REINFORCEMENT LEARNING BASED ADAPTIVE BLOCKLENGTH AND MCS SELECTION FOR MINIMIZATION OF AGE VIOLATION PROBABILITY

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

AYŞENUR ÖZKAYA

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

AUGUST 2022

Approval of the thesis:

REINFORCEMENT LEARNING BASED ADAPTIVE BLOCKLENGTH AND MCS SELECTION FOR MINIMIZATION OF AGE VIOLATION PROBABILITY

submitted by **AYŞENUR ÖZKAYA** in partial fulfillment of the requirements for the degree of **Master of Science in Electrical and Electronics Engineering Department, Middle East Technical University** by,

Prof. Dr. Halil Kalıpçılar
Dean, Graduate School of **Natural and Applied Sciences** _______________

Prof. Dr. İlkay Ulusoy
Head of Department, **Electrical and Electronics Engineering** _______________

Assist. Prof. Dr. Elif Tuğçe Ceran Arslan
Supervisor, **Electrical and Electronics Engineering, METU** _______________

Examining Committee Members:

Prof. Dr. Elif Uysal
Electrical and Electronics Engineering, METU _______________

Assist. Prof. Dr. Elif Tuğçe Ceran Arslan
Electrical and Electronics Engineering, METU _______________

Assist. Prof. Dr. Gökhan Muzaffer Güvensen
Electrical and Electronics Engineering, METU _______________

Assist. Prof. Dr. Serkan Sarıtaş
Electrical and Electronics Engineering, METU _______________

Prof. Dr. Tolga Girici
Electrical and Electronics Engineering, TOBB ETU _______________

Date: 26.08.2022

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Surname:	Ayşenur Özkaya
Signature :	

ABSTRACT

REINFORCEMENT LEARNING BASED ADAPTIVE BLOCKLENGTH AND MCS SELECTION FOR MINIMIZATION OF AGE VIOLATION PROBABILITY

Özkaya, Ayşenur
M.S., Department of Electrical and Electronics Engineering
Supervisor: Assist. Prof. Dr. Elif Tuğçe Ceran Arslan

August 2022, 72 pages

As a measure of data freshness, Age of Information (AoI) is an important semantic performance metric in systems where small status update packets need to be delivered to a monitor in a timely manner. This study aims to minimize the age violation probability (AVP), which is defined as the probability that instantaneous age exceeds a certain threshold. The AVP can be considered as one of the key performance indicators in emerging 5G and beyond technologies such as massive machine-to-machine communications (mMTC) and ultra-reliable low latency communications (URLLC). This thesis focuses on two main problems regarding the adaptive transmission of short packets in time-sensitive systems. Firstly, we propose two methods for choosing the optimal blocklength for coding in short packet transmissions. We utilize finite blocklength theory approximations along with dynamic programming (DP) and reinforcement learning (RL) methods. Adopting state-aggregated value iteration and Q-learning algorithms, we present adaptive policies that dynamically select the optimal blocklength according to the state of the system. Our second problem focuses on choosing the appropriate modulation and coding scheme (MCS) for minimization of age violation probability. We construct a deep reinforcement learning (DRL) framework and employ deep Q networks (DQN) to exploit a policy for the dynamic selection of MCS among available MCSs defined in 5G standards. The performances of the proposed approaches are demonstrated in different scenarios and compared with the performances of benchmark policies and state-of-the-art algorithms.

Keywords: age of information, dynamic programming, reinforcement learning, adaptive modulation and coding, finite blocklength

ÖZ

BİLGİ YAŞI İHLALİ OLASILIĞININ AZALTILMASI İÇİN PEKİŞTİRMELİ ÖĞRENMEYE DAYALI ADAPTİF BLOK UZUNLUĞU VE MCS SEÇİMİ

Özkaya, Ayşenur
Yüksek Lisans, Elektrik ve Elektronik Mühendisliği Bölümü
Tez Yöneticisi: Dr. Öğr. Üyesi. Elif Tuğçe Ceran Arslan

Ağustos 2022 , 72 sayfa

Bilginin tazeliğinin bir ölçüsü olan Bilgi Yaşı (BY), küçük boyuttaki durum güncelleme paketlerinin eskimeden bir gözlemciye iletilmesini gerektiren sistemlerde önemli bir ölçüt haline gelmiştir. Bu çalışmadaki amaç, anlık bilgi yaşının belirli bir eşik değerini aşması olasılığı olarak tanımlanan bilgi yaşı ihlali olasılığının (AVP) en aza indirilmesidir. AVP, kitlesel makine tipi haberleşme (mMTC) ve ultra güvenilir ve düşük gecikmeli iletişim (URLLC) gibi 5G ve ötesi sistemlerinde anahtar performans göstergelerinden biri olarak görülmektedir. Bu tez çalışmasında, kısa paketlerin adaptif iletimini konu alan iki ana problem üzerine odaklanılmıştır. İlk olarak, kısa paket iletimlerinde kodlama için ideal blok uzunluğunu seçmek amacıyla iki yöntem önerilmiştir. Sonlu blok uzunluğu teorisi yaklaşımlarının yanı sıra dinamik programlama (DP) ve pekiştirmeli öğrenme (RL) metotlarından yararlanılmıştır. Durum toplamalı değer iterasyonu ve Q-öğrenme algoritmaları eğitilerek sistemin anlık durumuna göre optimal blok uzunluğunu gösteren adaptif politikalar elde edilmiştir. İkinci problemde bilgi yaşı ihlali olasılığını en aza indirecek modülasyon ve kodlama şemasının (MCS) seçimine odaklanılmıştır. Derin pekiştirmeli öğrenme (DRL) ortamı kurulmuş ve derin Q ağları (DQN) kullanılarak 5G standartlarında tanımlı MCS'ler arasından seçim yapan bir politika elde edilmiştir. Önerilen çözümlerin performansı referans politikalar ve en gelişmiş algoritmalar ile karşılaştırılmış ve farklı senaryolar için elde edilen sonuçlar sunulmuştur.

Anahtar Kelimeler: bilgi yaşı, dinamik programlama, pekiştirmeli öğrenme, adaptif modülasyon ve kodlama, sonlu blok uzunluğu

To my beloved family

ACKNOWLEDGMENTS

I would like to thank my advisor Assist. Prof. Elif Tuğçe Ceran for her great supervision, support, kindness, and understanding throughout this study. Her guidance and advice are very valuable to me, and I am glad I got the opportunity to work with her.

I am grateful to my parents and my sister for their love and support.

I would like to thank Erman for helping me in every possible way, for motivating me, and for his love and patience.

Last but not least, I am thankful to my cats Cacık and Kekik for keeping me company and giving me strength in the most stressful times.

TABLE OF CONTENTS

CHAPTERS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

5G	fifth generation
6G	sixth generation
AoI	Age of Information
AMC	adaptive modulation and coding
AVP	age violation probability
BLER	block error rate
CQI	channel quality indicator
CSI	channel state information
CU	channel use
DP	dynamic programming
DQN	deep Q network
DQN-AMC	DQN based adaptive modulation and coding
DRL	deep reinforcement learning
FBL	finite blocklength
ILLA	inner loop link adaptation
M-QAM	M-ary quadrature amplitude modulation
MCS	modulation and coding scheme
MDP	Markov decision process
NN	neural network
OLLA	outer loop link adaptation
QL-ABM	Q-learning based adaptive blocklength selection method
RL	reinforcement learning
SNR	signal-to-noise ratio
URLLC	Ultra Reliable Low Latency Communications

VI	value iteration
VI-ABM	value iteration based adaptive blocklength selection method

LIST OF SYMBOLS

$\Delta_r(t)$	AoI at the receiver
$\Delta_q(t)$	AoI at the queue
P_{av}	age violation probability
k	number of information bits
n	number of coded bits
λ	packet arrival rate
P	transmit power
Δ_{max}	age threshold
ϵ	block error rate
γ	signal-to-noise ratio
h	fading coefficient
R	coding rate
C	channel capacity
V	channel dispersion
$\mathcal{S}$	state space
$\mathcal{A}$	action space
$\mathcal{R}$	reward function
$\mathcal{P}$	state transition probability matrix
$v(s)$	state-value function
$Q(s, a)$	action-value function
$\pi(s)$	policy
Γ	discount factor
α	learning rate
ε	exploration rate

CHAPTER 1

INTRODUCTION

With the immense progress of technology over the years, the demand for fast and reliable communication networks increased significantly. A wide range of applications has been developed that rely on the transmission of status update messages from a source to a monitor: remote surgery systems, smart cities, wearable devices, etc. These applications brought the need for timely delivery of information [2]. As a result, *Age of Information (AoI)* [3,4] has become an important research topic in recent years as a measure of the freshness of data. AoI is defined as the time elapsed since the last successfully received packet was generated. It is one of the key performance indicators in status update systems where information is needed before it becomes stale or irrelevant. AoI is a critical metric in applications such as autonomous driving, factory automation, and smart grids, along with fifth-generation (5G) technologies such as massive machine-to-machine communications (mMTC) and [5] and Ultra Reliable Low Latency Communications (URLLC) [6]. Especially with the introduction of extreme URLLC (xURLLC) [7] in sixth generation (6G) communications, AoI-related metrics are expected to gain more importance. 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 [8]. AoI is considered as one of the 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 [9,10].

AoI is a destination-centric metric: it is concerned only with the successfully delivered packets of fresh updates [11], making it a substantially different metric from *delay* or *latency*. Also, age and delay vary in their responses to different network parameters. For example, consider a First Come First Served (FCFS) queue in a status update system. If status update frequency is low, then the queuing delay of the packets would be small, thus leading to a smaller end-to-end delay. However, the age at the receiver would increase since the information becomes stale before a new update arrives. On the other hand, high-frequency updates would cause longer queuing delays and higher age because of the waiting time in the queue [12].

The most commonly used age-related metric is the *average age*, i.e., the time-average AoI. Although it is a key performance indicator in status update systems, it is not enough to fully represent the timeliness of the information. In some applications such as URLLC, the *age violation probability*, i.e., the probability that instantaneous age exceeds a threshold, is required to be kept low. In this thesis, we focus on minimizing the age violation probability.

In age-aware status update systems such as autonomous driving, augmented reality and factory automation, information packets generally have a small number of bits; hence the term *short packet communications* is used to describe such communication systems. Short packet communications have certain disadvantages. In conventional communication networks where long packets are transmitted, the distortions caused by the propagation channel and the thermal noise are averaged out. With short packets, this is not possible. Thus, in short packet communication, Shannon capacity [13], which is based on infinite blocklength, cannot be used as a performance metric. The results of classic information theory do not apply to short packet transmissions [14]. Instead, finite blocklength theory approximations need to be considered [15].

The main problem in short packet transmissions with AoI in consideration is the selection of the finite blocklength. A large blocklength implies that more redundancy bits are added to information packets; thus, the error probability is small. However, a large blocklength directly increases the transmission time of the packet and the age at the receiver even if the transmission is successful. On the other hand, a small blocklength results in a short transmission time, with the disadvantage of a higher error probability which will increase the age at the receiver. So, there exists a trade-off in the selection of the blocklength. One of our purposes in this thesis is to select the blocklength dynamically so that the age violation probability is minimized.

Adaptive modulation and coding (AMC) is another approach to the age optimization problem. In communication systems, the number of bits that will be transmitted in one symbol is defined by the modulation and coding scheme (MCS); thus, it directly affects the age. While MCSs with higher modulation order and coding rate shorten the transmission time of a packet, the probability of an erroneous transmission is higher. On the other hand, MCSs with lower modulation and coding rates yield lower error probability but longer transmission time. Hence, similar to blocklength selection, MCS selection also faces a trade-off regarding age optimization. In this study, we aim to overcome this trade-off by selecting the MCS dynamically in order to minimize the age violation probability.

For the adaptive selection of the blocklength and MCS, we utilize reinforcement learning and dynamic programming methods. *Reinforcement learning (RL)* is one of the three types of machine learning, along with *supervised learning* and *unsupervised learning*, and it has been widely used for optimization problems. Reinforcement learning is based on the interaction between a decision-maker called the *agent* and an *environment*, and its aim is to maximize an accumulated reward. The agent does not know about the environment dynamics or the possible consequences of its actions, and it learns entirely by trial and error. *Dynamic programming (DP)*, on the other hand, requires an exact mathematical model of the environment. In dynamic programming, complex problems are broken down into simple subproblems, and these subproblems are solved in a recursive manner. Thus DP methods are generally used in cases with optimal substructures and overlapping subproblems.

For the blocklength selection problem, RL and DP methods are sufficient. However, the MCS selection problem requires utilizing more complex learning methods because it presents a more complicated problem with large numbers of states and actions. Hence, we use *deep reinforcement learning (DRL)*, which is the combination of reinforcement learning with deep learning. DRL methods make use of *neural networks (NN)* to solve reinforcement learning problems.

1.1 Related Work

One of our main goals in this study is to find the optimal blocklength that minimizes the probability of age violation. Our motivation comes from the numerous works in the literature showing the existence of an optimal blocklength that minimizes the age-related metrics. In [16], the average age is analyzed for three different packet management schemes, and it is shown that the average age is minimized with an optimal blocklength. In [17], the average age in M/G/1/1 queues with incremental redundancy (IR) hybrid automatic repeat request (HARQ) and fixed redundancy (FR) HARQ is analyzed. It is shown that for both policies, average age is minimized by an optimal blocklength. [18] also works on IR-HARQ scheme along with simple automatic repeat request (ARQ). The average age in the two schemes is formulated, and optimal blocklength for minimum average age is found. In [19], single transmission and HARQ schemes are compared in terms of average age, and for both schemes, the average age is minimized at certain blocklengths. Results of [20] display the ideal blocklength for minimum average age in a URLLC system with a decode-and-forward relay scheme. In [21], age and energy trade-off in a dual-hop status update system is discussed, and it is found that an optimal blocklength minimizes the average age but maximizes the energy cost. In [22], average age is analyzed in a dual-queue system in which short information packets flow in parallel paths. While it is shown that the dual-queue outperforms the single queue in terms of average age; in both systems, average age is minimized with an ideal blocklength.

There are also many studies that focus on age-related metrics other than the average age. In [23], an M/G/1 queue with ARQ and HARQ schemes is considered. It is shown that there is an optimal blocklength minimizing both the peak age and the average age. [24] analyzes delay violation and peak age violation probabilities in steady state, adopting frame-synchronous and frame-asynchronous system models. Results show that optimal blocklength exists for both minimum delay violation and maximum throughput. In [25], an age-aware machine type communications (MTC) system is considered. Age violation probability and average age are investigated, and it is shown that the optimal blocklengths that minimize the two metrics may differ. [26] considers a downlink cellular network where base stations (BS) transmit packets to a user with finite blocklength, and derives the age violation probability and the average age. As in [25], it is shown that the two metrics are minimized with different optimal blocklengths.

Since it exists as confirmed by many studies, some of which we mentioned above, optimizing the blocklength has been a topic of discussion. In [27], an adaptive blocklength selection scheme is proposed for minimizing end-to-end delay minimization. A variable transmission time interval (V-TTI) approach is adapted with a dynamic buffering model, and a blocklength optimization method is proposed. Also, a resource allocation scheme based on multiple DQNs is suggested, used for adaptive allocation of bandwidth and TTI for multiple users. In [28], the correlation between age and delay is investigated analytically in finite blocklength regime, and a method for joint optimization of age and delay is proposed based on optimal blocklength and update rate selection. In [29], the AoI of wireless sensor networks is studied in the FBL regime. To minimize the long-term discounted AoI of the system, an adaptive blocklength allocation scheme based on Q-learning is proposed. The same authors propose a recursive optimization method for mitigating AoI outage, i.e., AoI being critically high, in [30]. AoI outage probability in steady-state is analyzed, and a recursive policy optimizer is presented. In [31], the average age of a two-hop relay working with a decode-and-forward rule is investigated. An iterative algorithm is proposed for joint optimization of blocklengths allocated to both hops to minimize the average age. Our study differs from the ones mentioned above as it focuses on the age violation probability and proposes a dynamic blocklength selection method based on reinforcement learning and dynamic programming that adapts to the varying channel conditions.

Although reinforcement learning is commonly used in age-related problems, the majority of the studies are on scheduling and resource allocation [32–37]. Age optimization or minimization with RL methods is also popular in many energy harvesting [38–41] and UAV trajectory planning [42–45] applications.

There are also some work in the literature that use reinforcement learning techniques for adaptive modulation and coding in order to optimize traditional performance metrics. In [46], Q-learning is used for adaptively selecting the MCS in a 5G framework.

A mapping between the channel conditions and suitable MCSs is obtained to maximize the spectral efficiency while a low block error rate (BLER) is maintained. [47] proposes an algorithm for spectral efficiency maximization in orthogonal frequency-division multiplexing (OFDM) wireless systems. The RL-based algorithm learns to select the best MCS according to the signal-to-noise ratio. In [48], goodput is maximized by using an actor-critic method for link adaptation, optimizing both the MCS in the physical layer (PHY) and frame size in the medium access control (MAC) layer in an IEEE 802.11n framework. [49] proposes an RL-based dynamic radio resource allocation for meeting key performance indicator (KPI) requirements in heterogeneous virtual radio access networks using differential semi-gradient State-Action-Reward-State-Action (SARSA) algorithm. In [50], multiple Deep Deterministic Policy Gradient agents are used for MCS selection and power allocation in order to maximize link-level throughput. [51] tackles the outdated channel state information (CSI) problem and proposes an adaptive modulation method based on deep reinforcement learning. [52] addresses MCS selection for maximizing the throughput in an Internet of Vehicles (IoV) framework. An RL approach is utilized for selecting the appropriate MCSs for each vehicle. In [53], again, the aim is to maximize the throughput, but in a 5G mobile network. Q-learning is used with a neural network to obtain an adaptive MCS and transmission rank selection method. MCS selection in age-aware systems has been considered only in [54], where an AoI-driven scheduler compliant with 5G standards is proposed to minimize the average age.

Outer loop link adaptation (OLLA) [55] is a baseline technique applied for adaptive modulation and coding. It is an addition to inner loop link adaptation (ILLA), which is a fixed lookup table method, mapping the CQI to the highest MCS that satisfies the block error rate requirement. OLLA improves ILLA by adjusting the SNR according to the positive or negative acknowledgment following a transmission; thus, the effects of delayed CQI or quantization errors are avoided. Although quite functional, OLLA has drawbacks such as slow or no convergence, and it can not fully adapt to non-stationary channel conditions. [56] proposes a dynamic OLLA algorithm that adapts to the variability of the SNR, hence improving the robustness of traditional OLLA. [57] suggests adjusting the OLLA parameters according to the convergence status.

To the best of our knowledge, our study is the first to propose a reinforcement learning based dynamic MCS selection method to minimize the age violation probability and provide superior performance compared to baseline methods. Similarly, while there are numerous studies on optimal blocklength in age-aware systems, some of which we have mentioned previously, we present a novel method of dynamically selecting the optimal blocklength according to channel conditions based on reinforcement learning, and we consider not average age but the age violation probability.

1.2 Objectives, Contributions, and Thesis Structure

Our main objective is to minimize the age violation probability by adaptive selection of the blocklength or modulation and coding scheme (MCS), and the main contributions of this thesis are as follows:

- We propose two adaptive blocklength selection methods for minimizing the age violation probability. First, we utilize a dynamic programming method that uses the known system characteristics to select the appropriate blocklength for the current channel conditions. Secondly, we propose a reinforcement learning algorithm for obtaining an adaptive policy that chooses the optimal blocklength without actually knowing the system characteristics.
- We propose a deep reinforcement learning (DRL) approach to the MCS selection problem for the minimization of age violation probability. Using deep Q networks, we adopt a DRL policy that dynamically selects the appropriate MCS among the available MCSs defined in 5G standards to minimize the age violation probability.

The structure of the thesis is as follows: In Chapter 2, we present background information on Age of Information (AoI) and finite blocklength theory. We provide an overview of reinforcement learning (RL) and go over the dynamic programming (DP) and RL methods used in the proposed solutions. In Chapter 3, we present the blocklength selection problem with a detailed system model. We explain our DP and RL based approaches and demonstrate their performances in comparison to fixed blocklength schemes. In Chapter 4, we study age violation probability minimization with

MCS selection. We describe our deep RL based solution and compare the performance of our solution with the baseline methods, namely, ILLA and OLLA. Lastly, in Chapter 5, we summarize the thesis.

1.3 Scientific Contributions

Part of Chapter 3 was presented at the 30^{th} IEEE Conference on Signal Processing and Communication Applications (SIU 2022) and has received the IEEE Best Student Paper Award [58].

CHAPTER 2

CONCEPTUAL FRAMEWORK

This chapter provides background information on subjects discussed in further chapters. Firstly, we explain the AoI and finite blocklength concepts. Then, we present an overview of Markov decision processes (MDP), dynamic programming, reinforcement learning (RL) and deep RL methods used in this thesis.

2.1 Age of Information

Age of Information (AoI) is a metric that characterizes the *timeliness* or *freshness* of a monitor's knowledge about a process or an entity [2] and is an important performance indicator in status update applications. AoI is expressed as [4]

$$\Delta(t) = t - u(t), \tag{2.1}$$

where $u(t)$ is the time stamp, i.e., the generation time of the last update the monitor has successfully received. A general model for a status updating system is depicted in Figure 2.1. The source generates packets to be transmitted to the monitor through the network. We can observe two AoI processes here: $\Delta_1(t)$ represents the age of the packet at the source side, whereas $\Delta(t)$ is the age at the monitor side.

Figure 2.2 shows new packets arriving at the system at times $t_1, t_2, ...$ At each t_i, $\Delta_1(t)$ is reset to zero because of fresh updates. If there is no incoming fresh update, $\Delta_1(t)$ increases with unit rate at each time instant. $t_1^{'}, t_2^{'}, ...$ in Figure 2.2 denote the time instants when packets are received successfully at the monitor. At each $t_j^{'}$, $\Delta(t)$

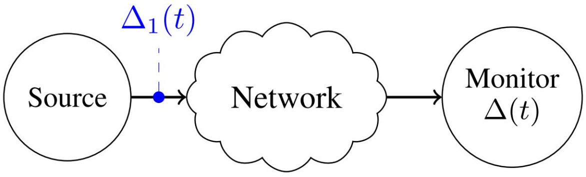


Figure 2.1: A general model of a status update system. The source generates status updates. $\Delta_1(t)$ denotes the age of the packet at the source side. The updates are transmitted to a monitor through the network. The age at the monitor is expressed as $\Delta(t)$. (Figure retrieved from [2, Fig.1(a)]

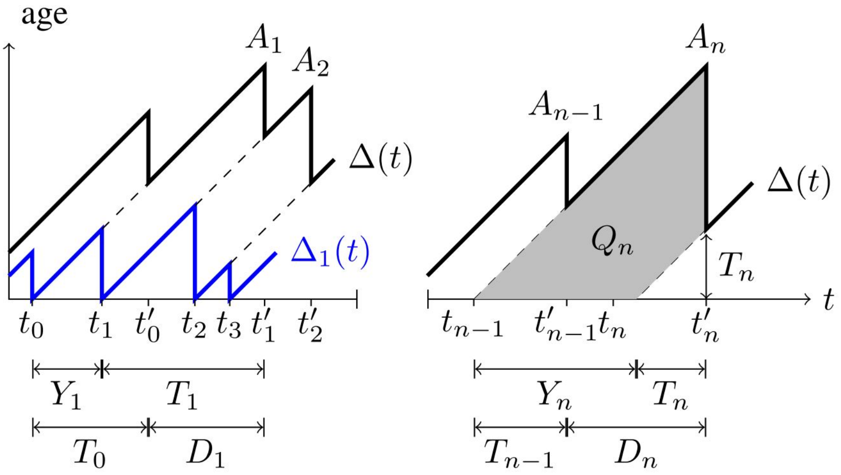


Figure 2.2: The ages at the source side, $\Delta_1(t)$ and the receiver side, $\Delta(t)$ (Figure retrieved from [2, Fig.1(b)]

is reset to $\Delta(t_j^{'}) = t_j^{'} - t_j$, i.e., the age of the j^{th} packet when it is delivered. During the transmission of a packet or when there is no fresh update, $\Delta(t)$ also grows at unit rate. Thus, both $\Delta_1(t)$ and $\Delta(t)$ follow a sawtooth pattern.

There are different metrics related to age. One of the most commonly used metrics is the *time-average age* [2]. For large enough T, the time-average age is

$$\langle \Delta \rangle_T = \frac{1}{T} \int_0^T \Delta(t) dt. \tag{2.2}$$

Average age analysis has been the topic of discussion in many works in the literature [59–63]. A method of analyzing the average age is to use graphics such as in Figure 2.2: The area under the $\Delta(t)$ curve can be decomposed into trapezoidal areas such as Q_n. Denoted with $Y_n = t_n - t_{n-1}$ and $T_n = t_n^{'} - t_n$ are the interarrival time and service time of the n^{th} update, respectively. For a stationary ergodic (Y_n, T_n) process, time average AoI $\Delta = \lim_{T\to\infty}\langle\Delta\rangle_T$ can be calculated as $\Delta = \mathbb{E}[Q_n]/\mathbb{E}[Y_n]$ [2].

Peak age is another important measure of timeliness, indicating the value of age just before an update is correctly received [64]. In Figure 2.2, peak AoI values are displayed as A_n. Mathematically, peak AoI can be expressed as $A_n = T_{n-1} + D_n$ where $D_n = t_n^{'} - t_{n-1}^{'}$ is the interdeparture time. In some applications, such as factory automation and autonomous driving, ensuring the freshness of data is critical. In such cases, age of the data received at the monitor should be below a predetermined threshold value, or else it becomes outdated and irrelevant [65]. This situation is called *peak age violation*, which is further studied in [24, 66, 67].

In this thesis, we focus on a timeliness measure similar to peak age violation, namely, *age violation*, which is defined as the event that the age exceeds a threshold. Age violation differs from peak age violation as it is concerned on the timeliness of the whole process instead of only the peak age. Hence, age violation considers the extreme AoI incidents with a very low probability of happening [66]. Therefore, in applications with stringent requirements such as URLLC, age violation is a more relevant metric for measuring the timeliness compared to average or peak age. Our aim is to minimize the probability of age violation, as studied in [66, 68–70]. Let us denote the threshold with Δ_{max}; then the age violation probability is expressed as

$$P_{av}(\Delta_{max}) = P(\Delta(t) > \Delta_{max}). \tag{2.3}$$

We measure the age violation probability by calculating the ratio of time in which $\Delta(t)$ exceeds the threshold to the total time T [2]:

$$P_{av}(\Delta_{max}) = \lim_{t\to\infty} \frac{1}{T} \int_0^T \mathbb{1}(\Delta(t) > \Delta_{max})dt, \tag{2.4}$$

where $\mathbb{1}(\cdot)$ is the indicator function.

2.2 Finite Blocklength Theory

In communication systems, *information payload* refers to the raw information packet coming from a source. At the transmitter, the information payload is mapped to a signal to be transmitted over a wireless channel, and this mapping is called *channel coding*. The length of the packet after channel coding is the *blocklength*, represented with n. A blocklength of n means that n channel uses are required to transmit the information payload [14].

In status update applications mentioned before, machine-to-machine communications and URLLC systems, n values are considerably small; hence the term *short packet transmission* is generally used to characterize the transmissions in these systems. Short packet transmission is a challenge for several reasons. Firstly, traditional wireless communication systems are not suitable for transmission of short packets: After channel coding and transmission through the channel, the information payload needs to be recovered from the signal distorted by the channel and noise. Shannon's theorem [13] states that for long packets, i.e., for large n, channel codes with a very high probability of recovering the information payload exist. This is because, with long packets, the effects of noise and channel distortion are averaged out according to the law of large numbers. However, this is not the case for short packets. Another difference in short packet transmission is that since the information payload in a coded packet is small, the size of the control information (generally referred to as *metadata*) is not negligible as in long packet transmissions. This leads to a decrease in the efficiency of the transmission [14].

Finite blocklength (FBL) information theory has been a heavily discussed topic in order to address the problems with transmission of short packets. The work of Polyanskiy et al. in 2010 [15] has been a great improvement in FBL theory, as it provided formulations and bounds on maximal achievable coding rate in short packet transmissions.

When an information payload consisting of k bits is mapped to a packet with blocklength n, the ratio k/n expresses the coding rate (R). The packet error probability (P_e), i.e., the probability of an erroneous transmission is strongly related to the rate.

The maximum coding rate with packet length n and packet error probability ϵ is shown with $R^*(n,\epsilon)$, implying the largest rate at which an encoder/decoder pair with packet length n and P_e lower than ϵ exists. [15] provides the following fundamental formulation of the maximum coding rate for FBL codes in additive white Gaussian noise (AWGN) channel:

$$R^*(n,\epsilon) = C - \sqrt{\frac{V}{n}}Q^{-1}(\epsilon) + \mathcal{O}\Big(\frac{\log n}{n}\Big). \tag{2.5}$$

Here, C denotes the *capacity*; the largest rate k/n with an arbitrarily small probability of error when the packet is sufficiently long:

$$C = \lim_{\epsilon \to 0} \lim_{n \to \infty} R^*(n,\epsilon). \tag{2.6}$$

The capacity depends on the signal-to-noise ratio (SNR) γ:

$$C(\gamma) = \log_2(1+\gamma). \tag{2.7}$$

V denotes the *channel dispersion*, which is a measure of the stochastic variability of the channel in comparison to a deterministic channel with the same capacity. It also depends on the SNR as follows:

$$V(\gamma) = \frac{\gamma(\gamma+2)}{2(\gamma+1)^2}\, log_2^2(e). \tag{2.8}$$

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

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty e^{-t^2/2} dt. \tag{2.9}$$

The study of finite blocklength theory has been extended to different channel models such as Rayleigh block-fading channels with multiple antennas [71] and quasi-static fading channels [72], and its practical impact on short packet communications were further discussed in [14, 73, 74].

2.2.1 Modulation in FBL Regime

In this thesis, we aim to minimize the age violation probability not only with blocklength selection, but also with the proper selection of modulation and coding scheme (MCS). However, in [15], an infinite constellation is assumed; thus, Eqn. 2.5 is not suitable for practical systems with constellation diagrams consisting of a limited number of points such as M-ary quadrature amplitude modulation (M-QAM). In such cases, we can not use the capacity definition in Eqn. 2.7. Instead, we need to use the following mutual information bound [75]:

$$\begin{aligned} I(\gamma, M) =& \log_2 M \\ & - \frac{1}{M\pi} \sum_{i=1}^{M} \Big[\int e^{-||y-\sqrt{\gamma}x_i||^2} \times \log_2 \Big(\sum_{k=1}^{M} e^{-||y-\sqrt{\gamma}x_i||^2 - ||y-\sqrt{\gamma}x_k||^2} \Big) dy \Big]. \end{aligned} \tag{2.10}$$

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 [76], the authors provide the following approximation for $I(\gamma, M)$ based on multi-exponential decay curve fitting (M-EDCF):

$$I'(\gamma, M) \approx \log_2 M \times \Big(1 - \sum_{j=1}^{k_M} \varepsilon_j^{(M)} e^{-\vartheta_j^{(M)} \gamma} \Big). \tag{2.11}$$

The coefficients $\varepsilon_j^{(M)}$ and $\vartheta_j^{(M)}$ are provided (see Table 2.1 and 2.2) and the approximation is shown to be in correspondence with the simulation results.

For calculating the maximum coding rate in an equiprobable M-QAM constellation, the capacity C in 2.5 is replaced with $I'(\gamma, M)$ [75], with V and Q defined the same as in Eqn. 2.8 and 2.9, respectively:

Table 2.1: Fitting Coefficients for M-QAM in (2.11)

M^*	k_M	$\varepsilon_1^{(M)}$	$\varepsilon_2^{(M)}$	$\varepsilon_3^{(M)}$	$\varepsilon_4^{(M)}$	$\varepsilon_5^{(M)}$	$\varepsilon_6^{(M)}$
256	6	0.1187	0.3652	0.1658	0.2058	0.1066	0.0379
64	5	0.0435	0.2298	0.1469	0.4452	0.1346	–
16	4	0.2175	0.0486	0.1802	0.5537	–	–
4	3	0.7355	0.2402	0.0243	–	–	–

Table 2.2: Fitting Coefficients for M-QAM in (2.11) - Continued

M^*	k_M	$\vartheta_1^{(M)}$	$\vartheta_2^{(M)}$	$\vartheta_3^{(M)}$	$\vartheta_4^{(M)}$	$\vartheta_5^{(M)}$	$\vartheta_6^{(M)}$
256	6	0.5817	0.0065	0.1734	0.0497	0.0142	1.770
64	5	1.8741	0.1922	0.6441	0.0262	0.0537	–
16	4	0.7442	2.0559	0.2069	0.1090	–	–
4	3	0.5448	1.0226	3.0367	–	–	–

$$R^*(n, \epsilon, M) = I'(\gamma, M) - \sqrt{\frac{V}{n}} Q^{-1}(\epsilon) + \mathcal{O}\Big(\frac{\log n}{n}\Big). \tag{2.12}$$

2.3 Reinforcement Learning Overview

Reinforcement Learning (RL) is a type of machine learning (ML) where an *agent*, the learner and decision maker, learns to maximize an accumulated reward by trial and error in an interactive environment. Figure 2.3 shows the interaction between the agent and the environment, which can be described as follows: At each time step t, the agent obtains the environment's state $S_t \in \mathcal{S}$. Based on the current state, it chooses an action $A_t \in \mathcal{A}$. As a result of this action, the agent observes the new environment state S_{t+1}, and a numerical reward $R_{t+1} \in \mathcal{R}$ in the next time step. The reward is an indication of the effect of the action on the environment.

Major components of an RL agent are *policy*, *value function*, and *model* [77]:

- *Policy* is a mapping from state to action, defining the agent's behavior. Policy can be deterministic ($a = \pi(s)$) or stochastic ($\pi(a|s) = P(A_t = a|S_t = s)$).

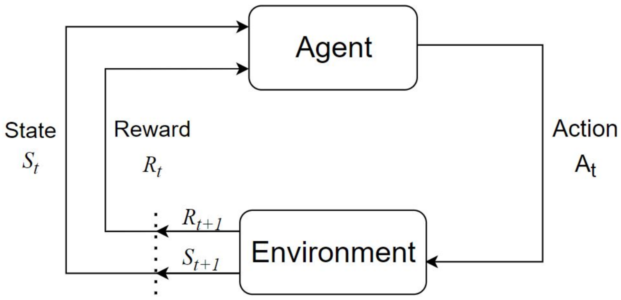


Figure 2.3: Interaction between the agent and the environment in an RL framework

- *State-value function* is the discounted accumulation of the future rewards given state s, following policy π, and it is a measure of the state's value:

$$v_\pi(s) = \mathbb{E}[R_{t+1} + \Gamma R_{t+2} + \Gamma^2 R_{t+3} + ...|S_t = s], \tag{2.13}$$

  where $\Gamma \in [0, 1)$ is the *discount factor*.

- *Action-value function* is similarly the discounted accumulation of the future rewards given action a and state s, following policy π, and it implies how good it is to take action a in state s:

$$Q_\pi(s, a) = \mathbb{E}[R_{t+1} + \Gamma R_{t+2} + \Gamma^2 R_{t+3} + ...|S_t = s, A_t = a]. \tag{2.14}$$

  The relation between $v_\pi(s)$ and $Q_\pi(s, a)$ is expressed as

$$v_\pi(s) = \sum_{a \in \mathcal{A}} \pi(a|s) Q_\pi(s, a). \tag{2.15}$$

- *Model* is a description of the environment dynamics, including state transition probabilities $\mathcal{P}^a_{ss'} \in \mathcal{P}$ and expected reward $\mathcal{R}^a_s \in \mathcal{R}$. $\mathcal{P}^a_{ss'}$ refers to the probability of going from state s to state s' by taking action a while $\mathcal{R}^a_s$ is the reward expected when action a is chosen in state s:

$$\mathcal{P}^a_{ss'} = P(S_{t+1} = s'|S_t = s, A_t = a). \tag{2.16}$$

$$\mathcal{R}^a_s = \mathbb{E}[R_{t+1}|S_t = s, A_t = a]. \tag{2.17}$$

The accumulated discounted reward mentioned before is called *return*:

$$G_t = R_{t+1} + \Gamma R_{t+2} + \Gamma^2 R_{t+3} + ... = \sum_{k=0}^{\infty} \Gamma^k R_{t+k+1}. \tag{2.18}$$

Combining Eqn. 2.13 and 2.18, it can be seen that the expected return starting from the state s gives us the state-value function of s, and it is possible to evaluate $v_\pi(s)$ in a recursive form:

$$\begin{aligned} v_\pi(s) &= \mathbb{E}[G_t | S_t = s] \\ &= \mathbb{E}[R_{t+1} + \Gamma R_{t+2} + \Gamma^2 R_{t+3} + ... | S_t = s] \\ &= \mathbb{E}[R_{t+1} + \Gamma(R_{t+2} + \Gamma R_{t+3} + ...) | S_t = s] \\ &= \mathbb{E}[R_{t+1} + \Gamma G_{t+1} | S_t = s] \\ &= \mathbb{E}[R_{t+1} + \Gamma v_\pi(s') | S_t = s]. \end{aligned} \tag{2.19}$$

The last part of Eqn. 2.19 gives us the recursive form of the state-value function. Similarly, $Q_\pi(s, a)$ can be expressed as

$$Q_\pi(s, a) = \mathbb{E}[R_{t+1} + \Gamma Q_\pi(s', a') | S_t = s, A_t = a]. \tag{2.20}$$

Both $v_\pi(s)$ and $Q_\pi(s, a)$ can be written in terms of the state transition probabilities $\mathcal{P}_{ss'}^a$ and reward functions $\mathcal{R}_s^a$:

$$v_\pi(s) = \sum_{a \in \mathcal{A}} \pi(a|s) \Big(\mathcal{R}_s^a + \Gamma \sum_{s' \in \mathcal{S}} \mathcal{P}_{ss'}^a v_\pi(s') \Big). \tag{2.21}$$

$$Q_\pi(s, a) = \mathcal{R}_s^a + \Gamma \sum_{s' \in \mathcal{S}} \mathcal{P}_{ss'}^a \sum_{a' \in \mathcal{A}} \pi(a'|s') Q_\pi(s', a'). \tag{2.22}$$

Eqs. 2.21 and 2.22 are called *Bellman expectation equations* for state-value function and action-value function, respectively [77].

The maximum state-value function over all policies is the optimal state-value function, $v_*(s) = \max_\pi v_\pi(s)$. Likewise, $Q_*(s,a) = \max_\pi Q_\pi(s,a)$ is the optimal action-value function over all policies. For optimal value functions, Eqs. 2.21 and 2.22 become *Bellman optimality equations*:

$$v_*(s) = \max_a \mathcal{R}_s^a + \Gamma \sum_{s' \in \mathcal{S}} \mathcal{P}_{ss'}^a v_*(s'). \tag{2.23}$$

$$Q_*(s,a) = \mathcal{R}_s^a + \Gamma \sum_{s' \in \mathcal{S}} \mathcal{P}_{ss'}^a \max_{a'} Q_*(s',a'). \tag{2.24}$$

There are various iterative methods for solving the optimality equations, some of which we will explain in detail in the future sections.

2.4 Markov Decision Process

Markov Decision Process (MDP) is a stochastic control process generally used in RL problems for a complete description of an environment. An MDP is a 5-tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \Gamma \rangle$ [78]. The elements of the tuple are defined as follows:

- State space $\mathcal{S}$: The set consisting of all possible states the agent can observe.

- Action space $\mathcal{A}$: The set consisting of all possible actions the agent can take.

- Transition probability space $\mathcal{P}$: The transition model of the system that consists of state transition probabilities $\mathcal{P}_{ss'}^a$, as defined in Eqn. 2.16.

- Reward function $R : \mathcal{S} \times \mathcal{A} \to \mathcal{R}$. $\mathcal{R}$ is the set of possible rewards $\mathcal{R}_s^a$ (see Eqn. 2.17)

- Discount factor $\Gamma \in [0,1]$: Discount factor determines the importance given to future rewards.

2.5 Dynamic Programming

Dynamic programming (DP) is a set of methods used in problems with optimal substructure and overlapping problems [79]. Having an optimal substructure means that *Bellman's principle of optimality* [80] applies: An optimal solution consists of subsolutions that are the optimal solutions of subproblems. On the other hand, overlapping problems refer to recurring subproblems whose solutions can be saved and reused. MDPs fit into the category of problems handled with DP because of the recursive decomposition provided by Bellman's equation (see Eqn. 2.23), and the value function storing and reusing the solutions [77]. Dynamic programming methods require full knowledge of the system model, i.e., the state transition probabilities and the reward models.

In this study, we focus on *value iteration*, which is a DP method for finding an optimal policy based on the state-value function.

2.5.1 Value Iteration

In a dynamic programming approach, the problem of finding the optimal state-value function $v_*(s)$ is decomposed into subproblems $v_*(s')$. If the solutions to $v_*(s')$ are known, then the solution of $v_*(s)$ can be found (see Eqn. 2.23).

In *value iteration (VI)* method, $v_*(s)$ is updated iteratively. First, for all $s \in \mathcal{S}$, $v(s)$ are initialized. At each iteration, state-value functions of all states $s \in \mathcal{S}$ are updated, using the maximum of all possible actions. The iteration terminates when the changes in $v(s)$ in consecutive iterations become arbitrarily small. Detailed instructions for value iteration are given in Algorithm 1.

The outcome of the value iteration is a deterministic policy, mapping the states to actions:

$$\pi(s) = \arg\max_{a} \; \mathcal{R}_s^a + \Gamma \sum_{s' \in \mathcal{S}} \mathcal{P}_{ss'}^a v_*(s'). \tag{2.25}$$

Algorithm 1 Value Iteration [81]

```
Initialize $v$ arbitrarily, e.g. $v(s) = 0$, for all $s \in \mathcal{S}$
Repeat
    $\Delta \leftarrow 0$
    For each $s \in \mathcal{S}$:
        $v \leftarrow v(s)$
        $v(s) \leftarrow \max_a \; \mathcal{R}_s^a + \Gamma \sum_{s' \in \mathcal{S}} \mathcal{P}_{ss'}^a v(s')$
        $\Delta \leftarrow \max(\Delta, |v - v(s)|)$
until $\Delta < \theta$ (a small positive number)
Output a deterministic policy $\pi$ such that
$\pi(s) = \arg\max_a \sum_{s'} \mathcal{P}_{ss'}^a [\mathcal{R}_s^a + \Gamma v(s')]$
```

2.6 Q-Learning

The solution of the optimal action-value function in Eqn. 2.24 requires knowledge of the state transition probabilities and reward models. However, it may not be possible to obtain a complete model of the environment dynamics in many problems. In such cases, *model-free* methods that learn by trial and error should be used.

One of the most commonly used model-free methods is Q-learning, a reinforcement learning algorithm that aims to find the optimal action-value function, or *Q-function*, $Q_*(s, a)$ (see Eqn. 2.24). Q-learning is an off-policy temporal difference (TD) algorithm. The agent has no prior knowledge about the environment, and it learns through episodes of trial and error, following a *behavior policy* that is different from the learned *target policy* to generate behavior [81, p.103].

The Q-learning agent faces a trade-off between exploration and exploitation [82]. Exploitation refers to choosing the action with the highest action-value estimate; hence it is also called the *greedy* approach. On the other hand, exploration is choosing 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.

At each step in an episode of the Q-learning algorithm, the agent chooses an action according to the ε-greedy policy, and observes the reward r and new state s'. Then the Q-function is updated according to the following rule:

$$Q(s,a) \leftarrow Q(s,a) + \alpha(r + \Gamma \max_{a'} Q(s',a') - Q(s,a)). \tag{2.26}$$

The parameter α, $0 \leq \alpha \leq 1$, is the *learning rate* or *step size*. It is a measure of the prominence given to new value compared to the old value. A higher learning rate implies more rapid changes in $Q(s,a)$. Under the assumption that all state-action pairs continue to be updated, and the parameters ϵ and α are set correctly, $Q(s,a)$ converges to the optimal value $Q_*(s,a)$ [81].

Q-learning algorithm is given in detail in Algorithm 2

Algorithm 2 Q-Learning [81]

```
Algorithm parameters: step size $\alpha \in (0,1]$, small $\varepsilon > 0$, discount factor $\Gamma \in (0,1]$
Initialize $Q(s,a)$ arbitrarily $\forall s \in \mathcal{S}$ and $\forall a \in \mathcal{A}$
Repeat (for each episode)
    Initialize $s$
    Loop for each step of episode:
        Choose $a$ from $s$ using policy derived from $Q$ (e.g., $\epsilon$-greedy)
        Take action $a$, observe $r$, $s'$
        $Q(s,a) \leftarrow Q(s,a) + \alpha(r + \Gamma * \max_{a'} Q(s',a') - Q(s,a))$
        $s \leftarrow s'$
    until $s$ is terminal
```

2.7 Deep Reinforcement Learning

In many decision-making problems with large state spaces, traditional RL algorithms such as Q-learning fails to be an adequate solution. The main reason is that Q-learning is a tabular method: action-value functions $Q(s,a)$ for all states $s \in \mathcal{S}$ and actions $a \in \mathcal{A}$ are stored in a table, which becomes impractical for large numbers of states or actions because the required memory and computation resources are too high. Such complex problems require using tools of *deep reinforcement learning (DRL)*, i.e., the integration of deep learning with reinforcement learning [82]. Deep reinforcement learning is a function approximation technique that uses deep neural networks (DNN). The action-value function $Q(s, a)$ is approximated by $Q(s, a; \theta)$, where θ is the vector consisting of the weights of the DNN mimicking the actual $Q(s, a)$.

A basic model of the deep neural network used in DRL is shown in Figure 2.4. The network can also be called a *deep Q network (DQN)*. It consists of an input layer, L hidden layers and an output layer. The network takes a state as an input, and as outputs, it gives the action-value functions for all possible actions.

The DQN agent learns with experiences. An experience can be represented with a (s, a, r, s') tuple: The state s, the action a taken in state s, the reward r obtained by taking action a in state s, and the resulting next state s'. 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 [83].

Another feature of DQN that limits the instabilities is 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 ($Q(s, a; \theta)$), and the target network is shown as $\hat{Q}(s, a; \theta^-)$. While main network is actively trained, the target network is updated at every N episodes. The purpose is to prevent rapid changes in $\hat{Q}(s, a; \theta^-)$, and avoid chasing a moving target. This improves the stability and raises the probability of convergence.

Figure 2.5 shows the main components of the DQN and the information flow between them. At each time step in an episode, the agent chooses an action a with an ε-greedy approach: with probability ε, a random action is selected. Otherwise, the action with the maximum Q value is selected. Execution of action a results in reward r and state s'. The experience (s, a, r, s') 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:

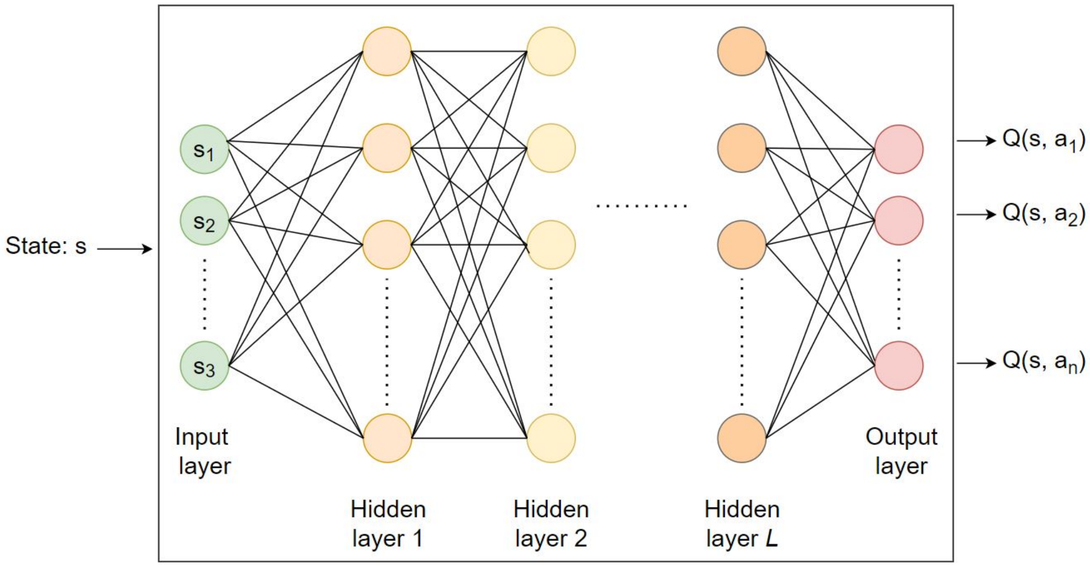


Figure 2.4: The structure of a deep neural network. The first layer takes the state as input. Hidden layers are located between the input and output layers. At the output layer, Q-functions of all of the actions are given.

$$L(\theta) = ((r + \Gamma \max_{a'} \hat{Q}(s, a; \theta^{-})) - Q(s, a; \theta))^2. \tag{2.27}$$

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 deep Q-learning is given in Algorithm 3.

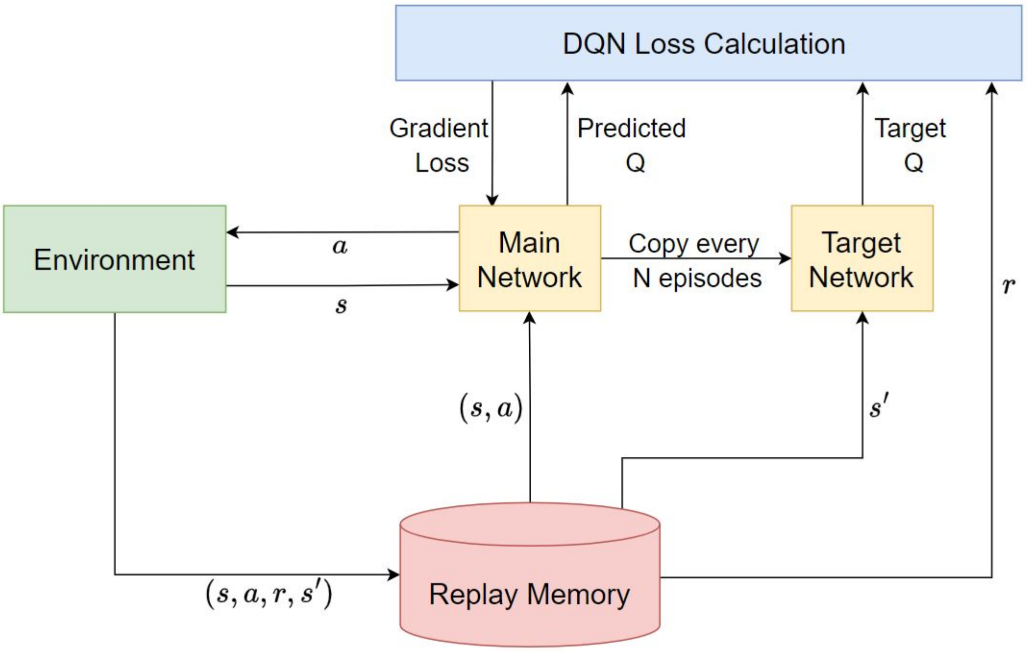


Figure 2.5: DRL model showing the information flow between the components. The main network selects action a according to the state s of the environment. The environment responds with reward r and next state s', and the tuple of (s, a, r, s') is stored in the replay buffer. A minibatch of experiences is sampled randomly from the replay buffer for training. Gradient loss is calculated. At every N episodes, the weights of the main network are copied to the target network.

Algorithm 3 Deep Q-learning with Experience Replay [82]

```
Initialize replay memory $\mathcal{D}$ to capacity $L$
Initialize action-value function $Q$ with random weights $\theta$
Initialize target action-value function $\hat{Q}$ with random weights $\theta^{-}$
for episode = 1, $M$ do
    Initialize state $s_t$
    for $t = 1, T$ do
        With probability $\varepsilon$, select a random action $a_t$
        otherwise, select $a_t = \max_a Q^*(s_t, a; \theta)$
        Execute action $a_t$ and observe reward $r_t$ and state $s_{t+1}$
        Store transition $(s_t, a_t, r_t, s_{t+1})$ in $\mathcal{D}$
        Set $s_{t+1} = s_t$
        Sample random minibatch of transitions $(s_j, a_j, r_j, s_{j+1})$ from $\mathcal{D}$
        Set $y_j = \begin{cases} r_j & \text{for terminal } s_{j+1} \\ r_j + \Gamma \max_{a'} \hat{Q}(s_{j+1}, a'; \theta) & \text{for non-terminal } s_{j+1} \end{cases}$
        Perform a gradient descent step on $(y_j - Q(s_j, a_j; \theta))^2$
        In every $N$ steps, reset $\hat{Q} = Q$, i.e., set $\theta^{-} = \theta$
    end
end
```

CHAPTER 3

ADAPTIVE BLOCKLENGTH SELECTION FOR MINIMIZING AGE VIOLATION PROBABILITY

In this chapter, we demonstrate the system model for the blocklength selection problem and present our solutions based on dynamic programming and reinforcement learning. We provide simulation results comparing our methods' performances with baseline solutions.

3.1 System Model

We consider a single-server queue with capacity 2, where information packets are generated at the source according to Bernoulli distribution. The probability of a new packet arrival in one channel use (CU) is λ. The queue works with Last Come First Serve (LCFS) policy, specifically, LCFS with preemption in the queue (LCFS-Q) as defined in [67]. According to the policy, if a new packet generated at the source arrives at the queue while it is empty, it is immediately sent to the server. On the other hand, if the queue is non-empty at the time of the new arrival, the packet waiting in the queue is discarded, and replaced by the newly arrived packet in the queue. It has been shown in [64] that LCFS policy is advantageous compared to First Come First Serve (FCFS) policy in applications where the main concern is AoI or delay.

The packet generated at the source consists of k information bits. At the server, it is mapped to a codeword with blocklength n, and then transmitted to the receiver through the wireless channel. Figure 4.1 depicts our system model.

The channel model is assumed to be a memoryless block-fading Rayleigh channel:

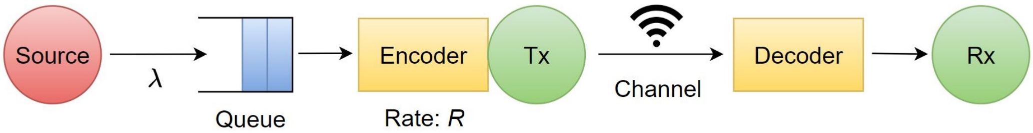


Figure 3.1: System model for the blocklength selection problem

the fading coefficient stays constant for a block of symbols [84]. Here we assume that each transmitted packet experiences identically and independently distributed (IID) fading coefficients. The input-output relation of the channel is

$$y = x \cdot h + w, \tag{3.1}$$

where x and y are the transmitted and received symbols, respectively. h denotes the fading coefficient of the channel, and lastly, w is the additive noise.

Channel state information at the transmitter (CSIT) is assumed. Let us denote the transmit power with P. Assuming noise with standard normal distribution ($\mathcal{N}(0, 1)$), instantaneous SNR is expressed as

$$\gamma = P|h|^2. \tag{3.2}$$

We denote with ϵ the block error rate (BLER), i.e., the probability that the transmitted packet is not decoded correctly at the receiver. According to finite blocklength theory, rewriting Eqn. 2.5, BLER for a code with rate $R = k/n$ and SNR γ is [85]:

$$\epsilon(\gamma) \approx Q\left(\frac{C(\gamma) - \frac{k}{n}}{\sqrt{\frac{V(\gamma)}{n}}}\right), \tag{3.3}$$

where $C(\lambda)$ denotes the channel capacity and $V(\lambda)$ is the channel dispersion as defined in Eqs. 2.7 and 2.8, respectively.

At any time instant t, the age at the receiver $\Delta_r(t)$ is defined as the time elapsed since the generation of the last packet that was successfully received:

$$\Delta_r(t) = t - u(t), \tag{3.4}$$

where $u(t)$ is the time stamp of the packet. Hence, if there is no packet in the system, the age at the receiver keeps increasing. Similarly, an erroneous transmission also increases the age. Figure 3.2 shows the changes in $\Delta_r(t)$ resulting from packet arrivals and successful/unsuccessful transmissions.

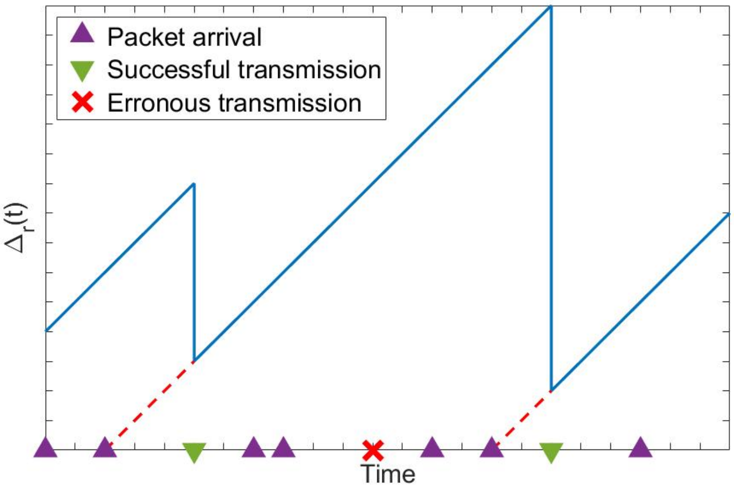


Figure 3.2: Age at the receiver ($\Delta_r(t)$) continues to increase until the packet in service is transmitted successfully. When a transmission error occurs, the packet in service is discarded, and the packet in the queue gets service. $\Delta_r(t)$ is reset to the age at the queue when a packet is transmitted correctly. When there is no packet in the queue, $\Delta_r(t)$ keeps increasing.

Our purpose is to minimize the probability that $\Delta_r(t)$ exceeds a threshold Δ_{max} by choosing the blocklength dynamically. To calculate the age violation probability, we use the following formulation:

$$\begin{aligned} P_{av}(\Delta_{max}) &= \lim_{t\to\infty} P(\Delta_r(t) > \Delta_{max}), \\ &= \lim_{t\to\infty} \frac{1}{T} \sum_{t=1}^{T} \mathbb{1}(\Delta_r(t) > \Delta_{max}), \end{aligned} \tag{3.5}$$

where $\mathbb{1}(\cdot)$ is the indicator function. This is slightly different from the age violation probability expression in Eqn. 2.4: Since time is discrete, instead of integration, we use summation. We count the total number of age violations in a sufficiently long time T, and dividing it by T gives the age violation probability.

3.2 Proposed Solutions

We propose two solutions for the blocklength selection problem; the first one is based on value iteration method, while the second one utilizes Q-learning. In both approaches, we consider our system as a Markov decision process and define the states, actions, and reward as follows:

- $\mathcal{S} = (\Delta_q, \Delta_r, CQI)$
  - Age of the packet in the queue ($\Delta_q(t)$): The time elapsed since the last packet arrival to the system.
  - Age at the receiver ($\Delta_r(t)$): Time elapsed since the arrival of the last successfully decoded packet to the system.
  - Channel quality indicator (CQI): A measure of the channel condition depending on the SNR, defined as in [46]:

$$CQI = \begin{cases} 0, & SNR \leq SNR_{min}, \\ (N_{cqi} - 1), & SNR \geq SNR_{max}, \\ \frac{(SNR - SNR_{min})(N_{cqi}-1)}{SNR_{max} - SNR_{min}}, & \text{otherwise}, \end{cases} \tag{3.6}$$

    where SNR_{min} and SNR_{max} are the minimum and maximum SNR values, respectively, and N_{cqi} is the total number of CQI states.

    In the construction of the state space, we apply *state aggregation* [86]: we combine similar states into groups to reduce the number of states, hence reducing the complexity of the problem. Here, although the time unit is one channel use, $\Delta_q(t)$ and $\Delta_r(t)$ components of the state does not point to a single value, but a collection of m values. Hence, the mapping from ages at the queue and the receiver to the states $\Delta_q(t)$ and $\Delta_r(t)$ is not direct.

- $\mathcal{A}$: The set of blocklengths, plus *stay idle* action.
- $\mathcal{R}$: The reward includes a function of the age at the receiver, but it also takes into account the states in which the queue is empty, which we express as $\Delta_q(t) = -1$. In these states, since there are no packets to transmit, there should be no blocklength selection. The system should stay idle until a new packet arrives. Thus, the corresponding state-action pair is given a reward of zero. To prevent the agent from choosing a blocklength when the queue is empty, we assign large negative rewards to the corresponding state-action pairs. Similarly, staying idle when a packet is waiting in the queue also results in such rewards. Denoting the action of staying idle as $a = a_0$, we formulate the reward function as follows:

$$\mathcal{R}_s^a = \begin{cases} -1000, & \Delta_q(t) = -1 \ \ \& \ \ a \neq a_0, \\ -1000, & \Delta_q(t) \neq -1 \ \ \& \ \ a = a_0, \\ 0, & \Delta_q(t) = -1 \ \ \& \ \ a = a_0, \\ -\sum_{k=1}^{n} \mathbb{1}(\Delta_r(t) > \Delta_{max}), & \text{otherwise.} \end{cases} \tag{3.7}$$

3.2.1 Value Iteration Based Adaptive Blocklength Selection

As a dynamic programming method, value iteration requires knowledge of the state transition probabilities $\mathcal{P}_{ss'}^a$ and reward function $\mathcal{R}_s^a$ given in Eqs. 2.16 and 2.17, respectively. Thus, we need to consider all possible state transitions and mathematically express their probabilities.

Let us start with the case where the queue is empty ($\Delta_q(t) = -1$). When in this state, the system remains idle for one CU. The next state depends on whether there is a new arrival or not in the current CU. Let us define a function $f(i,j)$ for $j \leq i$ where $f(i,j) = 1$ means that in a time duration of i CUs, the last packet arrival occurred during the j^{th} CU. Whereas, $f(i,j) = 0 \ \forall j$ refers to the case of no packet arrivals throughout the i CUs. For an arrival rate of λ, corresponding probabilities of these two events can be written as

$$P(f(i,j)=1)=\lambda\cdot(1-\lambda)^{i-j}. \tag{3.8}$$

$$P(f(i,j)=0)=(1-\lambda)^{i}. \tag{3.9}$$

The change in state Δ_q depends on the outcome of $f(1,1)$:

$$\Delta_q(t+1)=\begin{cases}-1, & \Delta_q(t)=-1 \;\;\&\;\; f(1,1)=0,\\ 0, & \Delta_q(t)=-1 \;\;\&\;\; f(1,1)=1.\end{cases} \tag{3.10}$$

In both scenarios (packet arrival or no packet arrival), there is not any successful transmission, hence the age at the receiver increases:

$$\Delta_r(t+1)=\Delta_r(t)+1. \tag{3.11}$$

When $\Delta_q(t)\neq -1$, i.e., there is a packet at the service, the probabilities about Δ_q and Δ_r need to be calculated. $\Delta_q(t+n)$ depends on whether a new packet arrived to the queue during n CUs, while $\Delta_r(t+n)$ depends on whether a block error occurred or not:

$$\Delta_q(t+n)=\begin{cases}-1, & \Delta_q(t)\neq -1 \;\;\&\;\; f(n,j)=0\,\forall j,\\ n-j, & \Delta_q(t)\neq -1 \;\;\&\;\; f(n,j)=1.\end{cases} \tag{3.12}$$

$$\Delta_r(t+n)=\begin{cases}\Delta_r(t)+n, & \text{with probability } \epsilon,\\ \Delta_q(t)+n, & \text{with probability } (1-\epsilon).\end{cases} \tag{3.13}$$

Different from $\Delta_q(t)$ and $\Delta_r(t)$, the change in the CQI state is completely independent of other states and the previous CQI state. SNR is calculated as $\gamma = P|h|^2$ where the channel coefficient h is a Rayleigh random variable. Since the probability density function of the Rayleigh distribution is known, probabilities corresponding to the defined SNR, hence CQI, intervals can be calculated.

Knowing $\mathcal{P}_{ss'}^a$, we can calculate the expected reward and find the optimal policy using Algorithm 1. First, we initialize the state-value function $v(s)$ for all $s \in \mathcal{S}$. At each iteration, state-value functions of all states are updated. In each state, we calculate the following for all actions $a \in \mathcal{A}$:

$$\mathcal{R}_s^a + \Gamma \sum_{s' \in \mathcal{S}} \mathcal{P}_{ss'}^a v(s'). \tag{3.14}$$

Then we find the action that maximizes 3.14, and update $v(s)$ according to the following rule:

$$v(s) \leftarrow \max_a \ \mathcal{R}_s^a + \Gamma \sum_{s' \in \mathcal{S}} \mathcal{P}_{ss'}^a v(s'). \tag{3.15}$$

After completing the iterations, we obtain a deterministic policy, i.e., a mapping from the states s to the actions a:

$$\pi(s) = \arg\max_a \sum_{s'} \mathcal{P}_{ss'}^a [\mathcal{R}_s^a + \Gamma v(s')]. \tag{3.16}$$

Table 3.1 lists the parameters we use in value iteration based adaptive blocklength selection method.

3.2.1.1 Simulation Results

Before displaying the performance of our value iteration based adaptive blocklength selection method (VI-ABM), we first show the existence of the optimal blocklength in various scenarios. With a fixed number of information bits ($k = 100$), different blocklengths n imply different coding rates $R = k/n$. In Figure 3.3, we plot the coding rate versus age violation probability for different transmit powers P. The minimum P_{av} values for each P are shown with red circles. It is clear that, the rate, hence, the blocklength that minimizes P_{av} differs as P increases. Similarly, as

Table 3.1: Simulation parameters of value iteration based adaptive blocklength selection method

Parameter	**Value**
Number of information bits (k)	100
Blocklengths (n)	(100, 125, 150, 175, 200, 225, 250, 275, 300)
Transmit power (P)	0 dB
SNR_{min}	-20 dB
SNR_{max}	10 dB
Packet arrival rate (λ)	0.01
Age threshold (Δ_{max})	800 CUs
Number of iterations	200
Discount factor (Γ)	0.95

seen in Figures 3.4 and 3.5, changing the packet arrival rate λ or the threshold Δ_{max} changes the optimal rate. These figures illustrate the motivation behind our adaptive blocklength scheme: we aim to find and use the optimal blocklength according to the current condition of the system so that P_{av} is minimized.

Let us now inspect the blocklength selection policies obtained with VI-ABM. Figure 3.6 shows the effect of $\Delta_r(t)$ and the SNR on the selected blocklength, where $\Delta_q(t)$ is constant. It can be seen that when SNR is low, a small blocklength is selected because the probability of a successful transmission is low for all blocklengths. When SNR increases, larger blocklengths are used as they can guarantee low BLER. For highest SNR levels, the chosen blocklengths are small again, as they can provide low error probabilities at this point. For varying $\Delta_r(t)$, the blocklength selection can be explained as follows: when $\Delta_r(t)$ is low, it is reasonable to use the large blocklengths because the threshold is not exceeded. As $\Delta_r(t)$ increases, smaller blocklengths are selected.

Figure 3.7 displays the effect of $\Delta_r(t)$ and $\Delta_q(t)$ on blocklength selection for constant SNR. The effect of $\Delta_r(t)$ is the same as in Figure 3.14 while the changes in $\Delta_q(t)$ causes a certain pattern in the selected blocklength for fixed $\Delta_r(t)$: As $\Delta_q(t)$ increases, the blocklength of choice becomes smaller. This is because when a successful transmission happens, $\Delta_r(t)$ is reset to $\Delta_q(t)$, hence it is reasonable to prevent $\Delta_q(t)$ from becoming too large. The leftmost side of $\Delta_q(t)$ axis shows the empty

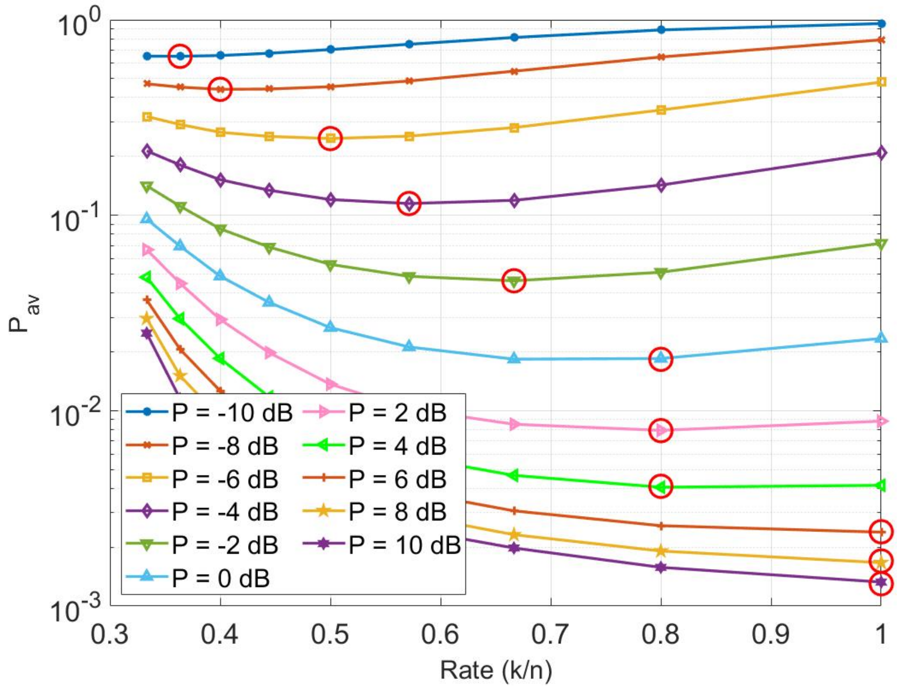


Figure 3.3: Coding rate versus P_{av} for different transmit power levels ($\lambda = 0.01$, $\Delta_{max} = 800$ CUs)

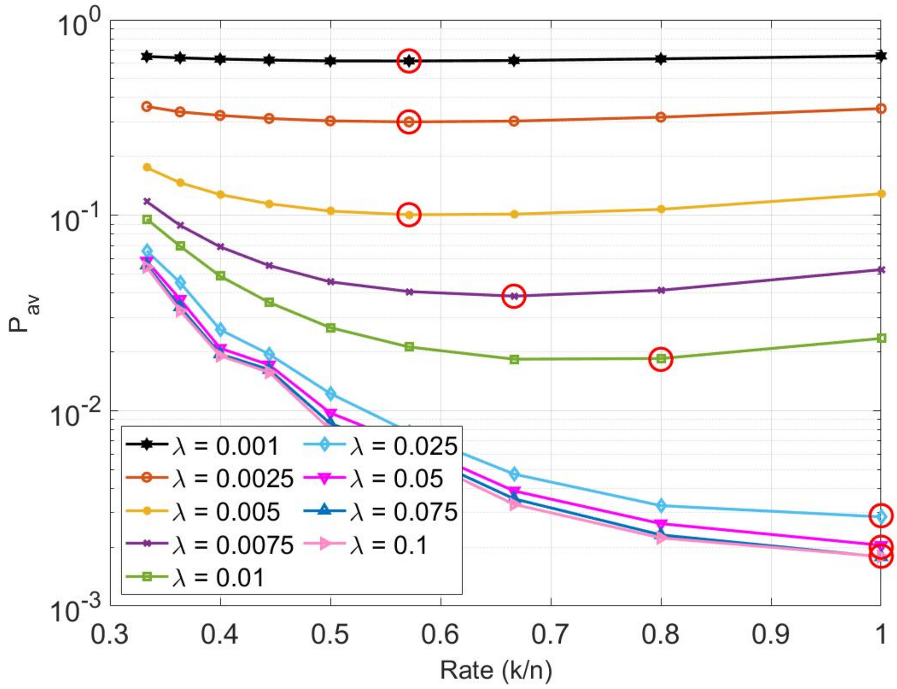


Figure 3.4: Coding rate versus P_{av} for different arrival rates ($P = 0$ dB, $\Delta_{max} = 800$ CUs)

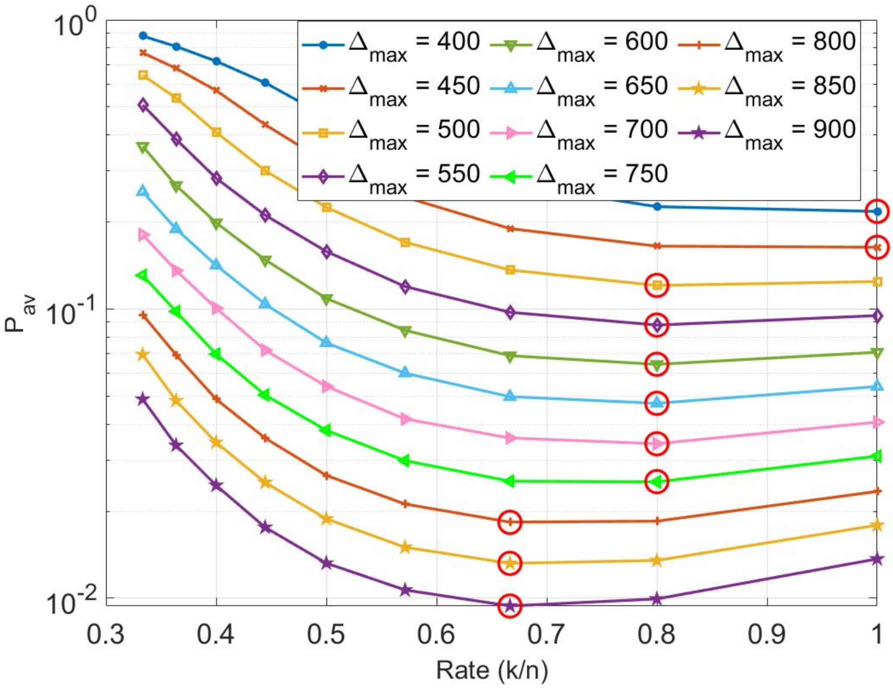


Figure 3.5: Coding rate versus P_{av} for different transmit power levels ($\lambda = 0.01$, $\Delta_{max} = 800$ CUs)

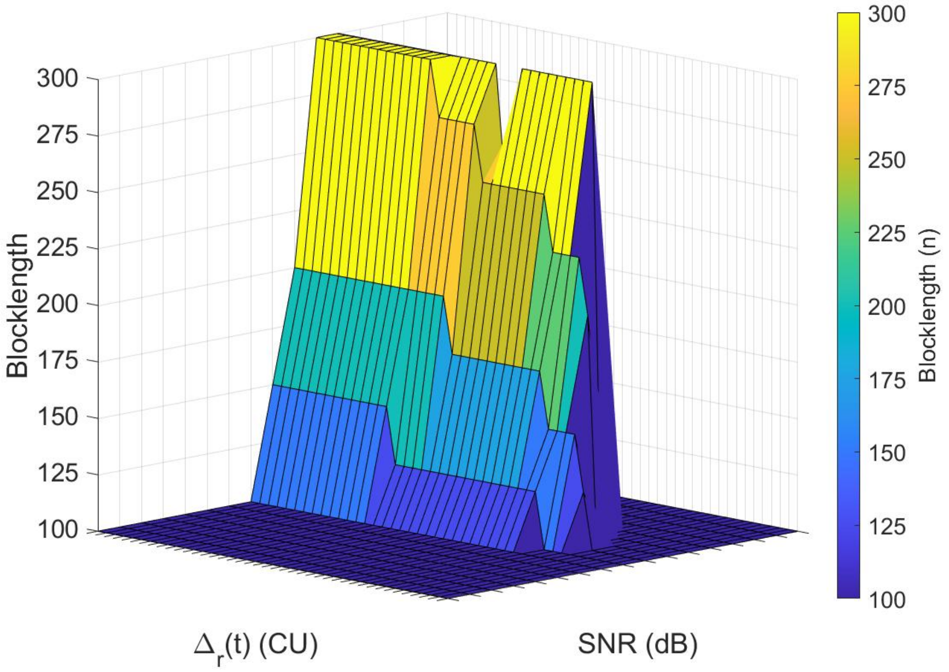


Figure 3.6: Blocklength selection according to $\Delta_r(t)$ and SNR in VI-ABM for fixed $\Delta_q(t)$

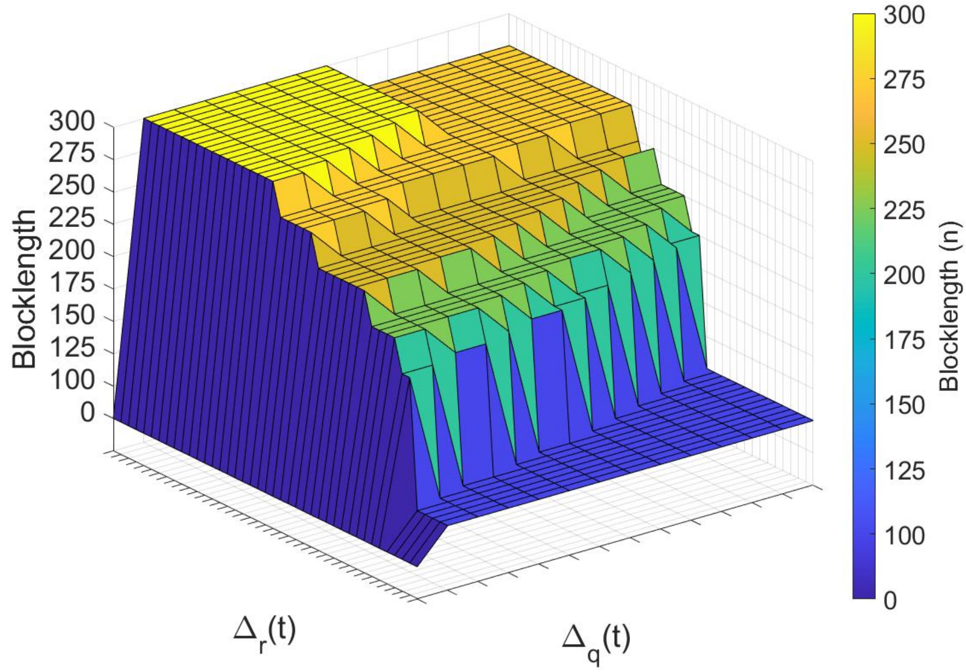


Figure 3.7: Blocklength selection according to $\Delta_r(t)$ and $\Delta_q(t)$ in VI-ABM for fixed SNR

queue case, i.e., $\Delta_q(t) = -1$. In this case, there is no transmission and the system waits for a new packet arrival.

After displaying the blocklength selection, we now compare VI-ABM with the fixed blocklength schemes. Figure 3.8 shows the results obtained with different transmit power (P) levels where the arrival rate and threshold are fixed ($\lambda = 0.01$ and $\Delta_{max} = 800$ CUs). Low transmit power means the states where SNR is low are seen more frequently. It can be seen that for low P values, among all the fixed blocklength schemes, large blocklengths ($n \geq 200$) result in lower P_{av}. This is because more redundancy bits are needed for reliable transmission, i.e., low BLER, in low SNR cases. As the transmit power P increases, smaller n values such as 100 and 125 become advantageous. On the other hand, our adaptive blocklength method provides much lower P_{av} since it chooses the optimal blocklength in all conditions.

In Figure 3.9, 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

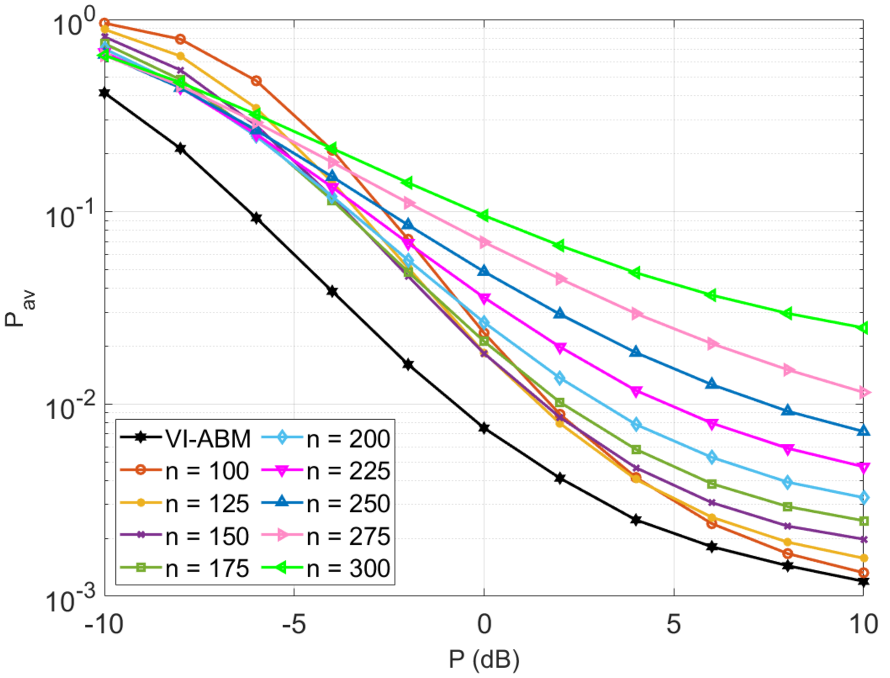


Figure 3.8: Comparison of P_{av} for VI-ABM and fixed blocklength schemes for different transmit power levels ($\lambda = 0.01$, $\Delta_{max} = 800$ CUs)

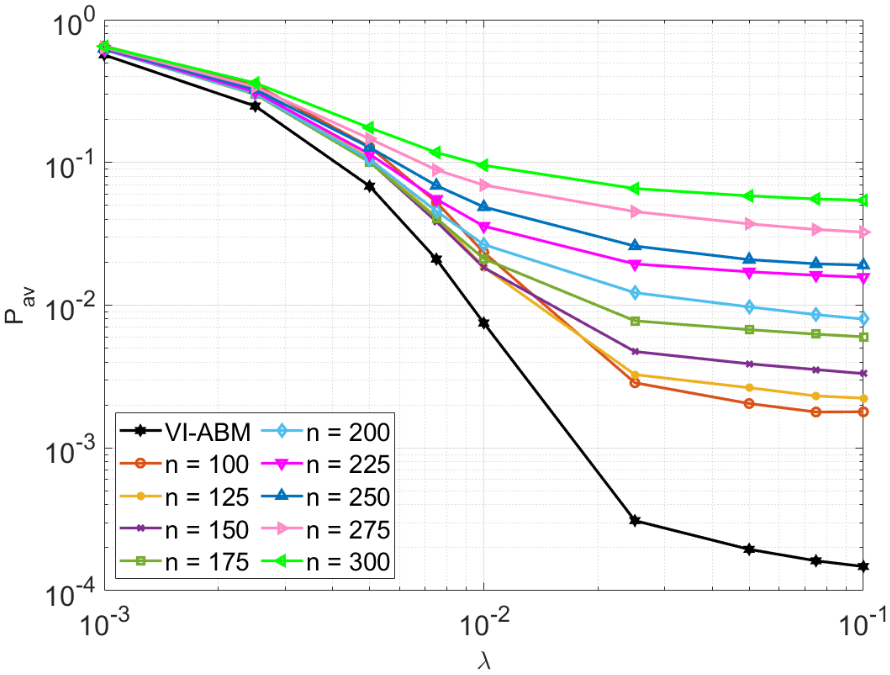


Figure 3.9: Comparison of P_{av} for VI-ABM and fixed blocklength schemes for different arrival rates ($P = 0$ dB, $\Delta_{max} = 800$ CUs)

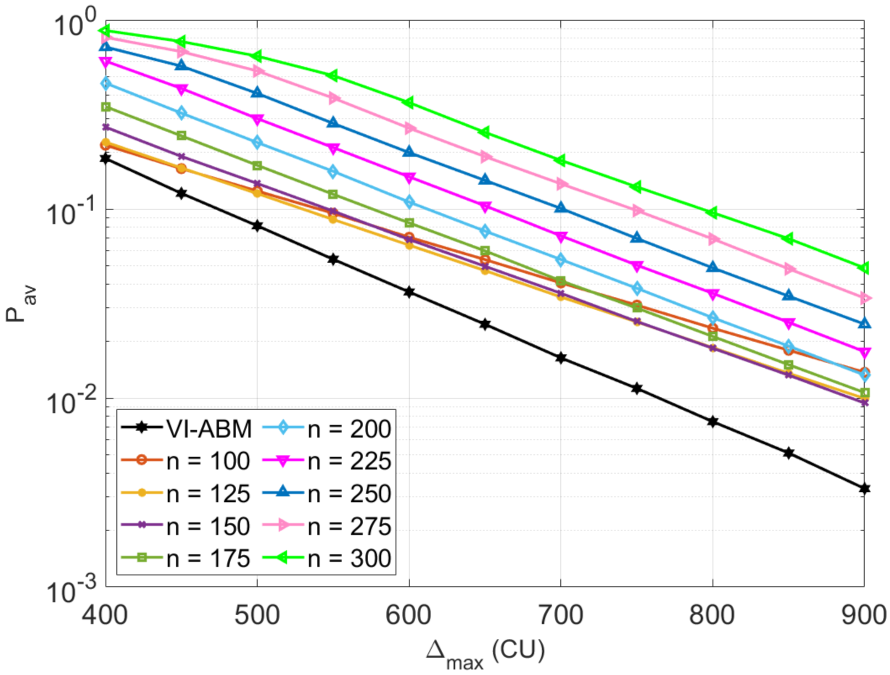


Figure 3.10: Comparison of P_{av} for VI-ABM and fixed blocklength schemes for different age thresholds ($P = 0$ dB, $\lambda = 0.01$)

packet arrival. Thus, for both the fixed blocklength schemes and our method, P_{av} is very high. As λ increases, these idle periods are shortened; hence P_{av} decreases for all schemes. When $\lambda = 0.1$, the probability of discarding a packet in the queue with a newly-arrived packet is high. This leads to smaller Δ_q; therefore, smaller Δ_r and P_{av}. The performance of VI-ABM is superior to fixed blocklength schemes for all arrival rates, while the performance gap is more significant for larger λ.

In Figure 3.10, 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 Δ_{max} values, P_{av} is large for all cases, as expected. As Δ_{max} is increased, P_{av} decreases steadily for all schemes. For all threshold values, VI-ABM outperforms the fixed blocklength schemes and provides considerably lower age violation probabilities as threshold increases.

Table 3.2: Simulation parameters of Q-learning based adaptive blocklength selection method

Parameter	**Value**
Number of information bits (k)	100
Blocklengths (n)	(100, 125, 150, 175, 200, 225, 250, 275, 300)
Transmit power (P)	0 dB
SNR_{min}	-20 dB
SNR_{max}	10 dB
Packet arrival rate (λ)	0.01
Age threshold (Δ_{max})	800 CUs
Number of iterations	100000
Discount factor (Γ)	0.9
Maximum exploration rate (ϵ_{max})	0.5
Minimum exploration rate (ϵ_{min})	0.01
Maximum learning rate (α_{max})	0.5
Minimum learning rate (α_{min})	0.3
Decay rate	0.9999

3.2.2 Q-Learning Based Adaptive Blocklength Selection

The second solution we propose is adaptive blocklength selection based on Q-learning. As Q-learning is a model-free method, state transition probabilities are not required; the agent only learns from trial and error. Firstly, we initialize the action-value functions $Q(s, a)$ to zero for all states $s \in \mathcal{S}$ and all actions $a \in \mathcal{A}$. We follow an ε-greedy policy with a decaying exploration rate: at each iteration, the exploration rate ε is multiplied by a decay rate. In each iteration, according to the observed state s, the agent has to select either to use a blocklength n or to stay idle for one CU. After the action is executed, the environment goes to the next state s', and returns reward r. We update the corresponding Q-table entry $Q(s, a)$ according to Bellman's rule:

$$Q(s, a) \leftarrow Q(s, a) + \alpha(r + \Gamma \max_{a'} Q(s', a') - Q(s, a)). \tag{3.17}$$

Algorithm 4 gives detailed instructions for the Q-learning based adaptive blocklength selection method and Table 3.2 lists the related parameters.

Algorithm 4 QL-ABM

```
Initialize $Q(s,a)$ $\forall s \in \mathcal{S}$ and $\forall a \in \mathcal{A}$
for each blocklength selection point do
    Observe state $s = (\Delta_q(t), \Delta_r(t), CQI)$
    Select action $a$ according to $\varepsilon$-greedy policy: choose a blocklength $n$ or stay idle
    Go to next state $s'$ and receive reward $r$:
        if queue is empty ($\Delta_q(t) = -1$) and action is not stay idle
            $r = -1000$
        else if queue is not empty ($\Delta_q(t) \neq -1$) and action is stay idle
            $r = -1000$
        else if queue is empty ($\Delta_q(t) = -1$) and action is stay idle
            $r = 0$
        else
            $r = -\sum_{k=1}^{n} \mathbb{1}(\Delta_r(t) > \Delta_{max})$
    Update the 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 each
```

3.2.2.1 Simulation Results

Referring to our Q-learning based adaptive blocklength selection method as QL-ABM, we demonstrate its performance in comparison to the fixed blocklength scheme.

Figure 3.11 depicts the QL-ABM performance compared with fixed blocklength schemes for varying transmit power P. Threshold Δ_{max} and packet arrival rate λ are fixed at 800 CUs and 0.01, respectively. Low transmit power implies that the probability of experiencing low SNR levels is high. Thus, it can be seen that for low SNR, the largest blocklengths achieve the smallest age violation probability among all blocklength schemes. Meanwhile, QL-ABM is clearly the most advantageous as it can dynamically select the optimal blocklength to use in each different channel realization. As P increases, small values of n provide lower P_{av}, while using large blocklength constantly becomes inefficient as higher SNR levels are frequently seen. QL-ABM also shows worse performance compared to fixed blocklength schemes with $n = 100$ and $n = 125$. This can be explained as follows: since low SNR levels are rarely experienced, it is possible that the Q-learning agent cannot learn about them thoroughly and does not know which action is optimal in the states corresponding to low SNR. This affects the age violation probability as the block error rate changes substantially according to the blocklength when SNR is low. Thus, we conclude that QL-ABM is advantageous in low transmit power, i.e., low SNR regions.

In Figure 3.12, results obtained for various packet arrival rates are shown while $P = 0$ dB and $\Delta_{max} = 800$ CUs. When $\lambda = 0.001$, P_{av} is high for all fixed blocklength schemes and QL-ABM, as packet arrival is very infrequent. With higher λ, the idle periods in which the system waits for a packet arrival are shorter. Therefore, P_{av} drops significantly in all cases. QL-ABM yields lower P_{av} than fixed blocklength schemes for the whole range of λ values, but the performance gap becomes more visible with increasing λ.

Lastly, Figure 3.13 displays P_{av} for various threshold values with fixed packet arrival rate ($\lambda = 0.01$) and transmit power ($P = 0$ dB). Similar to the results with varying λ, increasing the threshold value Δ_{max} leads to a substantial decrease in P_{av} for QL-ABM and the fixed blocklength schemes. Also, the performance of QL-ABM

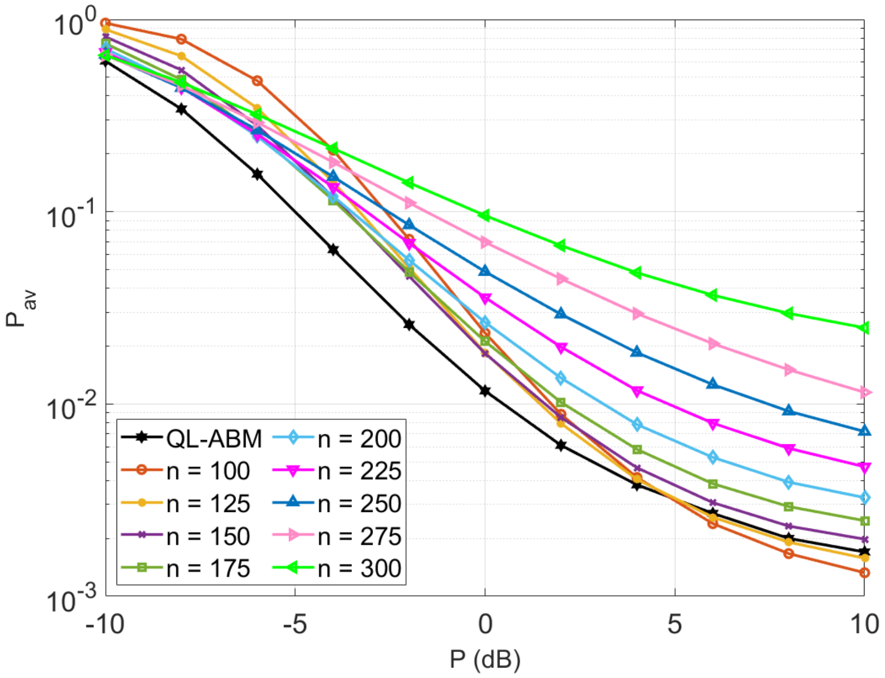


Figure 3.11: Comparison of P_{av} for QL-ABM and fixed blocklength schemes for different transmit power levels ($\lambda = 0.01$, $\Delta_{max} = 800$ CUs)

becomes superior for higher Δ_{max} values.

3.2.3 Comparison of Solution Methods

After demonstrating the performances of value iteration and Q-learning based adaptive blocklength selection methods compared to fixed blocklength schemes, in this section we compare the two solutions. As a baseline performance, we use the lowest age violation probability values obtained with fixed blocklength schemes.

Figure 3.14 shows the performances of VI-ABM and QL-ABM along with the fixed blocklength scheme for varying transmit power P. Threshold Δ_{max} and arrival rate λ are constant at 800 CUs and 0.01, respectively. It can be seen that VI-ABM achieves significantly lower P_{av} for all P values compared two the other two schemes, and QL-ABM is better than the fixed blocklength scheme for P up to around 5 dB.

In Figure 3.15, age violation probabilities of VI-ABM, QL-ABM and the fixed blocklength scheme are given for varying packet arrival rate λ. Transmit power P is 0 dB

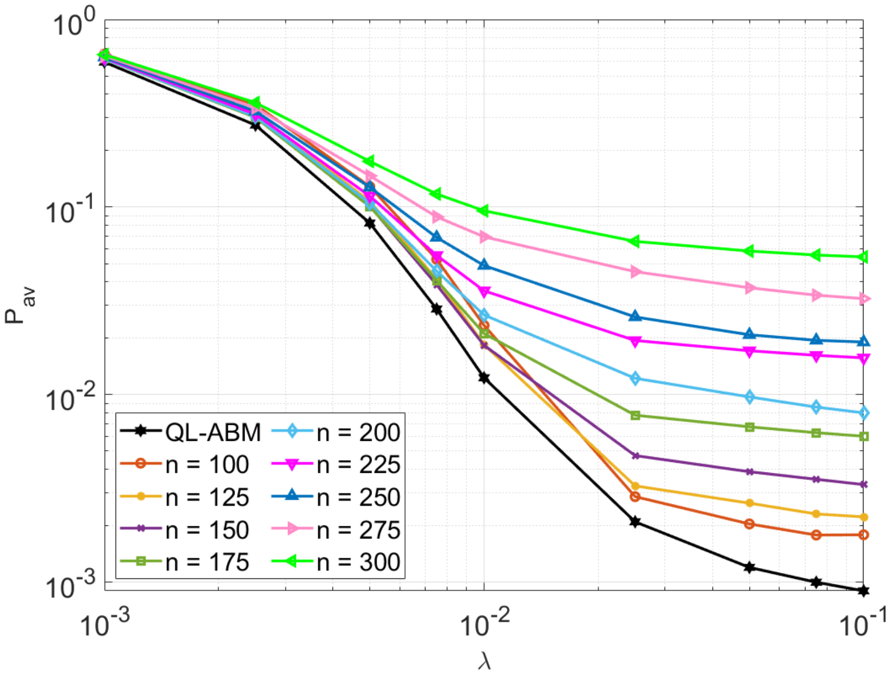


Figure 3.12: Comparison of P_{av} for QL-ABM and fixed blocklength schemes for different arrival rates ($P = 0$ dB, $\Delta_{max} = 800$ CUs)

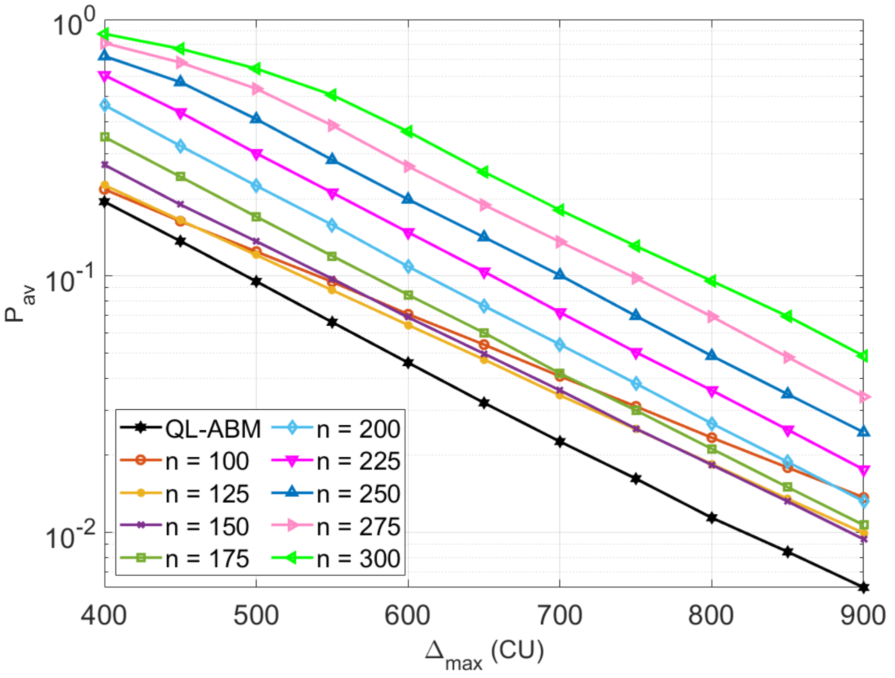


Figure 3.13: Comparison of P_{av} for QL-ABM and fixed blocklength schemes for different age thresholds ($P = 0$ dB, $\lambda = 0.01$)

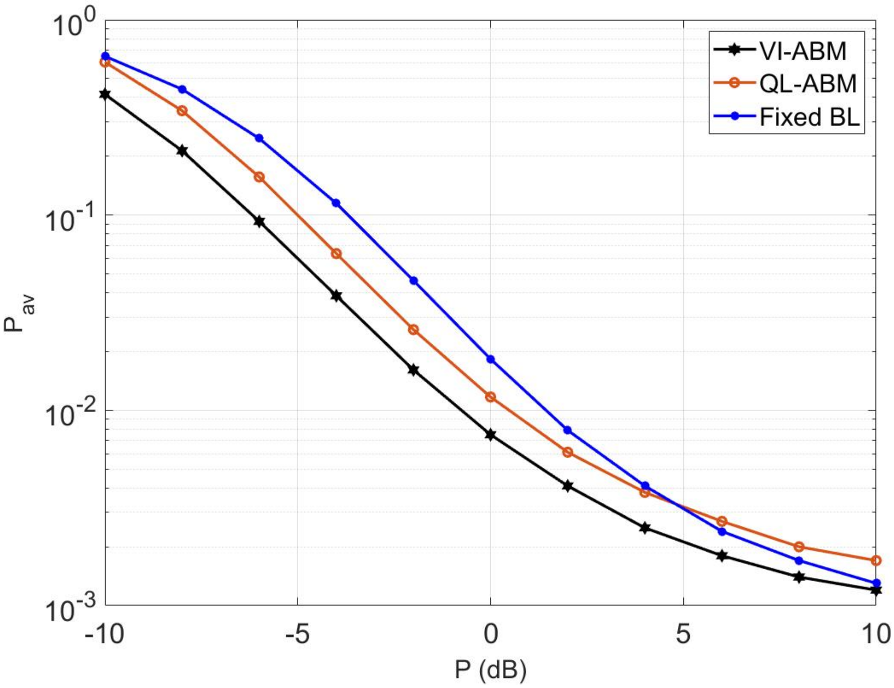


Figure 3.14: Comparison of P_{av} for VI-ABM and QL-ABM for different transmit power levels ($\lambda = 0.01$, $\Delta_{max} = 800$ CUs)

and threshold Δ_{max} is 800 CUs. Again, VI-ABM is superior to fixed blocklength and QL-ABM, and the performance gap is significant for high λ values. Although not as good as VI-ABM, QL-ABM achieves lower P_{av} than the fixed blocklength scheme for all packet arrival rates.

Lastly, Figure 3.16 depicts the performances of VI-ABM, QL-ABM, and fixed blocklength for varying threshold Δ_{max} while transmit power P and packet arrival rate λ are fixed at 0 dB and 0.01, respectively. Supporting the previous results, again the lowest age violation probability for all threshold values is achieved by VI-ABM, followed by QL-ABM.

It is clear that for all scenarios, VI-ABM is superior to QL-ABM. Nevertheless, it is important to recall that value iteration is a model-based method; hence it requires full knowledge of the environment dynamics, such as state transition probabilities and reward models. On the other hand, Q-learning learns with trial and error, as it has no prior knowledge about the environment, and suffers from the exploration-exploitation trade-off mentioned in Section 2.6. Thus, it is reasonable that VI-ABM shows better performance than QL-ABM, considering its prior knowledge and higher complexity.

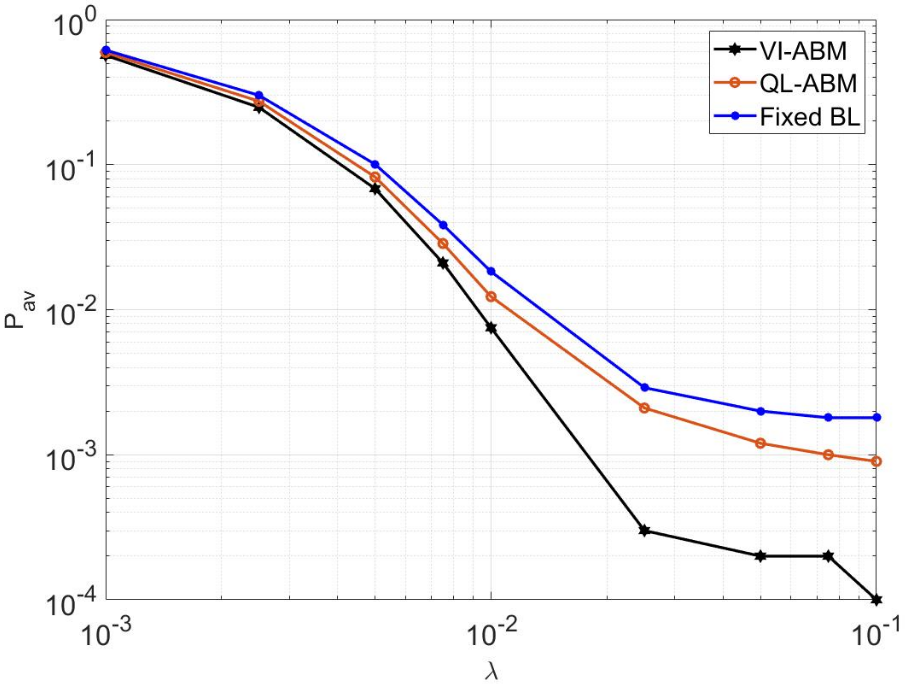


Figure 3.15: Comparison of P_{av} for VI-ABM and QL-ABM for different arrival rates ($P = 0$ dB, $\Delta_{max} = 800$ CUs)

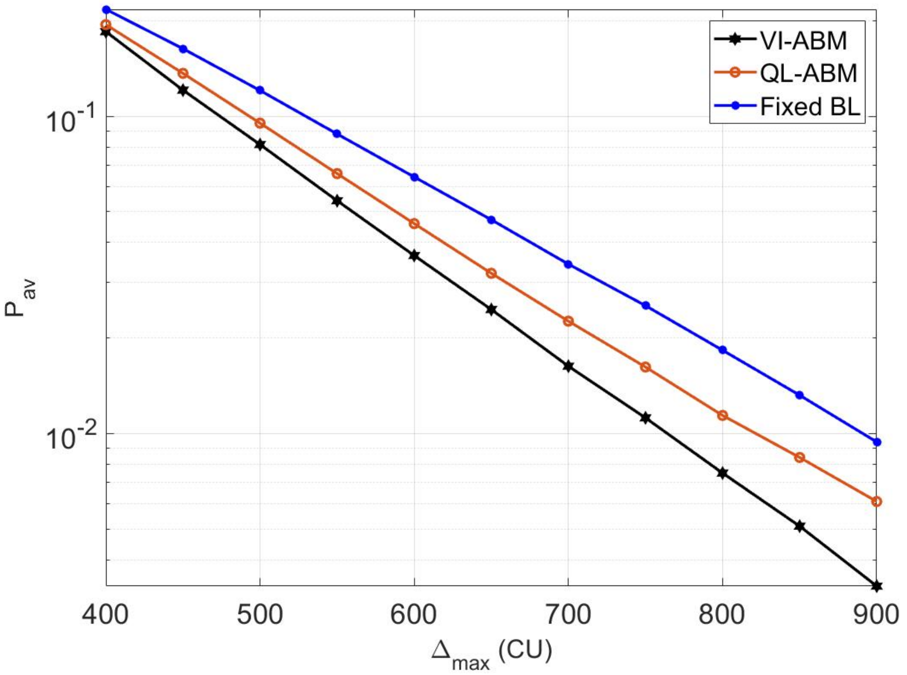


Figure 3.16: Comparison of P_{av} for VI-ABM and QL-ABM for different age thresholds ($P = 0$ dB, $\lambda = 0.01$)

CHAPTER 4

ADAPTIVE MCS SELECTION FOR MINIMIZING AGE VIOLATION PROBABILITY

In this chapter, we present the system model for the modulation and coding scheme selection problem. With a deep reinforcement learning approach, we propose a solution based on deep Q-networks to minimize the age violation probability. We demonstrate the simulation results comparing the performance of our solution with the baseline methods.

4.1 System Model

We consider a system model very similar to the one in Chapter 3, with the addition of modulation (see Figure 4.1). We have a single-server queue with capacity 2 that accepts information packets from a source. The source generates packets according to the Bernoulli distribution, where the packet arrival rate, i.e., the probability of a packet arrival in one channel use (CU), is denoted with λ. The queue follows a Last Come First Serve (LCFS) policy: A new packet arriving at an empty queue gets served immediately. However, if the queue is non-empty, the packet already in the queue is replaced with the new one.

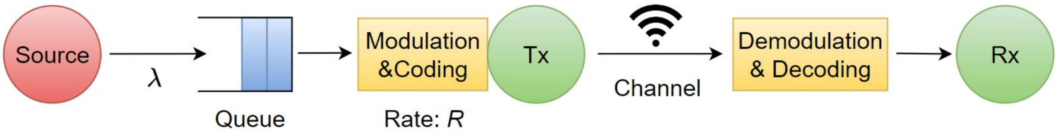
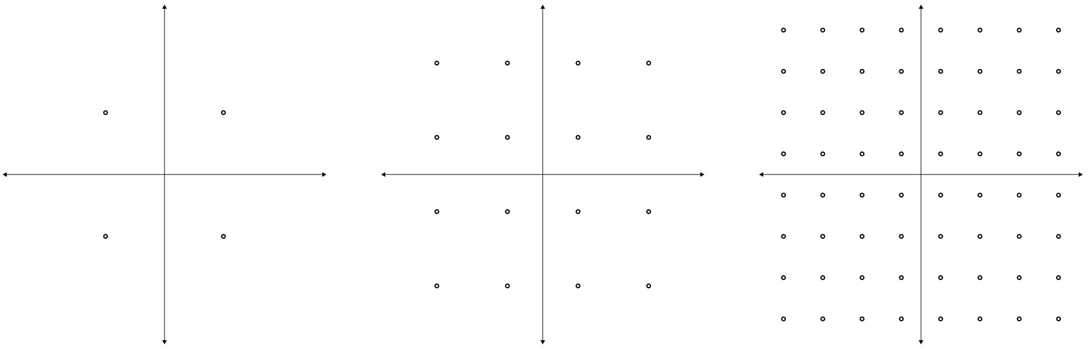


Figure 4.1: System model for the MCS selection problem

The information packet from the source consists of k bits. When it is taken to service, it is modulated and encoded according to the chosen MCS index. We use M-ary quadrature amplitude modulation (M-QAM). The number of bits transmitted in one CU with M-QAM is equal to $\log_2 M$, which is also called the *modulation order*. Thus, with M-QAM, the transmission of a packet with blocklength n takes $n/\log_2 M$ CUs.

Figure 4.2 shows the constellation diagrams for 4-QAM, 16-QAM, and 64-QAM. It can be seen that higher modulation order means the points on the constellation diagram are closer; thus, the probability of decoding error is higher [87, p.200]. So; when channel conditions are bad, lower modulation orders should be used, and vice versa. In this study, we use one of three MCS tables defined in the 5G standards. While one of the tables lists MCSs with modulation up to 256QAM, the other two tables define MCSs with 64QAM at most. In this work, we use the third table [1, Table 5.1.3.1.-3], which is used for low spectral efficiency cases. Table 4.1 lists the MCS indexes we use with the corresponding modulation orders, code rates, and spectral efficiencies.



(a) 4-QAM

(b) 16-QAM

(c) 64-QAM

Figure 4.2: M-QAM constellations for $M = 4, 16, 64$

After modulation and coding, the packet is transmitted through the wireless channel. As in Chapter 4, we assume a memoryless block-fading Rayleigh channel. Each packet transmission goes through IID fading coefficients h. For transmitted signal x, received signal y and additive noise w, the input-output relation of the channel is

$$y = x \cdot h + w. \tag{4.1}$$

Table 4.1: MCS index table [1]

MCS Index	Modulation order	Code Rate R x 1024	Spectral Efficiency
0	2	30	0.2344
1	2	40	0.3770
2	2	50	0.6016
3	2	64	0.8770
4	2	78	1.1758
5	2	99	1.4766
6	2	120	1.6953
7	2	157	1.9141
8	2	193	2.1602
9	2	251	2.4063
10	2	308	2.5703
11	2	379	2.7305
12	2	449	3.0293
13	2	526	3.3223
14	2	602	3.6094
15	4	340	3.9023
16	4	378	4.2129
17	4	434	4.5234
18	4	490	4.8164
19	4	553	5.1152
20	4	616	5.3320
21	6	438	5.5547
22	6	466	5.8906
23	6	517	6.2266
24	6	567	6.5703
25	6	616	6.9141
26	6	666	7.1602
27	6	719	7.4063
28	6	772	4.5234
29	2	Reserved	Reserved
30	4	Reserved	Reserved
31	6	Reserved	Reserved

We assume channel state information at the transmitter (CSIT) and noise with standard normal distribution ($\mathcal{N}(0, 1)$). For a transmit power P, instantaneous SNR is expressed as

$$\gamma = P|h|^2. \tag{4.2}$$

We calculate the block error rate (BLER) by rewriting Eqn. 2.12 in the following form:

$$\epsilon(\gamma) \approx Q\left(\frac{I'(\gamma, M) - \frac{k}{n}}{\sqrt{\frac{V(\gamma)}{n}}}\right), \tag{4.3}$$

where $I'(\gamma, M)$ is the approximation to the mutual information in Eqn. 2.11, and $V(\gamma)$ is the channel dispersion:

$$I'(\gamma, M) \approx \log_2 M \ \times \ \Big(1 - \sum_{j=1}^{k_M} \varepsilon_j^{(M)} e^{-\vartheta_j^{(M)}\gamma}\Big). \tag{4.4}$$

$$V(\gamma) = \frac{\gamma(\gamma + 2)}{2(\gamma + 1)^2} \ \log_2^2(e). \tag{4.5}$$

The coefficients $\varepsilon_j^{(M)}$ and $\vartheta_j^{(M)}$ in Eqn. 4.4 are given in Tables 2.1 and 2.2, respectively.

We inspect the age at the receiver, $\Delta_r(t) = t - u(t)$ where $u(t)$ is the generation time of the last packet that was delivered to the receiver without error. As shown in Figure 3.2 before, the age at the receiver increases until a packet is successfully received, then it is reduced to the age at the queue. If the transmission is unsuccessful or the system is waiting idly for a new packet arrival, $\Delta_r(t)$ increases linearly.

We aim to minimize the probability of age violation by dynamically selecting the modulation and coding scheme. We calculate the age violation probability as follows: We find the number of age violations in a sufficiently long time interval T. Then the ratio of the resulting number to the total time T gives the age violation probability:

$$\begin{aligned} P_{av}(\Delta_{max}) &= \lim_{t\to\infty} P(\Delta_r(t) > \Delta_{max}) \\ &= \lim_{t\to\infty} \frac{1}{T}\sum_{t=1}^{T} \mathbb{1}(\Delta_r(t) > \Delta_{max}), \end{aligned} \tag{4.6}$$

where $\mathbb{1}(\cdot)$ is the indicator function.

4.2 DQN-based Adaptive MCS Selection Method

The selection of modulation and coding scheme is a more complex problem compared to blocklength selection. This is because the number of actions and states is significantly larger. Thus, Q-learning fails to be a satisfactory solution, and we utilize deep Q networks. Here we consider two approaches: in the first one, the state includes CQI information, as in Chapter 3. In the second approach, however, the state includes only the ages at the queue and receiver. Actions and reward are the same for the two approaches. Denoting the state spaces of the first and second methods as $\mathcal{S}_1$ and $\mathcal{S}_2$, respectively, we define the states, actions, and reward as follows:

- $\mathcal{S}_1 = (\Delta_q, \Delta_r, CQI)$, $\mathcal{S}_2 = (\Delta_q, \Delta_r)$
  - Age of the packet in the queue ($\Delta_q(t)$): The time elapsed since the last packet arrival to the system. $\Delta_q(t) = -1$ if the queue is empty.
  - Age at the receiver ($\Delta_r(t)$): Time elapsed since the arrival of the last successfully decoded packet to the system.
  - Channel quality indicator (CQI): Instead of quantization as in Chapter 3, here we obtain the CQI simply by rounding the SNR to the nearest integer.

The evolutions of $\Delta_q(t)$ and $\Delta_r(t)$ in time are the same as in Chapter 3: 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, $\Delta_q(t)$ is reset to zero. Otherwise, it increases with unit rate. The age at the receiver $\Delta_r(t)$, on the other hand, grows until a transmission is completed successfully. Let n denote the blocklength used according the chosen MCS index, and $n = 1$ imply the action of staying idle for one CU. Then, the changes in $\Delta_q(t)$ and $\Delta_r(t)$ after n CUs can be expressed as follows:

$$\Delta_q(t{+}n) = \begin{cases} -1, & \text{no packet in the queue,} \\ j, & \text{the freshest packet in the queue arrived } j \text{ CUs ago.} \end{cases} \tag{4.7}$$

$$\Delta_r(t{+}n) = \begin{cases} \Delta_q(t) + n, & \Delta_q(t) \neq -1, n \neq 1, \text{successful transmission,} \\ \Delta_r(t) + n, & \text{otherwise.} \end{cases} \tag{4.8}$$

Unlike $\Delta_q(t)$ or $\Delta_r(t)$, the CQI state after n CUs does not depend on the previous CQI state or the other states. According to our channel model, the fading coefficient h realized in each packet transmission is an IID Rayleigh random variable. With transmit power P, the corresponding SNR is $\gamma = P|h|^2$. Since it is a rounded version of SNR, the CQI state changes randomly according to Rayleigh distribution.

- $\mathcal{A}$: The MCSs in Table 4.1, plus *stay idle* action.

- $\mathcal{R}$: Here, we use a different reward function than Chapter 4. In each iteration, we count the number of age violations because of the selected action. However, this is not a sufficient solution: The reward of applying an action a is the same whether $\Delta_r(t)$ 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 and vice versa. Again, rewards corresponding to these cases are large negative values. On the other hand, the reward of choosing to stay idle when queue is empty is zero, as it is the optimal action to take in that state. We follow the same notation as Chapter 3 here, where a_0 means staying idle. Then, the reward function is expressed as

$$\mathcal{R}_s^a = \begin{cases} -5000, & \Delta_q(t) = -1 \ \ \& \ \ a \neq a_0, \\ -5000, & \Delta_q(t) \neq -1 \ \ \& \ \ a = a_0, \\ 0, & \Delta_q(t) = -1 \ \ \& \ \ a = a_0, \\ -\sum_{k=1}^{n} \mathbb{1}(\Delta_r(t) > \Delta_{max}) & \\ \quad + \max(0, \Delta_r(t) - \Delta_{max}), & \text{otherwise.} \end{cases} \tag{4.9}$$

For both DQN-based solutions, we construct a deep Q network with three layers: the input and output layers, and a hidden layer. An ε-greedy policy with a decaying exploration rate is followed. As the loss function, we use Huber loss [88]:

$$L_\delta(y, f(x)) = \begin{cases} \frac{1}{2}(y - f(x))^2, & \text{for} \leq \delta, \\ \delta(|y - f(x)| - \frac{1}{2}\delta), & \text{otherwise.} \end{cases} \tag{4.10}$$

Eqn. 4.10 states that if the loss value is less than δ, Huber loss is equal to the *mean squared error (MSE)*; however, for loss values greater than δ, 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.

The algorithm for our DQN based adaptive MCS selection method is described in detail in Algorithm 5.

4.3 Baseline Solutions

For assessing the performance of our DQN-AMC solutions, we compare them with two baseline methods: inner loop link adaptation (ILLA) and outer loop link adaptation (OLLA) [55]. ILLA is a basic method used for adaptive MCS selection based on a fixed lookup table. Given the current SNR, ILLA selects an MCS that satisfies the

Algorithm 5 DQN-based Adaptive MCS Selection

```
Initialize replay memory
Initialize main network $Q$ with random weights $\theta$
Initialize target network $\hat{Q}$ with random weights $\theta^-$
for each episode do
    Initialize state $s$ = ($\Delta_q(t)$, $\Delta_r(t)$, CQI)
    for each step do
        Select an action $a$ following an $\varepsilon$-greedy policy: select an MCS index or stay idle for 1 CU
        Execute action $a$ and observe state $s'$
        Receive reward $r$:
            if queue is empty ($\Delta_q(t) = -1$) and action is not stay idle
                $r = -5000$
            else if queue is not empty ($\Delta_q(t) \neq -1$) and action is stay idle
                $r = -5000$
            else if queue is empty ($\Delta_q(t) = -1$) and action is stay idle
                $r = 0$
            else
                $r = -\sum_{k=1}^{n} \mathbb{1}(\Delta_r(t) > \Delta_{max}) + \max(0, \Delta_r(t) - \Delta_{max})$
        Store transition $(s, a, r, s')$ in replay memory
        Sample random 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 the loss $L_\delta(y_j, Q(s, a; \theta))$
        Update target network at every $N$ episodes
    end
end
```

target block error rate (BLER) requirement. While it is straightforward, ILLA fails to be an efficient solution: The variations in the wireless channel along with delays and quantization errors cause instability in the measured SNR. In such cases, OLLA is applied in addition to ILLA for improvement. In OLLA, based on the positive or negative acknowledgment (ACK/NACK) about the transmitted packet, an offset Δ_{olla} is used to adjust the measured SNR γ, and the MCS is selected from the lookup table according to the resulting value of SNR γ_{olla}:

$$\gamma_{olla} = \gamma - \Delta_{olla}. \tag{4.11}$$

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

$$\Delta_{olla} \leftarrow \Delta_{olla} + \Delta_{up} \cdot \mathbb{1}_{nack} - \Delta_{down} \cdot \mathbb{1}_{ack}, \tag{4.12}$$

where $\mathbb{1}(\cdot)$ is the indicator function and Δ_{up} and Δ_{down} are the *step up* and *step down* parameters, related to each other in terms of the target BLER denoted as $BLER_T$:

$$\Delta_{down} = \frac{\Delta_{up}}{\frac{1}{BLER_T} - 1}. \tag{4.13}$$

OLLA algorithm is also given in detail in Algorithm 6.

Algorithm 6 Outer Loop Link Adaptation (OLLA)

```
Algorithm parameters: $\Delta_{up}, \Delta_{down}$
Initialize offset to zero ($\Delta_{olla} = 0$)
At each transmission
    **if** ACK **then**
        $\Delta_{olla} \leftarrow \Delta_{olla} - \Delta_{down}$
    **else**
        $\Delta_{olla} \leftarrow \Delta_{olla} + \Delta_{up}$
**end**
$\gamma_{olla} = \gamma - \Delta_{olla}$
$MCS = MCS(\gamma_{olla})$
```

Table 4.2: Simulation parameters of DQN-based adaptive MCS selection method

Parameter	**Value**
Number of information bits (k)	256
Transmit power (P)	0 dB
Packet arrival rate (λ)	0.005
Age threshold (Δ_{max})	5000 CUs
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

4.4 Simulation Results

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 $BLER_T$ to 10^{-1}. We name our solutions DQN-AMC-1 and DQN-AMC-2. In DQN-AMC-1, CQI is included in the state, whereas in DQN-AMC-2 the state consists of the ages at the queue and the receiver. Table 4.2 summarizes the simulation parameters.

Figure 4.3 shows the age violation probability P_{av} of different schemes for various transmit power levels P. Age threshold Δ_{max} and arrival rate λ are fixed at 5000 CUs and is 0.005, respectively. When P is low, the probability of having bad channel conditions is higher; thus, the frequently seen SNR values are low, and P_{av} is heavily influenced by erroneous transmissions. As ILLA and OLLA schemes use low MCS indexes to achieve the target BLER, age violation probability is high because of the large blocklengths, so the DQN-AMC schemes provide lower P_{av}. As P in-

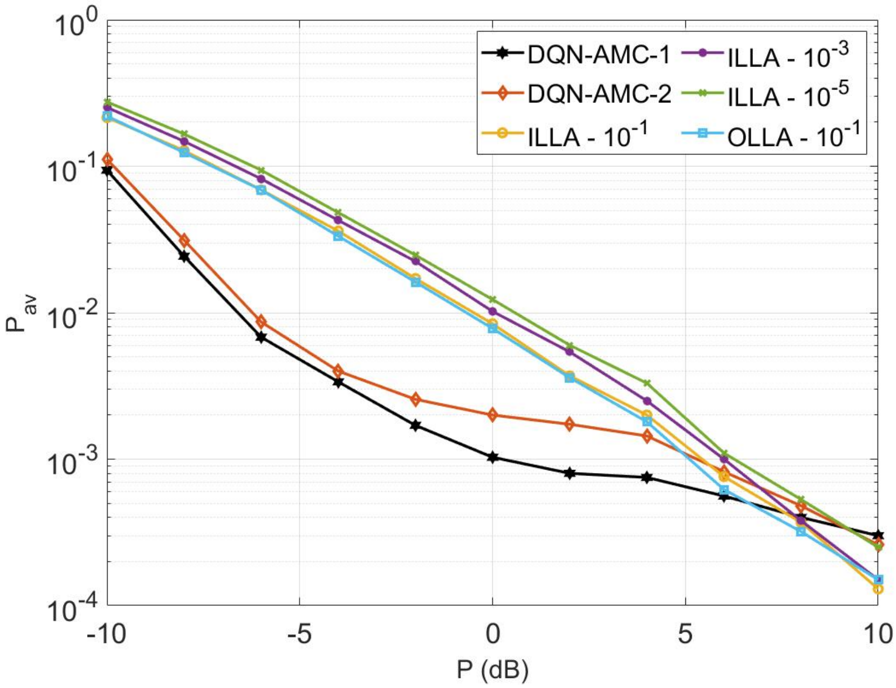


Figure 4.3: Comparison of P_{av} for DQN-AMC, ILLA and OLLA methods for different transmit power levels ($\Delta_{max} = 5000$ CUs, $\lambda = 0.005$)

creases, 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. It is notable that while the ILLA schemes have similar performances, as $BLER_T$ of ILLA goes from 10^{-1} to 10^{-5} the age violation probability 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 have a significant effect on 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 in terms of the number of states, it stands as a feasible solution.

Figure 4.4 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 high exploration rate. Therefore, it cannot fully learn the optimal actions for when the queue is non-empty. Increasing λ to about 0.005 leads to a

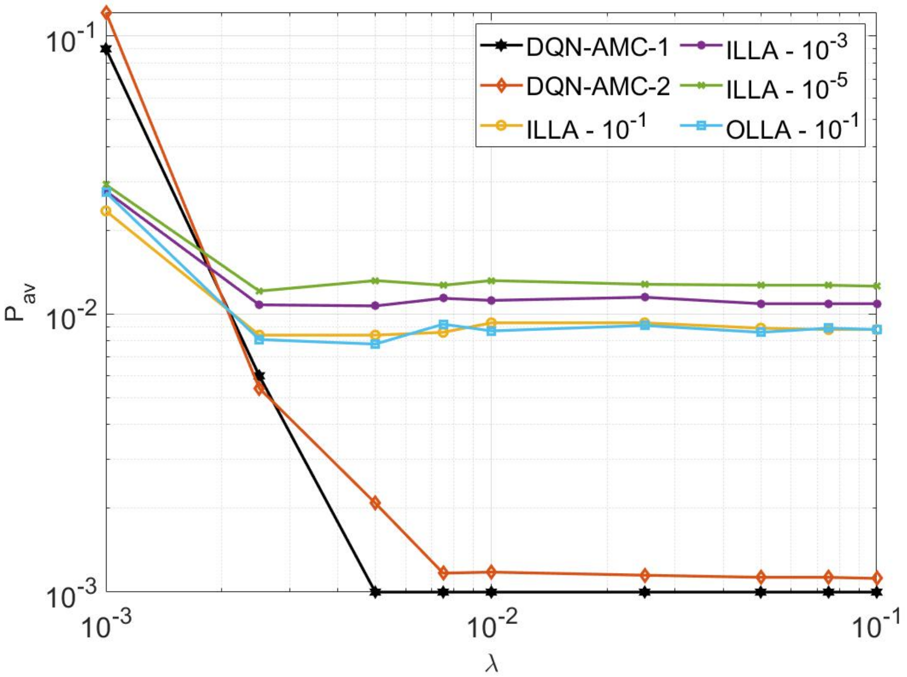


Figure 4.4: Comparison of P_{av} for DQN-AMC, ILLA and OLLA methods for different arrival rates ($P = 0$ dB, $\Delta_{max} = 5000$ CUs)

substantial reduction of P_{av} in all schemes, but the difference is much higher for DQN-AMC schemes. For λ values above 0.005, changes in P_{av} become negligible for all schemes. As in the previous results, ILLA with $BLER_T = 0.1$ and OLLA perform very similary, and for ILLA with smaller target BLER, we observe higher P_{av}.

In Figure 4.5, P_{av} is plotted for different age thresholds Δ_{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 P_{av} than DQN-AMC-2 for almost all threshold values. Consistent with the previous results, ILLA scheme with $BLER_T = 10^{-5}$ has the highest age violation probability, and the difference between the ILLA schemes are visible. Again the OLLA scheme improves the performance negligibly. As the threshold increases, the probability of age violation is reduced for all schemes.

To conclude, we can say that our DQN-AMC methods achieve lower age violation probabilities for most of the test scenarios. DQN-AMC-1, which includes CQI in-

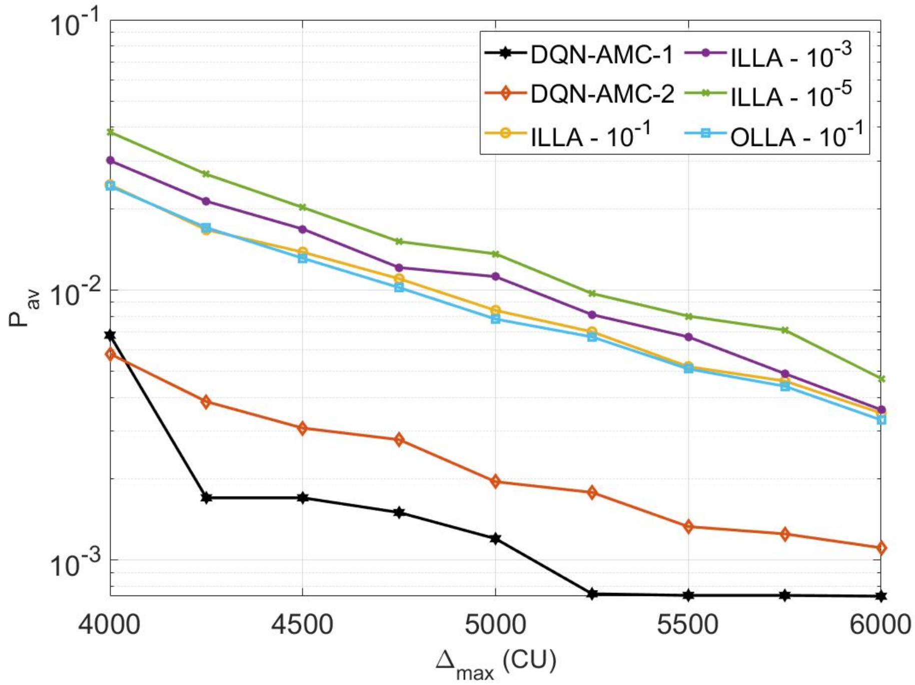


Figure 4.5: Comparison of P_{av} for DQN-AMC, ILLA and OLLA methods for different thresholds ($P = 0$ dB, $\lambda = 0.005$)

formation in the state, generally performs better than DQN-AMC-2. 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.

CHAPTER 5

CONCLUSIONS AND FUTURE WORK

In this thesis, our main aim is to minimize the age violation probability by dynamically choosing the blocklength and the modulation and coding scheme (MCS) in a short packet transmission framework. Firstly, we provide a detailed background on Age of Information (AoI) and finite blocklength (FBL) theory and review the reinforcement learning (RL), dynamic programming (DP), and deep RL (DRL) methods we use in this study.

In Chapter 3, we define our first problem, which focuses on blocklength selection in FBL regime. Our first solution to the blocklength selection problem is based on a state-aggregated value iteration method. We show that this solution provides age violation probability much lower than optimal fixed blocklength schemes in different scenarios such as varying transmit power, packet arrival rate, and age threshold. Our second solution based on Q-learning also shows superior performance in various scenarios compared to fixed blocklength schemes, although it could not achieve age violation probabilities as low as value iteration based method. Nevertheless, considering that it assumes no prior knowledge on the system characteristics and has lower complexity, Q-learning based solution has a satisfactory performance.

In Chapter 4, we address the adaptive MCS selection problem with a deep reinforcement learning (DRL) approach. Utilizing finite blocklength approximations and deep Q networks (DQN), we exploit policies for choosing the appropriate MCS among the MCSs defined in 5G standards. Compared to the baseline solutions, namely, inner loop link adaptation (ILLA) and outer loop link adaptation (OLLA), our DQN based adaptive MCS selection methods yield lower age violation probability in various scenarios.

In this thesis, we have shown that the solutions we have proposed for blocklength and MCS selection based on dynamic programming and reinforcement learning are efficient and promising for optimizing semantic communication metrics. In future work, we suggest that the methods proposed in this thesis can be extended to more complex systems, such as more realistic 5G environments with multiple users. The flexible numerology and frame structure of 5G communications can be utilized. A cell-free network with multiple distributed access points can be considered. The channel model can be time-correlated or slow fading, and a long-term adaptation method can be formed. Instead of direct transmission, relay schemes can be studied. For status update generation, a generate-at-will model can be considered, in which the source can generate status updates at any time. Also, exploiting retransmission schemes such as hybrid automatic repeat request (HARQ) in order to improve the performance can be considered. In addition to blocklength/MCS selection, dynamic resource allocation and adaptive power control can also be investigated.

REFERENCES

[1] 3GPP, “NR; physical layer procedures for data,” tech. rep., June 2022.

[2] R. D. Yates, Y. Sun, D. R. Brown, S. K. Kaul, E. Modiano, and S. Ulukus, “Age of information: An introduction and survey,” *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 5, pp. 1183–1210, 2021.

[3] S. Kaul, M. Gruteser, V. Rai, and J. Kenney, “Minimizing age of information in vehicular networks,” in *Sensor, Mesh and Ad-Hoc Communications and Networks (SECON), 2011 8th Annual IEEE Communications Society Conference*, pp. 350–358, June 2011.

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

[5] C. Bockelmann, N. Pratas, H. Nikopour, K. Au, T. Svensson, C. Stefanovic, P. Popovski, and A. Dekorsy, “Massive machine-type communications in 5G: physical and mac-layer solutions,” *IEEE Communications Magazine*, vol. 54, no. 9, pp. 59–65, 2016.

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

[7] J. Park, S. Samarakoon, H. Shiri, M. Abdel-Aziz, T. Nishio, A. Elgabli, and M. Bennis, “Extreme URLLC: Vision, challenges, and key enablers,” 01 2020.

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

[9] E. Uysal, O. Kaya, A. Ephremides, J. Gross, M. Codreanu, P. Popovski, M. Assaad, G. Liva, A. Munari, T. Soleymani, *et al.*, “Semantic communications in networked systems,” *arXiv preprint arXiv:2103.05391*, 2021.

[10] M. Kountouris and N. Pappas, "Semantics-empowered communication for networked intelligent systems," *IEEE Communications Magazine*, vol. 59, no. 6, pp. 96–102, 2021.

[11] R. Talak and E. Modiano, "Age-delay tradeoffs in single server systems," in *2019 IEEE International Symposium on Information Theory (ISIT)*, pp. 340–344, 2019.

[12] 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 INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications*, pp. 1–9, 2016.

[13] C. E. Shannon, "A mathematical theory of communication," *The Bell System Technical Journal*, vol. 27, no. 3, pp. 379–423, 1948.

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

[15] Y. Polyanskiy, H. V. Poor, and S. Verdu, "Channel coding rate in the finite block-length regime," *IEEE Trans. on Inf. Theory*, vol. 56, no. 5, pp. 2307–2359, 2010.

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

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

[18] Y. Wang, S. Wu, D. Li, J. Jiao, and Q. Zhang, "Age-optimal IR-HARQ design in the presence of non-trivial propagation delay," in *2019 11th International Conference on Wireless Communications and Signal Processing (WCSP)*, pp. 1–6, 2019.

[19] P. Parag, A. Taghavi, and J.-F. Chamberland, "On real-time status updates over symbol erasure channels," in *2017 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1–6, 2017.

[20] C. M. Wijerathna Basnayaka, D. N. K. Jayakody, T. D. Ponnimbaduge Perera, and M. Vidal Ribeiro, “Age of information in a URLLC-enabled decode-and-forward wireless communication system,” in *2021 IEEE 93rd Vehicular Technology Conference (VTC2021-Spring)*, pp. 1–6, 2021.

[21] M. Xie, J. Gong, and X. Ma, “Age-energy tradeoff in dual-hop status update systems with the m-th best relay selection,” in *2021 IEEE 93rd Vehicular Technology Conference (VTC2021-Spring)*, pp. 1–5, 2021.

[22] Z. Zhang, X. Zhu, Y. Jiang, J. Cao, and Y. Liu, “Closed-form AoI analysis for dual-queue short-block transmission with block error,” in *2021 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1–6, 2021.

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

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

[25] B. Yu, Y. Cai, D. Wu, and Z. Xiang, “Average age of information in short packet based machine type communication,” *IEEE Trans. on Vehicular Technology*, vol. 69, no. 9, pp. 10306–10319, 2020.

[26] H. Sung, M. Kim, and J. Lee, “Impact of finite blocklength on AoI violation probability in UAV networks,” in *2020 International Conference on Information and Communication Technology Convergence (ICTC)*, pp. 248–250, 2020.

[27] W. Cheng, Y. Xiao, S. Zhang, and J. Wang, “Adaptive finite blocklength for ultra-low latency in wireless communications,” *IEEE Transactions on Wireless Communications*, vol. 21, no. 6, pp. 4450–4463, 2022.

[28] J. Cao, X. Zhu, Y. Jiang, Z. Wei, and S. Sun, “Information age-delay correlation and optimization with finite block length,” *IEEE Transactions on Communications*, vol. 69, no. 11, pp. 7236–7250, 2021.

[29] 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 *2019 IEEE Globecom Workshops (GC Wkshps)*, pp. 1–6, 2019.

[30] B. Han, Z. Jiang, Y. Zhu, and H. D. Schotten, "Recursive optimization of finite blocklength allocation to mitigate age-of-information outage," in *2020 IEEE International Conference on Communications Workshops (ICC Workshops)*, pp. 1–6, 2020.

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

[32] E. T. Ceran, D. Gündüz, and A. György, "A reinforcement learning approach to age of information in multi-user networks," in *2018 IEEE 29th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, pp. 1967–1971, 2018.

[33] E. T. Ceran, D. Gündüz, and A. György, "A reinforcement learning approach to age of information in multi-user networks with HARQ," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 5, pp. 1412–1426, 2021.

[34] E. Sert, C. Sönmez, S. Baghaee, and E. Uysal-Biyikoglu, "Optimizing age of information on real-life TCP/IP connections through reinforcement learning," in *26th Signal Process. and Commun. App. Conf. (SIU)*, pp. 1–4, 2018.

[35] H. B. Beytur and E. Uysal, "Age minimization of multiple flows using reinforcement learning," in *2019 International Conference on Computing, Networking and Communications (ICNC)*, pp. 339–343, 2019.

[36] X. Chen, C. Wu, T. Chen, H. Zhang, Z. Liu, Y. Zhang, and M. Bennis, "Age of information aware radio resource management in vehicular networks: A proactive deep reinforcement learning perspective," *IEEE Transactions on Wireless Communications*, vol. 19, no. 4, pp. 2268–2281, 2020.

[37] E. T. Ceran, D. Gündüz, and A. György, "Average age of information with hybrid ARQ under a resource constraint," in *2018 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1–6, 2018.

[38] M. A. Abd-Elmagid, H. S. Dhillon, and N. Pappas, “A reinforcement learning framework for optimizing age of information in RF-powered communication systems,” *IEEE Transactions on Communications*, vol. 68, no. 8, pp. 4747–4760, 2020.

[39] S. Leng and A. Yener, “An actor-critic reinforcement learning approach to minimum age of information scheduling in energy harvesting networks,” in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 8128–8132, 2021.

[40] M. Hatami, M. Leinonen, and M. Codreanu, “AoI minimization in status update control with energy harvesting sensors,” *IEEE Transactions on Communications*, vol. 69, no. 12, pp. 8335–8351, 2021.

[41] 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 INFOCOM 2019 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pp. 656–661, 2019.

[42] J. Hu, H. Zhang, L. Song, R. Schober, and H. V. Poor, “Cooperative internet of UAVs: Distributed trajectory design by multi-agent deep reinforcement learning,” *IEEE Transactions on Communications*, vol. 68, no. 11, pp. 6807–6821, 2020.

[43] M. Yi, X. Wang, J. Liu, Y. Zhang, and B. Bai, “Deep reinforcement learning for fresh data collection in UAV-assisted IoT networks,” in *IEEE INFOCOM 2020 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pp. 716–721, 2020.

[44] W. Li, L. Wang, and A. Fei, “Minimizing packet expiration loss with path planning in UAV-assisted data sensing,” *IEEE Wireless Communications Letters*, vol. 8, no. 6, pp. 1520–1523, 2019.

[45] 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 *2019 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6, 2019.

[46] 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 *IEEE Globecom Workshops (GC Wkshps)*, pp. 1–6, 2019.

[47] 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 *2012 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 809–814, 2012.

[48] K. Zhou, “Robust cross-layer design with reinforcement learning for IEEE 802.11n link adaptation,” in *2011 IEEE International Conference on Communications (ICC)*, pp. 1–5, 2011.

[49] S. Tripathi, C. Puligheddu, and C. F. Chiasserini, “An RL approach to radio resource management in heterogeneous virtual RANs,” in *2021 16th Annual Conference on Wireless On-demand Network Systems and Services Conference (WONS)*, pp. 1–8, 2021.

[50] S. Jamshidiha, V. Pourahmadi, A. Mohammadi, and M. Bennis, “Link-level throughput maximization using deep reinforcement learning,” *IEEE Networking Letters*, vol. 2, no. 3, pp. 101–105, 2020.

[51] S. Mashhadi, N. Ghiasi, S. Farahmand, and S. M. Razavizadeh, “Deep reinforcement learning based adaptive modulation with outdated CSI,” *IEEE Communications Letters*, vol. 25, no. 10, pp. 3291–3295, 2021.

[52] W. Xu, S. Guo, S. Ma, H. Zhou, M. Wu, and W. Zhuang, “Augmenting drive-thru internet via reinforcement learning-based rate adaptation,” *IEEE Internet of Things Journal*, vol. 7, no. 4, pp. 3114–3123, 2020.

[53] S. Wu, G. Tsoukaneri, and B. Mouhouche, “Q-learning based link adaptation in 5G,” in *2020 IEEE 31st Annual International Symposium on Personal, Indoor and Mobile Radio Communications*, pp. 1–6, 2020.

[54] 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 *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*, pp. 1466–1475, 2019.

[55] 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 *2007 IEEE 66th Vehicular Technology Conference*, pp. 1792–1796, 2007.

[56] M. G. Sarret, D. Catania, F. Frederiksen, A. F. Cattoni, G. Berardinelli, and P. Mogensen, “Dynamic outer loop link adaptation for the 5G centimeter-wave concept,” in *Proceedings of European Wireless 2015; 21th European Wireless Conference*, pp. 1–6, 2015.

[57] F. Blanquez-Casado, G. Gomez, M. d. C. Aguayo-Torres, and J. T. Entrambasaguas, “eOLLA: an enhanced outer loop link adaptation for cellular networks,” *EURASIP Journal on Wireless Communications and Networking*, vol. 2016, no. 1, pp. 1–16, 2016.

[58] A. Özkaya and E. T. Ceran, “Minimizing age violation probability with adaptive blocklength selection in short packet transmissions,” in *2022 30th Signal Processing and Communications Applications Conference (SIU)*, pp. 1–4, 2022.

[59] R. D. Yates, E. Najm, E. Soljanin, and J. Zhong, “Timely updates over an erasure channel,” in *2017 IEEE International Symposium on Information Theory (ISIT)*, pp. 316–320, 2017.

[60] R. D. Yates and S. K. Kaul, “The age of information: Real-time status updating by multiple sources,” *IEEE Transactions on Information Theory*, vol. 65, no. 3, pp. 1807–1827, 2019.

[61] E. T. Ceran, D. Gündüz, and A. György, “Average age of information with hybrid ARQ under a resource constraint,” in *2018 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1–6, 2018.

[62] I. Krikidis, “Average age of information in wireless powered sensor networks,” *IEEE Wireless Communications Letters*, vol. 8, no. 2, pp. 628–631, 2019.

[63] H. B. Beytur, S. Baghaee, and E. Uysal, “Measuring age of information on real-life connections,” in *2019 27th Signal Processing and Communications Applications Conference (SIU)*, pp. 1–4, 2019.

[64] M. Costa, M. Codreanu, and A. Ephremides, “On the age of information in status update systems with packet management,” *IEEE Transactions on Information Theory*, vol. 62, no. 4, pp. 1897–1910, 2016.

[65] J. Östman, R. Devassy, G. Durisi, and E. Uysal, “Peak-age violation guarantees for the transmission of short packets over fading channels,” in *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pp. 109–114, 2019.

[66] 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 of Things Journal*, vol. 8, no. 14, pp. 11329–11344, 2021.

[67] 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 Journal on Selected Areas in Commun.*, vol. 37, no. 4, pp. 721–734, 2019.

[68] X. Zheng, S. Zhou, Z. Jiang, and Z. Niu, “Closed-form analysis of non-linear age of information in status updates with an energy harvesting transmitter,” *IEEE Transactions on Wireless Communications*, vol. 18, no. 8, pp. 4129–4142, 2019.

[69] 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 *2019 IEEE Symposium on Computers and Communications (ISCC)*, pp. 1–6, 2019.

[70] M. Song, H. H. Yang, H. Shan, J. Lee, H. Lin, and T. Q. S. Quek, “Analysis of AoI violation probability in wireless networks,” in *2021 17th International Symposium on Wireless Communication Systems (ISWCS)*, pp. 1–6, 2021.

[71] G. Durisi, T. Koch, J. Östman, Y. Polyanskiy, and W. Yang, “Short-packet communications over multiple-antenna rayleigh-fading channels,” *IEEE Transactions on Communications*, vol. 64, no. 2, pp. 618–629, 2016.

[72] W. Yang, G. Durisi, T. Koch, and Y. Polyanskiy, “Quasi-static multiple-antenna

fading channels at finite blocklength," *IEEE Transactions on Information Theory*, vol. 60, no. 7, pp. 4232–4265, 2014.

[73] H.-M. Wang, Q. Yang, Z. Ding, and H. V. Poor, "Secure short-packet communications for mission-critical IoT applications," *IEEE Transactions on Wireless Communications*, vol. 18, no. 5, pp. 2565–2578, 2019.

[74] P. Mary, J.-M. Gorce, A. Unsal, and H. V. Poor, "Finite blocklength information theory: What is the practical impact on wireless communications?," in *2016 IEEE Globecom Workshops (GC Wkshps)*, pp. 1–6, 2016.

[75] 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, 2022.

[76] C. Ouyang, S. Wu, C. Jiang, J. Cheng, and H. Yang, "Approximating ergodic mutual information for mixture gamma fading channels with discrete inputs," *IEEE Communications Letters*, vol. 24, no. 4, pp. 734–738, 2020.

[77] D. Silver, "Lectures on reinforcement learning." URL: `https://www.davidsilver.uk/teaching/`, 2015.

[78] M. L. Puterman, *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

[79] L. P. Kaelbling, M. L. Littman, and A. W. Moore, "Reinforcement learning: A survey," *Journal of artificial intelligence research*, vol. 4, pp. 237–285, 1996.

[80] R. Bellman, *Dynamic Programming*. Princenton University Press, 1957.

[81] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[82] Y. Li, "Deep reinforcement learning: An overview," *arXiv preprint arXiv:1701.07274*, 2017.

[83] V. François-Lavet, P. Henderson, R. Islam, M. G. Bellemare, J. Pineau, *et al.*, "An introduction to deep reinforcement learning," *Foundations and Trends® in Machine Learning*, vol. 11, no. 3-4, pp. 219–354, 2018.

[84] T. Marzetta and B. Hochwald, “Capacity of a mobile multiple-antenna communication link in Rayleigh flat fading,” *IEEE Transactions on Information Theory*, vol. 45, no. 1, pp. 139–157, 1999.

[85] W. Yang, G. Durisi, T. Koch, and Y. Polyanskiy, “Block-fading channels at finite blocklength,” in *ISWCS; 10th Int. Symp. on Wireless Commun. Syst.*, pp. 1–4, 2013.

[86] S. Singh, T. Jaakkola, and M. Jordan, “Reinforcement learning with soft state aggregation,” *Adv. in Neural Inf. Process. Syst.*, vol. 7, 1994.

[87] J. Proakis and M. Salehi, *Digital Communications*. McGraw-Hill, 2008.

[88] P. J. Huber, “Robust estimation of a location parameter,” in *Breakthroughs in statistics*, pp. 492–518, Springer, 1992.