action. However, this is not a sufficient solution: The reward of applying an action A_l is the same whether $\Delta_r(l)$ is above the threshold or not. Thus, the reward should include information about how much the threshold is exceeded. Also, as in blocklength selection problem, the DQN agent should not choose to stay idle unless the queue is empty. Again, rewards corresponding to these cases are large negative values. On the other hand, the reward of choosing to stay idle when the queue is empty is zero, as it is the optimal action to take in that state. We follow a slightly different notation from Section III here, a_0 corresponds to the action of staying idle, i.e., $n_l^{(M)} = 1$. Then, the reward function is expressed in (28), as shown at the bottom of the next page.

The DQN agent iteratively learns with experience. An experience can be represented with a $(S_l, A_l, \mathcal{R}(S_l, A_l), S_{l+1})$ tuple: The state S_l, the action A_l taken in state S_l, the reward $\mathcal{R}(S_l, A_l)$ obtained by taking action A_l in state S_l, and the resulting next state S_{l+1}. A *replay buffer* with a limited size stores the experiences, and to train the network, a batch of experiences is sampled randomly from the buffer. This method improves stability because it eliminates the correlations between the samples and covers a wider variety of state-action pairs [34]. The instabilities are also limited by the usage of two networks in the training process: the *main network* and the *target network*. The main network is represented with the action-value function with weight vector θ ($Q(S_l, A_l; \theta)$), and the target network is shown as $\hat{Q}(S_l, A_l; \theta^-)$. While the main network is actively trained, the target network is updated at every N episodes. The purpose is to improve stability and increase the probability of convergence by avoiding rapid changes in $\hat{Q}(S_l, A_l; \theta^-)$.

At each time step in an episode of the algorithm, the agent chooses an action A_l with an ε-greedy approach: with probability ε, a random action is selected. Otherwise, the action with the maximum Q value is selected. As in QL-ABM, we use a decaying exploration rate ε. Execution of action A_l results in reward $\mathcal{R}(S_l, A_l)$ and state S_{l+1}. The experience $(S_l, A_l, \mathcal{R}(S_l, A_l), S_{l+1})$ is stored in the replay buffer. The agent is trained with a minibatch of experiences sampled randomly from the replay buffer. The difference between the actual and predicted results, i.e., *gradient loss* ($L(\theta)$), is calculated. As the loss function, we use Huber loss [35]:

$$L_\phi(y, f(x)) = \begin{cases} \dfrac{(y-f(x))^2}{2} & \text{if } |y-f(x)| \leq \phi; \\ \delta(|y-f(x)| - \dfrac{\phi}{2}) & \text{otherwise.} \end{cases} \tag{29}$$

(29) states that if the loss value is less than ϕ, Huber loss is equal to the *mean squared error (MSE)*; however, for loss values greater than ϕ, Huber loss equals the *mean absolute error (MAE)*. As MSE loss squares the difference, it puts more weight on *outliers*, i.e., observations that differ substantially from the others. On the other hand, MAE loss weighs all errors with a linear scale, ignoring the outliers. By combining MSE and MAE, Huber loss balances the weight given to outliers.

As the training processes, the loss is expected to converge to arbitrarily small values. Lastly, at every N episodes, the weights of the main network are copied to the target network. The algorithm for our DQN-based adaptive MCS selection method is given in Algorithm 3, and the related parameters are listed in Table 3.

TABLE 3. DQN hyperparameters.

Parameter	Value
Number of layers in the DQN	3
Number of neurons in each layer	32,64,32
Activation function	Rectified Linear Unit (ReLU)
Optimizer	Adam optimizer
Loss function	Huber loss
Number of episodes	5000
Episode length	100
Discount factor (Γ)	0.95
Maximum exploration rate (ε_{max})	0.1
Minimum exploration rate (ε_{min})	0.0001
Decay rate	0.99
Learning rate (α)	0.005
Target network update frequency	10
Replay buffer size	3000
Minibatch size	64

B. BASELINE SOLUTIONS

To evaluate their performances, we compare our DQN-based solutions with two baseline methods: ILLA and OLLA [21]. ILLA is an adaptive MCS selection method based on a fixed lookup table approach; it chooses an MCS index that satisfies a target BLER requirement for a given SNR value. The measured SNR can be unstable because of variations in the wireless channel, quantization errors, and delays. In such cases, ILLA becomes an inefficient solution, and the OLLA technique is used in addition to ILLA for improving performance. OLLA adjusts the measured SNR γ with an offset η_{olla} according to the ACK/NACK feedback about the transmitted packet. The resulting SNR γ_{olla} is used for

Algorithm 3 DQN-AMC

Initialize replay memory
Initialize Q with random weights θ
Initialize $\hat{Q}$ with random weights θ^{-}
for episodes $p = 1, 2, \ldots$ **do**
  Initialize state s:
  $s = (\Delta_q(l), \Delta_r(l), CQI(l))$ for DQN-AMC-1
  $s = (\Delta_q(l), \Delta_r(l))$ for DQN-AMC-2
  for $l = 1, 2, \ldots$ **do**
    Observe the current state s:
    Choose an action $a \in \mathcal{A}$:
    $a \leftarrow I_{MCS}$ according to ε-greedy:
    Explore with probability ε `/* choose a randomly */`
    Exploit with probability $(1-\varepsilon)$ `/* choose a that maximizes` $\hat{Q}(s', a; \theta)$ `*/`
    if new packet arrives **then**
      Update $\Delta_q(l)$
    end if
    Observe the next state s' and reward r:
    if $\Delta_q(l) = -1$ & $a \neq a_0$ **then**
      $r = -5000$
    else if $\Delta_q(l) \neq -1$ & $a = a_0$ **then**
      $r = -5000$
    else
      $r = -\sum_{t=t_l}^{t_l+n_l^{(M)}-1} \mathbb{1}(\Delta_r(t) > \Delta_{\max}) + \max(0, \Delta_r(l) - \Delta_{\max})$
    end if
    Store transition (s, a, r, s') in replay memory
    Sample minibatch of transitions (s_j, a_j, r_j, s_j') from replay memory
    Set $y_j = r_j + \Gamma \max_{a'} \hat{Q}(s_{j+1}, a'; \theta)$
    Calculate $L_\phi(y_j, Q(s, a; \theta))$ `/* the loss */`
    Update $\hat{Q}(s, a; \theta)$ at every N episodes
  end for
end for

selecting the MCS index from the lookup table.

$$\gamma_{olla} = \gamma - \eta_{olla} \tag{30}$$

η_{olla} is updated in each transmission according to the following rule:

$$\eta_{olla} \leftarrow \eta_{olla} + \eta_{up} \cdot \mathbb{1}_{nack} - \eta_{down} \cdot \mathbb{1}_{ack}, \tag{31}$$

where η_{up} and η_{down} are the *step up* and *step down* parameters, related to each other in terms of the target BLER denoted as $BLER_T$:

$$\eta_{down} = \frac{\eta_{up}}{\frac{1}{BLER_T} - 1}. \tag{32}$$

OLLA algorithm is also given in detail in Algorithm 4.

Algorithm 4 OLLA

Input: η_{up}, η_{down}

$\eta_{olla} = 0$ `/* initialize offset to zero*/`
for each transmission **do**
  if ACK **then**
    $\eta_{olla} \leftarrow \eta_{olla} - \eta_{down}$
  else
    $\eta_{olla} \leftarrow \eta_{olla} + \eta_{up}$
  end if
end for
$\gamma_{olla} = \gamma - \eta_{olla}$
$MCS = MCS(\gamma_{olla})$

V. SIMULATION RESULTS

In this section, we demonstrate the performances of our adaptive blocklength and MCS selection methods, and compare them with baseline methods.

A. ADAPTIVE BLOCKLENGTH SELECTION

Before displaying the performance of our value iteration and Q-learning-based adaptive blocklength selection methods, we first show the existence of the optimal blocklength in various scenarios. The number of information bits is $k = 100$, and the blocklengths go from 100 to 300 with a step size of 25. With a fixed number of information bits, different blocklengths n imply different coding rates $R = k/n$. Figure 4 plots the coding rate versus age violation probability for different transmit powers P. The minimum AVP values for each P are shown with red circles. It is clear that the best rate, hence, the best blocklength that minimizes AVP differs as P increases. Similarly, as seen in Figures 5 and 6, changing the packet arrival rate λ or the threshold $\Delta_{\max}$ changes the optimal rate. These figures illustrate the motivation behind our adaptive blocklength scheme: we aim to find and use a dynamic blocklength selection scheme that achieves better performance than the optimal fixed blocklength.

In the simulation results, we refer to the proposed Q-learning-based policies when the information on the CQI state is available and unavailable as *QL-ABM-1* and *QL-ABM-2*, respectively. Then, we demonstrate the performances

$$\mathcal{R}(S_l, A_l) = \begin{cases} -5000, & \Delta_q(l) = -1 \quad \& \quad A_l \neq a_0; \\ -5000, & \Delta_q(l) \neq -1 \quad \& \quad A_l = a_0; \\ -\sum_{t=t_l}^{t_l+n^{(M)}{}_l-1} \mathbb{1}(\Delta_r(t) > \Delta_{\max}) + \max(0, \Delta_r(l) - \Delta_{\max}) & \text{otherwise.} \end{cases} \tag{28}$$

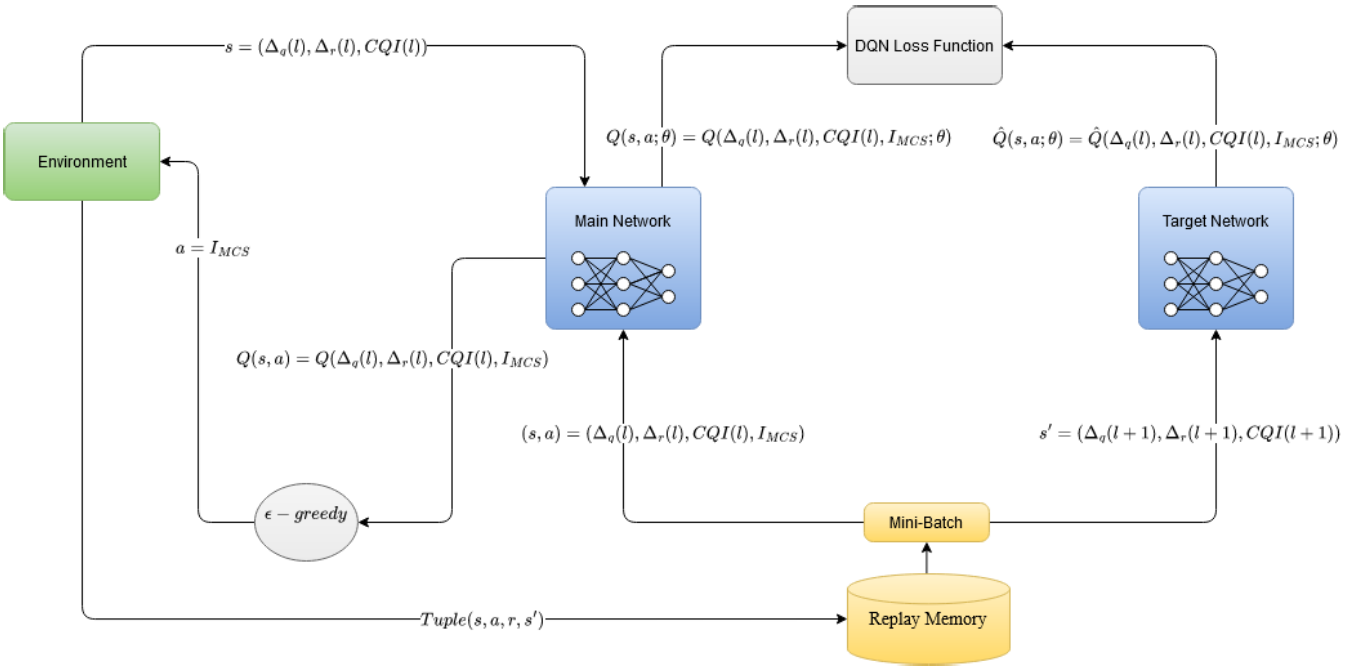

FIGURE 3. Block diagram for DQN-AMC.

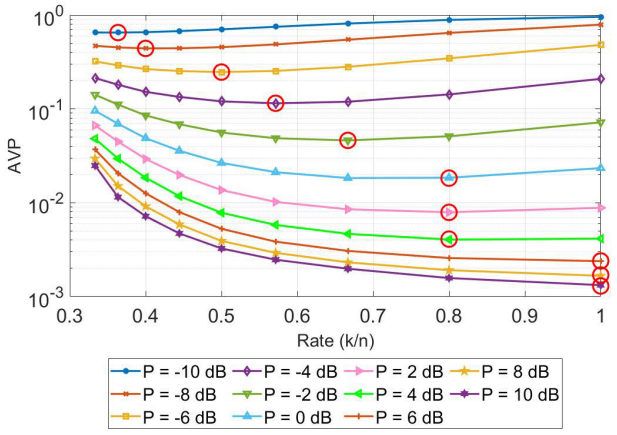

FIGURE 4. Coding rate versus AVP for different transmit power levels when $\lambda = 0.01$ and $\Delta_{max} = 800$ CUs (red circles correspond to the minimum AVPs).

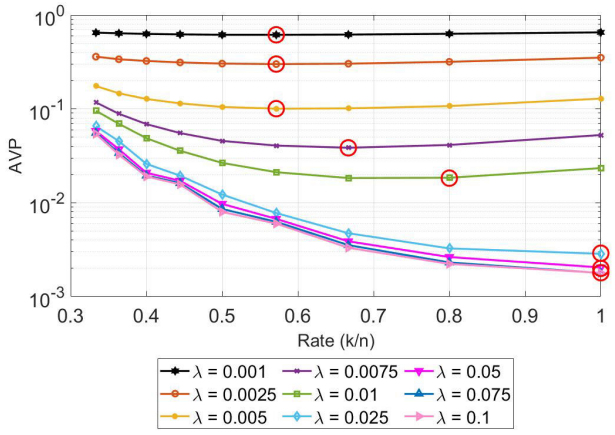

FIGURE 5. Coding rate versus AVP for different arrival rates when $P = 0$ dB and $\Delta_{max} = 800$ CUs (red circles correspond to the minimum AVPs).

of VI-ABM and QL-ABM-1&2 compared with fixed blocklength schemes. We fix the number of information bits to $k = 100$, and the blocklengths in our action space go from 100 to 300 with a step size of 25. In VI-ABM, the number of iterations run for each scenario is 200, and the discount factor Γ is 0.95. The number of iterations and the discount factor Γ in QL-ABM-1&2 are 100000 and 0.95, respectively. As mentioned before, we use a decaying exploration rate ε in QL-ABM-1&2, and the related parameters are $\varepsilon_{\max} = 1$, $\varepsilon_{\min} = 0.01$ and $\zeta = (1 - 10^{-4})$. We also use a decaying learning rate with the same decay rate ζ, and the maximum and minimum values are $\alpha_{\max} = 0.5$ and $\alpha_{\min} = 10^{-4}$.

Figure 7 shows the results obtained with different transmit power levels when the arrival rate and threshold are fixed ($\lambda = 0.01$ and $\Delta_{\max} = 800$ CUs). Low transmit power implies that the probability of experiencing low SNR levels is high. For low P values, large blocklengths ($n \geq 200$) result in lower AVP among all the fixed blocklength schemes. This is because more redundancy bits are needed for reliable transmission, i.e., low BLER, in low SNR cases. As P increases, using large blocklength constantly becomes inefficient, and smaller n values such as 100 and 125 become advantageous. On the other hand, our adaptive blocklength methods provide lower AVP for the majority of P levels since they can dynamically select the optimal blocklength to use in each different channel realization. Although QL-ABM-2 w/o CQI shows slightly worse performance than QL-ABM-1 with CQI, its performance still attains

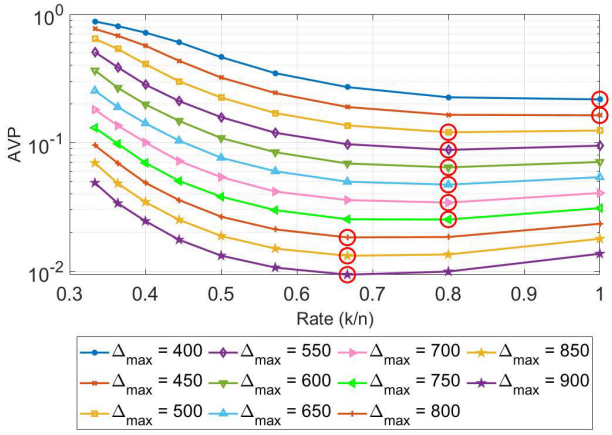

FIGURE 6. Coding rate versus AVP for different age thresholds when $P = 0$ dB and $\lambda = 0.01$ (red circles correspond to the minimum AVPs).

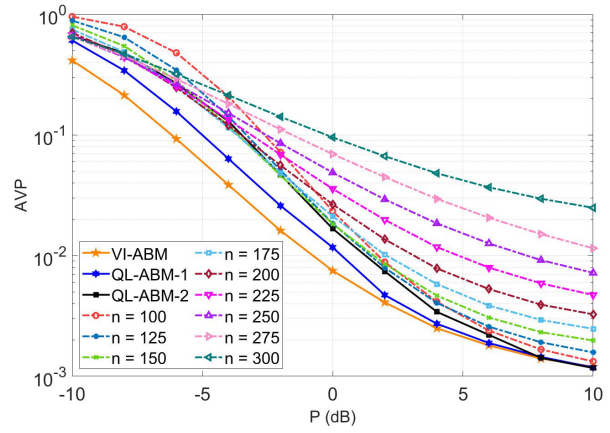

FIGURE 7. Comparison of AVP for VI-ABM, QL-ABM-1 (QL-ABM with CQI state), QL-ABM-2 (QL-ABM without CQI state) and fixed blocklength schemes for different transmit power levels ($\lambda = 0.01$, $\Delta_{max} = 800$ CUs).

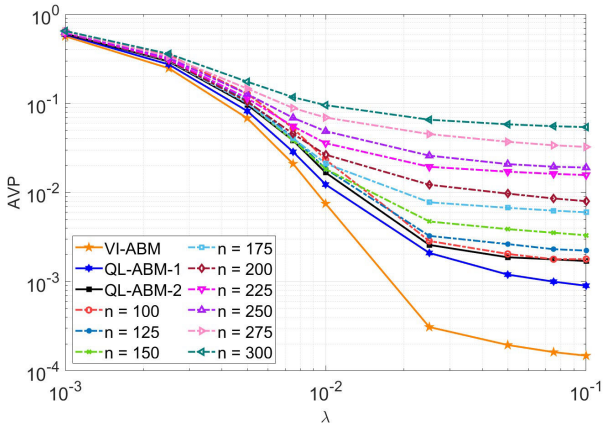

FIGURE 8. Comparison of AVP for VI-ABM, QL-ABM-1 (QL-ABM with CQI state), QL-ABM-2 (QL-ABM without CQI state) and fixed blocklength schemes for different arrival rates ($P = 0$ dB, $\Delta_{max} = 800$ CUs).

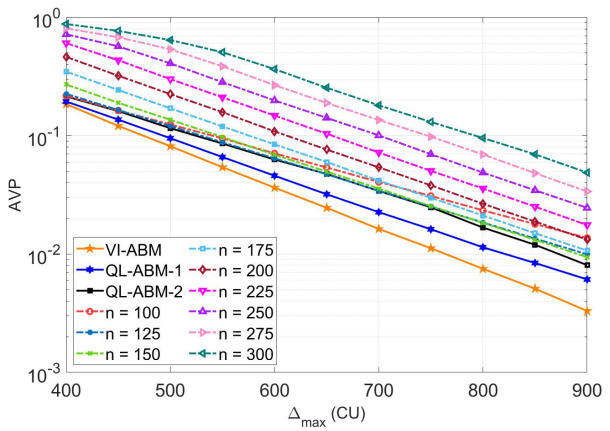

FIGURE 9. Comparison of AVP for VI-ABM, QL-ABM-1 (QL-ABM with CQI state), QL-ABM-2 (QL-ABM without CQI state) and fixed blocklength schemes for different age thresholds ($P = 0$ dB, $\lambda = 0.01$).

or surpasses the performance of best fixed blocklength schemes. The performance differences between VI-ABM, QL-ABM-1, and QL-ABM-2 are more apparent for lower P values. Since high SNR levels are rarely experienced for low transmit powers, the Q-learning agent cannot learn about them thoroughly, so it does not know which action is optimal in the states corresponding to high SNR without CQI knowledge. Meanwhile, VI-ABM and QL-ABM-1 achieve significantly lower AVP for all P values than the other schemes. It is worth noting that VI-ABM requires apriori knowledge of CSIT and system dynamics, which may not always be feasible.

In Figure 8, the results of varying packet arrival rate λ are displayed where $P = 0$ dB and $\Delta_{\max} = 800$ CUs. When λ is small, the packet arrivals are sparse, and the main factor increasing the age is the idle periods where the system waits for new packet arrival. Thus, AVP is very high for both the fixed blocklength schemes and our methods. As λ increases, these idle periods are shortened; hence AVP decreases significantly for all schemes. When $\lambda = 0.1$, the probability of updating the queue with a newly-arrived packet is high, this leads to smaller Δ_q; therefore, smaller Δ_r and AVP. VI-ABM performs better than the fixed blocklength schemes for the whole range of λ values, while the performance gap becomes more visible for larger λ. Although not as good as VI-ABM and QL-ABM-1, QL-ABM-2 also achieves lower AVP than the fixed blocklength schemes for all packet arrival rates.

Lastly, in Figure 9, age violation probabilities for different age thresholds are demonstrated. Transmit power P is kept constant at 0 dB and arrival rate λ is 0.01. For low $\Delta_{\max}$ values, AVP is large for all cases, as expected. As $\Delta_{\max}$ is increased, AVP decreases substantially for all schemes. For all threshold values, VI-ABM and QL-ABM-1&2 outperform the fixed blocklength schemes as the threshold increases, while VI-ABM achieves the lowest age violation probability for all threshold values.

It is clear that for all scenarios, VI-ABM is superior to both QL-ABM-1 and QL-ABM-2. Nevertheless, it is essential to recall that value iteration is a model-based method; hence it requires complete knowledge of the environment dynamics, such as state transition probabilities and reward models. On the other hand, Q-learning agents learn with trial and

error, as it has no prior knowledge about the environment. Also, it suffers from the exploration-exploitation trade-off mentioned in Section 2.6. Thus, it is reasonable that VI-ABM performs better than Q-learning-based methods, considering its prior knowledge and higher complexity. In addition, among two Q-learning-based methods, QL-ABM-1 outperforms QL-ABM-2 for all test scenarios, which is understandable, as SNR, hence CQI state, is a crucial factor in determining the probability of error and affects the action selection process. Nevertheless, QL-ABM-2 is a more practical method than QL-ABM-1 as it does not require knowledge about CSIT.

B. ADAPTIVE MCS SELECTION

We compare the performances of the two DQN-based solutions with the baseline methods ILLA and OLLA. Three target BLER values $(10^{-1}, 10^{-3}, 10^{-5})$ are used with the ILLA method, and for OLLA we set *BLER* to 10^{-1}. The number of information bits is set to $k = 200$. In the MCS table [22, Table 5.1.3.1-3], the modulation order M and the coding rate R for each MCS index are provided and the corresponding blocklength n can be computed as $n = \frac{k}{R \cdot \log_2 M}$. We refer to the proposed policies when the information on the CQI state is available and unavailable as *DQN-AMC-1* and *DQN-AMC-2*, respectively.

Figure 10 shows the age violation probability of different schemes for various transmit power levels P. Age threshold $\Delta_{\max}$ and arrival rate λ are fixed at 5000 CUs and is 0.005, respectively. When P is low, the probability of having lousy channel conditions is higher; thus, the frequently seen SNR values are low, and erroneous transmissions heavily influence AVP. As ILLA and OLLA schemes use low MCS indexes to achieve the target BLER, AVP is high because of the large blocklengths, so the DQN-AMC schemes provide lower AVP. As P increases, the superior performance of DQN-AMC becomes more visible. However, for transmit powers above around 4 dB, ILLA and OLLA schemes become more advantageous as higher MCS indexes with small blocklengths are used. Notably, while the ILLA schemes have similar performances, as BLER of ILLA goes from 10^{-1} to 10^{-5} AVP increases since a lower MCS index with a larger blocklength satisfies the lower BLER requirement at a certain SNR. Meanwhile, it is evident that using OLLA does not significantly affect the age violation probability. Comparing the two DQN-AMC schemes, it can be seen that DQN-AMC-1 clearly outperforms DQN-AMC-2 for most of the P levels. Still, considering that DQN-AMC-2 does not know the SNR and has lower complexity regarding the number of states, it is a feasible solution.

Figure 11 demonstrates the age violation probability for different packet arrival rates. At the lowest arrival rate ($\lambda = 0.001$), DQN-AMC schemes are insufficient. The reason is that the DRL agent mainly encounters the states in which the queue is empty, even with a high exploration rate. Therefore, it cannot fully learn the optimal actions when the queue is non-empty. Increasing λ to about 0.005 leads to a substantial reduction of AVP in all schemes, but the difference is much higher for DQN-AMC schemes. For λ values above 0.005, changes in AVP become negligible for all schemes. As in the previous results, ILLA with BLER = 0.1 and OLLA perform very similarly, and for ILLA with a smaller target BLER, we observe higher AVP.

In Figure 12, AVP is plotted for different age thresholds $\Delta_{\max}$ while the transmit power P is fixed at 0 dB, and arrival rate λ is 0.005. As can be seen, DQN-AMC schemes surpass the performances of ILLA and OLLA schemes. Also, DQN-AMC-1 achieves lower AVP than DQN-AMC-2 for almost all threshold values. Consistent with the previous results, ILLA scheme with BLER $= 10^{-5}$ has the highest AVP, and the difference between the ILLA schemes is visible. Again the OLLA scheme improves the performance negligibly. As the threshold increases, the probability of age violation is reduced for all schemes.

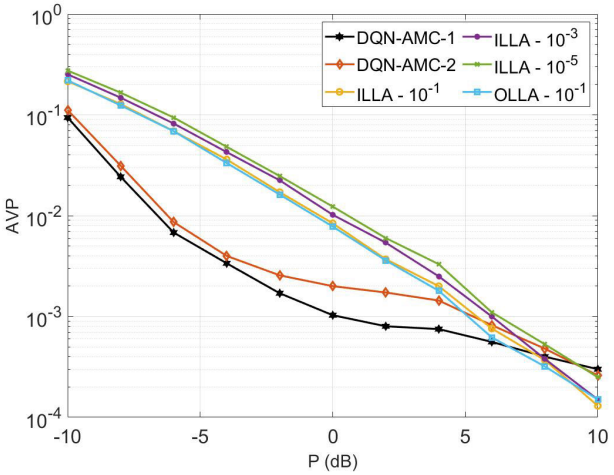

FIGURE 10. Comparison of AVP for DQN-AMC-1 (DQN-AMC with CQI state), DQN-AMC-2 (DQN-AMC without CQI state), ILLA and OLLA methods for different transmit power levels ($\Delta_{\max} = 5000$ CUs, $\lambda = 0.005$).

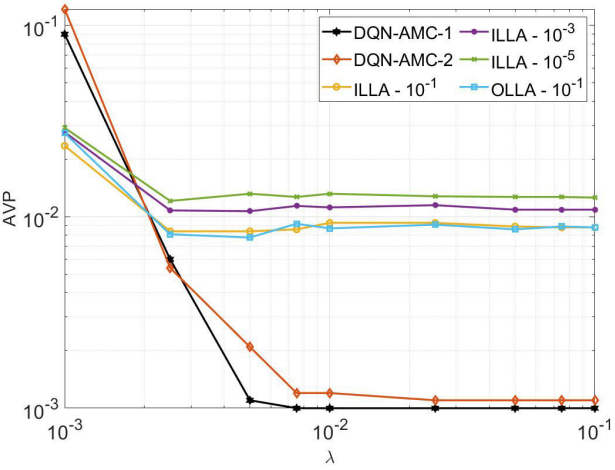

FIGURE 11. Comparison of AVP for DQN-AMC-1 (DQN-AMC with CQI state), DQN-AMC-2 (DQN-AMC without CQI state), ILLA and OLLA methods for different arrival rates ($P = 0$ dB, $\Delta_{\max} = 5000$ CUs).

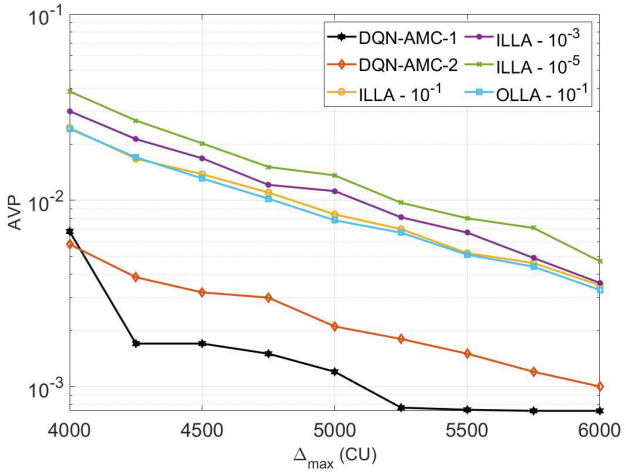

FIGURE 12. Comparison of AVP for DQN-AMC-1 (DQN-AMC with CQI state), DQN-AMC-2 (DQN-AMC without CQI state), ILLA and OLLA methods for different thresholds ($P = 0$ dB, $\lambda = 0.005$).

The proposed DQN-AMC methods achieve lower age violation probabilities for most of the test scenarios. DQN-AMC-1, which includes CQI information in the state performs better than DQN-AMC-2 in general. This is understandable, as SNR, hence CQI, is one of the main factors determining the probability of error and affecting the action selection process. Nevertheless, DQN-AMC-2 is an efficient method considering that it does not require knowledge about the SNR and has a lower number of states, thus lower complexity.

VI. CONCLUSION AND FUTURE WORK

This paper addresses short packet communication links with strict timeliness requirements for xURLLC and mMTC systems. To capture data timeliness, we optimize age violation probability for dynamic blocklength selection and modulation/coding scheme. We propose value iteration and Q-learning under non-asymptotic information theory approximations for dynamic blocklength. Simulation results show that the optimal blocklengths exist for different transmit powers, arrival rates, and predefined age thresholds. The proposed adaptive blocklength selection methods with/without CSIT significantly outperformed the fixed blocklengths even in an unknown arrival rate and block error rate conditions. For the adaptive modulation/coding scheme, due to a large state space, we introduce two algorithms based on DQN, with/without CSIT. Our DQN-based approach exhibits significantly lower age violation probability compared to ILLA and OLLA baseline methods. Across dynamic blocklength and modulation/coding problems, the gap between methods with/without channel state information narrows as SNR increases. These methods have the potential for xURLLC and mMTC systems with multiple users and various channel models considering distinct geographical locations and pathloss of the transmitter and the receiver in the future.

REFERENCES

[1] 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.*, Jun. 2011, pp. 350–358.

[2] M. Bennis, M. Debbah, and H. V. Poor, "Ultrareliable and low-latency wireless communication: Tail, risk, and scale," *Proc. IEEE*, vol. 106, no. 10, pp. 1834–1853, Oct. 2018.

[3] J. Park, S. Samarakoon, H. Shiri, M. K. Abdel-Aziz, T. Nishio, A. Elgabli, and M. Bennis, "Extreme URLLC: Vision, challenges, and key enablers," 2020, *arXiv:2001.09683*.

[4] X. Luo, H.-H. Chen, and Q. Guo, "Semantic communications: Overview, open issues, and future research directions," *IEEE Wireless Commun.*, vol. 29, no. 1, pp. 210–219, Feb. 2022.

[5] E. Uysal, O. Kaya, A. Ephremides, J. Gross, M. Codreanu, P. Popovski, M. Assaad, G. Liva, A. Munari, T. Soleymani, B. Soret, and K. H. Johansson, "Semantic communications in networked systems: A data significance perspective," 2021, *arXiv:2103.05391*.

[6] G. Durisi, T. Koch, and P. Popovski, "Toward massive, ultrareliable, and low-latency wireless communication with short packets," *Proc. IEEE*, vol. 104, no. 9, pp. 1711–1726, Sep. 2016.

[7] H. Sac, T. Bacinoglu, E. Uysal-Biyikoglu, and G. Durisi, "Age-optimal channel coding blocklength for an M/G/1 queue with HARQ," in *Proc. IEEE 19th Int. Workshop Signal Process. Adv. Wireless Commun. (SPAWC)*, Jun. 2018, pp. 1–5.

[8] R. Wang, Y. Gu, H. Chen, Y. Li, and B. Vucetic, "On the age of information of short-packet communications with packet management," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2019, pp. 1–6.

[9] E. Najm, R. Yates, and E. Soljanin, "Status updates through M/G/1/1 queues with HARQ," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jun. 2017, pp. 131–135.

[10] B. Han, Y. Zhu, Z. Jiang, Y. Hu, and H. D. Schotten, "Optimal blocklength allocation towards reduced age of information in wireless sensor networks," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, Dec. 2019, pp. 1–6.

[11] B. Yu, Y. Cai, X. Diao, and K. Cheng, "Adaptive packet length adjustment for minimizing age of information over fading channels," *IEEE Trans. Wireless Commun.*, vol. 22, no. 10, pp. 6641–6653, Oct. 2023.

[12] X. Yuan, Y. Zhu, H. Jiang, Y. Hu, and A. Schmeink, "Data freshness optimization in relaying network operating with finite blocklength codes," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2021, pp. 1–6.

[13] R. Devassy, G. Durisi, G. C. Ferrante, O. Simeone, and E. Uysal-Biyikoglu, "Delay and peak-age violation probability in short-packet transmissions," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jun. 2018, pp. 2471–2475.

[14] A. Özkaya and E. Tugçe Ceran, "Minimizing age violation probability with adaptive blocklength selection in short packet transmissions," in *Proc. 30th Signal Process. Commun. Appl. Conf. (SIU)*, May 2022, pp. 1–4.

[15] L. Hu, Z. Chen, Y. Dong, Y. Jia, L. Liang, and M. Wang, "Status update in IoT networks: Age-of-information violation probability and optimal update rate," *IEEE Internet Things J.*, vol. 8, no. 14, pp. 11329–11344, Jul. 2021.

[16] W. Cheng, Y. Xiao, S. Zhang, and J. Wang, "Adaptive finite blocklength for ultra-low latency in wireless communications," *IEEE Trans. Wireless Commun.*, vol. 21, no. 6, pp. 4450–4463, Jun. 2022.

[17] J. P. Leite, P. H. P. de Carvalho, and R. D. Vieira, "A flexible framework based on reinforcement learning for adaptive modulation and coding in OFDM wireless systems," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Apr. 2012, pp. 809–814.

[18] S. Jamshidiha, V. Pourahmadi, A. Mohammadi, and M. Bennis, "Link-level throughput maximization using deep reinforcement learning," *IEEE Netw. Lett.*, vol. 2, no. 3, pp. 101–105, Sep. 2020.

[19] M. P. Mota, D. C. Araujo, F. H. Costa Neto, A. L. F. de Almeida, and F. R. Cavalcanti, "Adaptive modulation and coding based on reinforcement learning for 5G networks," in *Proc. IEEE Globecom Workshops*, Dec. 2019, pp. 1–6.

[20] C. Li, Y. Huang, Y. Chen, B. Jalaian, Y. T. Hou, and W. Lou, "Kronos: A 5G scheduler for AoI minimization under dynamic channel conditions," in *Proc. IEEE 39th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Jul. 2019, pp. 1466–1475.

[21] K. I. Pedersen, G. Monghal, I. Z. Kovacs, T. E. Kolding, A. Pokhariyal, F. Frederiksen, and P. Mogensen, "Frequency domain scheduling for OFDMA with limited and noisy channel feedback," in *Proc. IEEE 66th Veh. Technol. Conf.*, Oct. 2007, pp. 1792–1796.

[22] *NR; Physical Layer Procedures for Data*, 3GPP, document TS 38.214, Jun. 2022.
[23] R. Devassy, G. Durisi, G. C. Ferrante, O. Simeone, and E. Uysal, "Reliable transmission of short packets through queues and noisy channels under latency and peak-age violation guarantees," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 4, pp. 721–734, Apr. 2019.
[24] M. Costa, M. Codreanu, and A. Ephremides, "On the age of information in status update systems with packet management," *IEEE Trans. Inf. Theory*, vol. 62, no. 4, pp. 1897–1910, Apr. 2016.
[25] Y. Polyanskiy, H. V. Poor, and S. Verdu, "Channel coding rate in the finite blocklength regime," *IEEE Trans. Inf. Theory*, vol. 56, no. 5, pp. 2307–2359, May 2010.
[26] Y. Gao, H. Yang, X. Hong, and L. Chen, "A hybrid scheme of MCS selection and spectrum allocation for URLLC traffic under delay and reliability constraints," *Entropy*, vol. 24, no. 5, p. 727, May 2022. [Online]. Available: https://www.mdpi.com/1099-4300/24/5/727
[27] C. Ouyang, S. Wu, C. Jiang, J. Cheng, and H. Yang, "Approximating ergodic mutual information for mixture gamma fading channels with discrete inputs," *IEEE Commun. Lett.*, vol. 24, no. 4, pp. 734–738, Apr. 2020.
[28] R. D. Yates, Y. Sun, D. Richard 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.
[29] 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," in *Proc. IEEE Symp. Comput. Commun. (ISCC)*, Jun. 2019, pp. 1–6.
[30] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Hoboken, NJ, USA: Wiley, 2014.
[31] R. Bellman, *Dynamic Programming*. Princenton, NJ, USA: Princenton Univ. Press, 1957.
[32] S. Singh, T. Jaakkola, and M. Jordan, "Reinforcement learning with soft state aggregation," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 7, 1994, pp. 1–12.
[33] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 2018.
[34] Y. Li, "Deep reinforcement learning: An overview," 2017, *arXiv:1701.07274*.
[35] P. J. Huber, "Robust estimation of a location parameter," in *Breakthroughs in Statistics*. Cham, Switzerland: Springer, 1992, pp. 492–518.

AYSENUR OZKAYA received the B.S. and M.S. degrees in electrical and electronics engineering from Middle East Technical University (METU), Ankara, Turkey, in 2018 and 2022, respectively. She is currently a Digital Design Engineer with Aselsan (Military Electronic Industries), Ankara. She is a member of the Communication Networks Research Group (CNG), METU. Her research interests include age of information, 5G URLLC systems, reinforcement learning, FPGA design, and signal processing.

AHSEN TOPBAS received the B.S. degree in electrical and electronics engineering from Middle East Technical University (METU), Ankara, Turkey, in 2023, where she is currently pursuing the M.S. degree. Since 2022, she has been a Researcher with the Communication Networks Research Group (CNG), METU. Her research interests include wireless communications, semantic communications, non-terrestrial networks, the Internet of Things, and reinforcement learning.

ELIF TUGCE CERAN received the B.S. and M.S. degrees in electrical and electronics engineering from Middle East Technical University (METU), Ankara, Turkey, in 2012 and 2014, respectively, and the Ph.D. degree in electrical and electronics engineering from the Imperial College London (ICL), in 2019. From 2020 to 2021, she was a Postdoctoral Researcher with the Communication Networks Research Group (CNG), METU. In 2021, she joined the Department of Electrical and Electronics Engineering, METU, where she is currently an Assistant Professor. Her research interests include intersection between machine learning, wireless communications, the Internet of Things, resource allocation, 6G networks, distributed/federated learning, reinforcement learning, performance evaluation, and optimization of computer networks. She has served as the Publicity Chair for the IEEE INFOCOM AoI Workshop 2021, the TPC Chair for WiOpt MOSC Workshop 2023, and the TPC Member for various conferences, including IEEE WCNC, IEEE ICC, IEEE SIU, and IEEE INFOCOM.