Joint Adaptive Transmission and Numerology Selection for 5G NR PDSCH with DQN-Based Reinforcement Learning Solution

by

Thet Naung San

B.Eng., Mandalay Technological University, Myanmar, 2012

A Project Submitted in Partial Fulfillment of the Requirements for the Degree of

Master of Engineering

in the Department of Electrical and Computer Engineering

University of Victoria

Joint Adaptive Transmission and Numerology Selection for 5G NR PDSCH with DQN-Based Reinforcement Learning Solution

by

Thet Naung San

B.Eng., Mandalay Technological University, Myanmar, 2012

Supervisory Committee

Prof. Hong-Chuan Yang, Supervisor

(Department of Electrical and Computer Engineering)

Prof. Daler Rakhmatov, Departmental Member

(Department of Electrical and Computer Engineering)

Abstract

The mission critical applications such as industrial automation, remote surgery and autonomous transportation systems demand low-latency, high-reliability communications service. As such, there is an urgent need to optimize transmission technologies in 5G New Radio (NR) to support Ultra-Reliable Low-Latency Communication (URLLC).

This project introduces a joint adaptive transmission and numerology selection scheme for Physical Downlink Shared Channel (PDSCH) in 5G NR, targeting URLLC support. The transmission scheme selection problem is modeled as a Markov Decision Process (MDP). A Deep Q-Network (DQN) reinforcement learning agent is trained to dynamically adjust Modulation and Coding Scheme (MCS) and numerology based on real-time channel conditions and latency constraints.

To evaluate the performance, we develop custom simulation environment by implementing PDSCH transmission model under frequency-selective fading channels, incorporating the Hybrid Automatic Repeat reQuest (HARQ) mechanism. The results demonstrate that the DQN agent effectively reduces transmission delays and improves reliability by optimizing transmission parameters. This approach enhances performance for 5G NR in URLLC support, achieving both higher reliability and lower latency than conventional adaptive transmission system.

Table of Contents

List of Tables

List of Figures

Acknowledgements

I would like to thank my supervisor, Dr. Hong Chuan Yang, expressing my deepest gratitude for his unwavering encouragement, support, and guidance during my master's studies. Furthermore, I would like to thank Dr. Daler Rakhmatov, my supervisory committee member, for his valuable support in the development of the project.

List of Acronyms

5G NR	5th Generation New Radio
AMC	Adaptive Modulation and Coding
AWGN	Additive White Gaussian Noise
CQI	Channel Quality Indicator
CP	Cyclic Prefix
CP-OFDM	Cyclic Prefix-Orthogonal Frequency Division Multiplexing
CSI-RS	Channel State Information Reference Signals
DM-RS	Demodulation Reference Signals
DQN	Deep Q-Network
DL-SCH	Downlink Shared Channel
FFT	Fast Fourier Transform
HARQ	Hybrid Automatic Repeat reQuest
IFFT	Inverse Fast Fourier Transform
IIoT	Industrial Internet of Things
LDPC	Low-Density Parity-Check
MAC	Media Access Control
MCS	Modulation and Coding Scheme
MDP	Markov Decision Process
MIMO	Multiple Input Multiple Output
OFDM	Orthogonal Frequency Division Multiplexing
PDSCH	Physical Downlink Shared Channel
PRB	Physical Resource Block

QAM	Quadrature Amplitude Modulation
RL	Reinforcement Learning
SNR	Signal-to-Noise Ratio
SCS	Subcarrier Spacing
TB	Transport Block
TBS	Transport Block Size
TDL	Tapped Delay Line
TTI	Transmission Time Interval
URLLC	Ultra-Reliable Low Latency Communications
eMBB	Enhanced Mobile Broadband
mMTC	Massive Machine-Type Communication

Chapter 1

Introduction

1.1 Background and Motivation

The 5G New Radio (NR) standard introduces revolutionary improvements in wireless communications, offering higher data rates, improved capacity and lower latency. One of the main areas of focus is the Physical Downlink Shared Channel (PDSCH), which is used for transmitting user data in 5G systems. For Ultra-Reliable Low Latency Communications (URLLC) use case, the PDSCH must deliver data with minimal latency and high reliability. Achieving this requires the optimal use of 5G NR features, such as Adaptive Modulation and Coding scheme (AMC), Hybrid Automatic Repeat reQuest (HARQ) mechanism and flexible numerology.

Traditionally, modulation and coding schemes (MCS) was selected using predefined static thresholds based on channel quality indicators. While this approach is effective under stable conditions, it often fails to optimize performance in highly dynamic 5G environments. To improve the performance, HARQ techniques are introduced for error correction and packet retransmission. However, these traditional techniques have difficulty meeting the stringent latency requirements of URLLC, especially when dealing with variable and unpredictable channel conditions in real-time transmission.

This project is motivated by the need for more effective solution to support URLLC use case in 5G NR PDSCH systems. We propose to jointly select MCS and numerology according to factors such as data packet size, channel conditions, latency constraints and the number of HARQ retransmission attempts. Such adaptive approach has the potential to improve transmission efficiency, reduce packet loss, and meet the stringent requirements of URLLC applications. We formulate the problem of sequential transmission scheme and parameter selection in a Markov decision process (MDP) and train a Deep Q-Network (DQN)-based Reinforcement Learning (RL) agent to learn the optimal policy.

1.2 Objectives

The main objective of this project is to develop a Deep Q-Network (DQN)-based framework to optimize the joint selection of Modulation and Coding Scheme (MCS) and numerology in 5G NR for Physical Downlink Shared Channel (PDSCH) transmission. The specific objectives are:

1. PDSCH transmission model implementation: Develop and implement a PDSCH transmission model using MATLAB 5G Toolbox to simulate realistic 5G NR transmission scenarios.
2. Adaptive MCS and Numerology selection: Formulation of Markov decision process (MDP), development of DQN solution to obtain the optimal policy. Design and train a DQN agent capable of dynamically adjusting MCS and numerology based on real-time channel conditions, latency requirements, and number of HARQ retransmissions attempts.
3. Minimize packet loss: Reduce packet loss by intelligently managing HARQ retransmissions and optimizing transmission parameters under varying channel conditions.
4. Meet URLLC latency requirements: Achieve reliable transmission within the strict latency limits required for URLLC applications.
5. Improve transmission efficiency: Improve the entire transmission process by selecting the most efficient combination of MCS and numerology, resulting in better transmission reliability and data rate.

By achieving these goals, the project aims to create a robust and intelligent 5G NR PDSCH transmission system that can adapt to the dynamics of modern wireless environments, particularly benefiting applications that require high reliability and low latency.

1.3 Structure of the report

This project report is divided into six chapters and covers the key aspects of adaptive MCS and numerology selection in 5G NR using reinforcement learning. Chapter 2 provides an overview of 5G NR, focusing on modulation and coding schemes, numerologies and subcarrier spacing. Chapter 3 explains the DL-SCH and PDSCH transmission models and covers the transmission process, transport block size and HARQ mechanisms. Chapter 4 describes the DQN-based reinforcement learning approach for adaptive transmission, including the MDP formulation, DQN solution, and environment setup. Chapter 5 presents simulation results and analyzes the performance of DQN agent. Finally, Chapter 6 summarizes the results and suggests future research directions.

Chapter 2

5G New Radio

2.1 5G NR Technology Overview

The evolution of mobile communications technology from 1G to 5G and the upcoming 6G is rapidly advancing, bringing transformative changes that improve everyday life. With increasing expectations for faster and more reliable connected systems, the mobile industry is undergoing a major transformation that extends beyond smartphones and impacts sectors such as automotive, manufacturing, entertainment, healthcare and more. At the heart of this transformation is 5G NR, a revolutionary radio access technology designed to enable significant advances across a wide range of services and deliver higher quality and improved performance to all users.

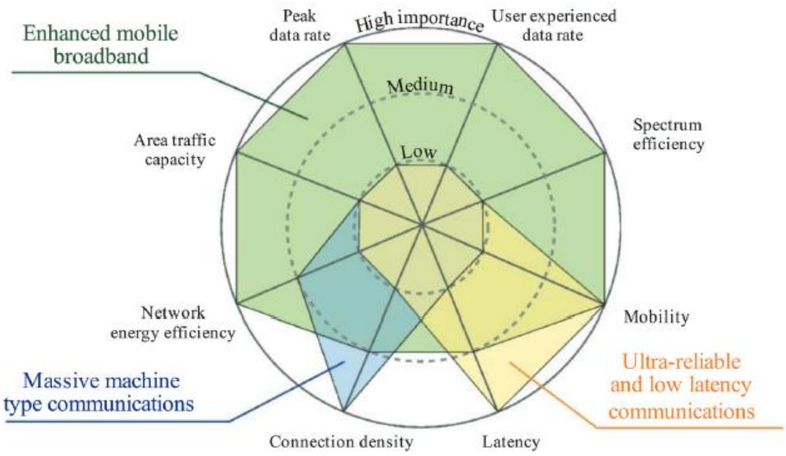

Figure 2.1: NR Service Categories [1]

5G NR services are divided into three main use cases, each addressing different aspects of communication requirements: Enhanced Mobile Broadband (eMBB), Ultra-Reliable Low Latency Communications (URLLC) and Massive Machine-Type Communications (mMTC) [1, 2]. These categories, as shown in Fig. 2.1, outline the key areas in which 5G NR excels:

- eMBB focuses on delivering faster data rates and improving user experience in data-intensive environments such as streaming, gaming, and virtual reality.
- URLLC is aimed at supporting applications that require real-time communications with extremely high reliability and minimal latency, which is critical for automated industries, smart cities, and mission-critical operations such as autonomous driving.
- mMTC facilitates the connection of large numbers of devices, enabling the Internet of Things (IoT) to transform industries by enabling machines to communicate and work in intelligent, connected environments.

These different service categories illustrate the flexibility and scalability of 5G NR, which is designed to meet the growing needs of modern communications networks and open the door to a variety of innovative applications [3].

2.2 Modulation and Coding Scheme (MCS)

The Modulation and Coding Scheme (MCS) is a key component of the 5G NR system and enables flexible and efficient data transmission across different channel conditions. MCS refers to the combination of a modulation scheme and a coding scheme that determines how data is transmitted over the wireless channel [12–14]. The selection of the MCS is dynamic and depends on the real-time quality of the channel, as measured by the Channel Quality Indicator (CQI) to ensure a balance between data throughput and reliability. CQI is a metric used by the receiver to inform the transmitter about the channel quality. Higher CQI values indicate better channel conditions, allowing the transmitter to use higher modulation and coding schemes for faster data rates. The modulation scheme defines how many bits can be carried by one symbol, while the coding scheme determines the percent of redundancy added for error correction applied during transmission.

The modulation schemes used in 5G NR include Quadrature Phase Shift Keying (QPSK), 16-Quadrature Amplitude Modulation (16-QAM), 64-QAM and 256-QAM. These schemes differ in their modulation order, as we can see in Table 2.1, and which defines how many bits each symbol represents:

- QPSK: Modulates 2 bits per symbol, offering low data rates with high robustness against noise and interference.
- 16QAM: Modulates 4 bits per symbol, providing a balance between data rate and noise resistance.
- 64QAM: Modulates 6 bits per symbol, supporting higher data rates with less resistance to noise.
- 256QAM: Modulates 8 bits per symbol, achieving the highest data rates with minimal robustness.

Table 2.1: Supported Modulation and Coding Scheme [12]

MCS Index I_{MCS}	Modulation Order Q_m	Modulation	Code Rate
1	2	QPSK	0.076
2	2	QPSK	0.188
3	2	QPSK	0.438
4	4	16QAM	0.369
5	4	16QAM	0.478
6	4	16QAM	0.602
7	6	64QAM	0.455
8	6	64QAM	0.554
9	6	64QAM	0.650
10	6	64QAM	0.754
11	6	64QAM	0.853
12	8	256QAM	0.694
13	8	256QAM	0.778
14	8	256QAM	0.864
15	8	256QAM	0.926

The coding rate in 5G NR is determined by the Channel Quality Indicator (CQI), which measures the quality of the wireless channel. It represents the ratio of the number of useful bit to the total transmitted bit (including useful bit and error correction bit) in a transmission block. The coding rate for 5G NR is typically between 0.0769 and 0.9688 [14], with lower values indicate greater error correction and higher values indicate less error correction. A lower coding rate implies that a larger portion of the transmitted bits is reserved for error correction, which is beneficial in poor channel conditions but comes at the expense of lower data throughput. Conversely, a higher coding rate uses fewer error correction bits, increasing data throughput but requiring better channel quality to maintain reliability.

2.3 5G Numerologies and Frame Structure

In 5G NR, numerology refers to the set of parameters that define the structure of the radio frame and subframes. These parameters include subcarrier spacing, slot duration, and symbol duration, which are crucial for configuring the physical layer of the 5G NR system. 5G NR supports OFDM (Orthogonal Frequency Division Multiplexing) flexible numerology, which adjusts the subcarrier spacing and affects the time division of each subframe. This flexibility enables 5G to support a wide range of use cases, from eMBB to URLLC and mMTC.

5G subcarrier spacing (SCS) is the distance between two adjacent subcarriers in an OFDM signal. It is a key parameter that affects the bandwidth and performance of a 5G network. SCS is determined by the formula $\Delta f = 15 \times 2^{\mu}$ kHz, where Δf represents subcarrier spacing, μ represents the numerology index and $\mu \in \{0,1,2,3,4,5\}$ [4-7]. Smaller subcarrier spacing is advantageous in scenarios that require larger coverage areas and longer transmission durations, as they provide better robustness to noise and interference. Larger subcarrier spacings are used for higher frequencies such as millimeter wave (mmWave) communications [2], where shorter transmission times are required to address problems such as phase noise and Doppler shifts.

The choice of numerology affects the transmission time interval (TTI) and cyclic prefix length, allowing optimization of latency, spectral efficiency or robustness based on the specific use case or environment.

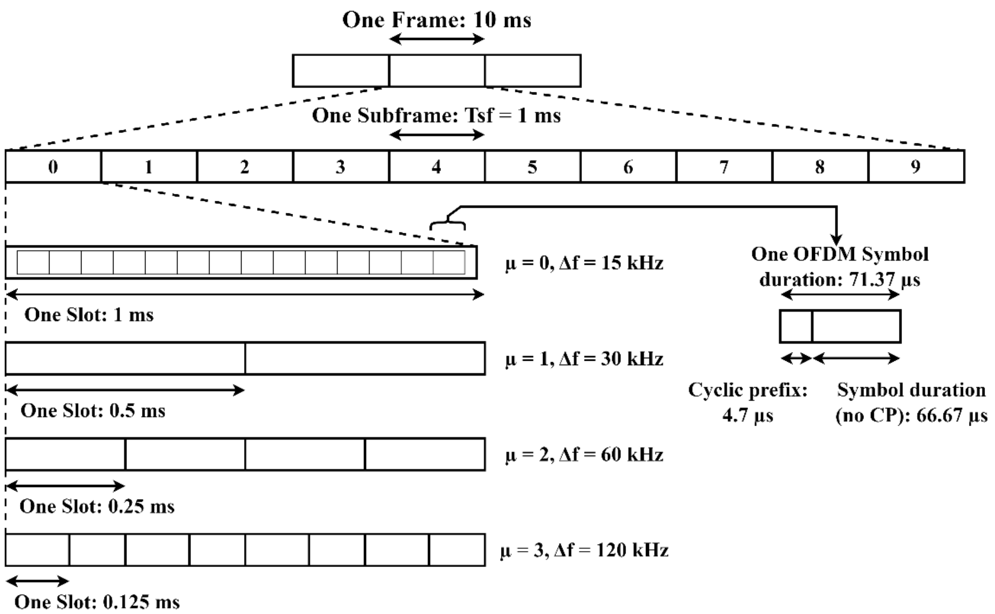

Figure 2.2: 5G NR Frame Structure

The 5G NR framework is extremely flexible and scalable and covers a wide range of services, frequencies and deployment scenarios [4, 8]. The heart of 5G NR's time structure is the frame, which serves as the basic unit of time. Each frame has a fixed duration of 10 milliseconds (ms) and is divided into 10 subframes with a duration of 1 ms each. Each subframe is further divided into slots, and the number of slots per subframe depends on the numerology and subcarrier spacing. In each slot, the number of OFDM symbols is typically fixed at 14 symbols per slot for normal the Cyclic prefix (CP) configuration [12]. However, the length of each subframe T_{sf} can vary depending on numerology and slot size. The subframe duration is calculated using the following formula [12, 14]:

$$T_{sf} = \frac{N_{sym}}{14 \times 2^{\mu}}\ ms \tag{2.1}$$

where N_{sym} is the number of OFDM symbols per slot. Fig. 2.2 illustrates the 5G NR frame structure and shows how the subframe duration and number of slots per subframe change with different numerologies and slot sizes.

Table 2.2 summarizes the number of OFDM symbols per slot, the number of slots per subframe, and the number of slots per frame for different numerologies:

Table 2.2: Numerology Index and Subcarrier Spacing [12]

Numerology (μ)	Subcarrier Spacing (kHz) $\Delta F = 15 \times 2^{\mu}$	No. of OFDM Symbols per Slot (N_{symb}^{slot})	No. of Slots per Subframe ($N_{slot}^{subframe,\mu}$)	No. of Slots per Frame ($N_{slot}^{frame,\mu}$)
0	15kHz	14	1	10
1	30kHz	14	2	20
2	60kHz	14	4	40
3	120kHz	14	8	80

2.4 Subcarrier Spacing and Frequency Range

In 5G New Radio (NR), there are two main frequency ranges have been specified by 3GPP: Frequency Range 1 (FR1) and Frequency Range 2 (FR2). FR1, also known as the sub-6 GHz range, operates between 0.45 GHz and 6 GHz, while FR2, commonly referred to as the millimeter wave (mmWave) range, is from 24 GHz to 52.6 GHz, as shown in Table 2.3. These frequency ranges are critical in determining the maximum bandwidth and subcarrier spacing used for data transmission. In FR1, the maximum bandwidth is 100 MHz, while FR2 supports a much larger maximum bandwidth of 400 MHz, enabling faster data rates and lower latency [11–14].

Table 2.3: Definition of frequency ranges [11]

Frequency range destination	Corresponding frequency range
FR1	0.45 GHz - 6 GHz
FR2	24 GHz - 52.6 GHz

Table 2.4 describes the flexibility that allows 5G NR to adapt to different deployment scenarios. Balancing coverage, spectral efficiency and latency depending on selected frequency range and subcarrier spacing.

Table 2.4: Variable Subcarrier Spacing, Symbol Duration and Frequency Range

Subcarrier spacing (kHz)	Frequency	Symbol duration (no CP) (µs)	Symbol duration (with CP) (µs)	Nominal Max BW (MHz)	Min Scheduling Interval (ms)
15	FR1	66.7	71.35	49.5	1
30	FR1	33.3	35.68	99	0.5
60	FR1, FR2	16.6	17.84	198	0.25
120	FR2	8.33	8.92	396	0.125
240	N/A (SSB only, No Data)	4.17	4.46	397.4	0.0625

Resource blocks (RBs) serve as basic units of bandwidth allocation, with their configuration dependent on both channel bandwidth and subcarrier spacing. The 3GPP standard [11] specifies the maximum transmission bandwidth configuration for each UE channel and subcarrier spacing, as shown in Table 2.5 and Table 2.6. This configuration defines how many RBs can be allocated within a given channel bandwidth depending on the selected SCS, allowing the network to adapt resource allocation to different service requirements.

Table 2.5: Maximum Transmission Bandwidth configuration (in RBs) for FR1

SCS (kHz)	5 MHz	10 MHz	15 MHz	20 MHz	25 MHz	50 MHz	40 MHz	50 MHz	60 MHz	70 MHz	80 MHz	90 MHz	100 MHz
	N_{RB}	N_{RB}	N_{RB}	N_{RB}	N_{RB}	N_{RB}	N_{RB}	N_{RB}	N_{RB}	N_{RB}	N_{RB}	N_{RB}	N_{RB}
15	25	52	79	106	133	160	216	270	N/A	N/A	N/A	N/A	N/A
30	11	24	38	51	65	78	106	133	162	189	217	245	273
60	N/A	11	18	24	31	38	51	65	79	93	107	121	135

Table 2.6: Maximum Transmission Bandwidth configuration (in RBs) for FR2

SCS (kHz)	50 MHz	100 MHz	200 MHz	400 MHz
	N_{RB}	N_{RB}	N_{RB}	N_{RB}
60	66	132	264	N/A
120	32	66	18	24

Chapter 3

DL-SCH and PDSCH Transmission Model

3.1 DL-SCH and PDSCH Transmit and Receive Processing

In the 5G NR communications system, the Downlink Shared Channel (DL-SCH) and the Physical Downlink Shared Channel (PDSCH) play a central role in the efficient and reliable transmission of data between transmitter and receiver. The DL-SCH is responsible for preparing the data before transmission, such as segmentation and encoding. This data is then mapped to the PDSCH, which transmits the user data over the radio interface, making it the primary channel for downlink data transmission in 5G NR [19, 20].

The processing of DL-SCH and PDSCH is designed to maximize the reliability and efficiency of data transmission, even under difficult channel conditions such as noise, fading and interference. The DL-SCH ensures that the data is encoded and segmented, while the PDSCH facilitates the actual transmission of this data over the wireless medium. On the receiver side, a series of demodulation, decoding, error detection and error correction processes work together to reconstruct the original data with high accuracy and ensure that the communication system meets the stringent requirements of modern wireless applications. This integrated processing chain is essential to achieve the high data rates, low latency and reliable communications which are the characteristics of 5G NR.

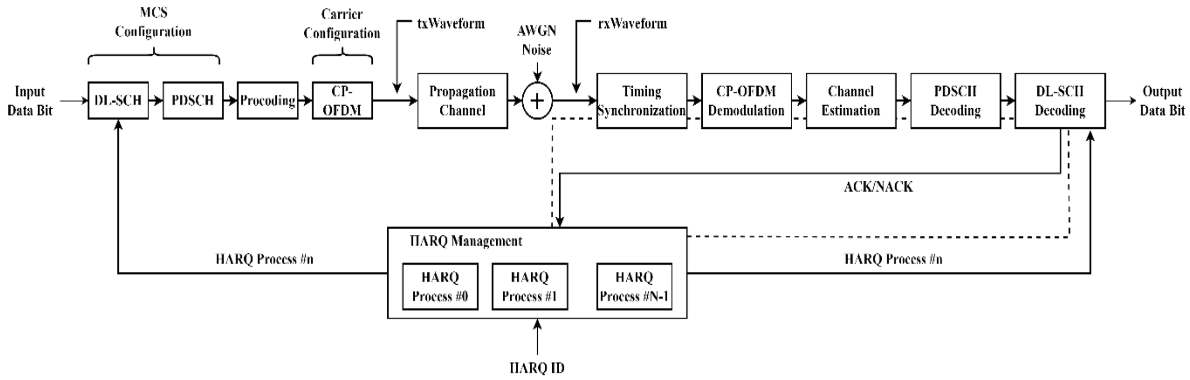

Figure 3.1: DL-SCH and PDSCH Transmit and Receive Process Diagram

The process diagram describes the end-to-end flow of data transmission and reception in a 5G NR PDSCH system, as shown in Fig. 3.1, focusing on the way data is managed and processed at different stages. It begins with the DL-SCH, where data is prepared through encoding and segmentation before being mapped to the PDSCH for transmission. The signal then undergoes precoding and CP-OFDM modulation to prepare it for propagation over the wireless channel. Upon reaching the receiver, the signal potentially affected by noise and channel interference, undergoes time synchronization, demodulation, and channel estimation. The PDSCH is then decoded, which involving demodulation, layer demapping, soft bit conversion, rate-matching, and low-density parity-check (LDPC) decoding to correct errors. The original data is reconstructed in the DL-SCH decoding step. When errors are detected, the HARQ process manages retransmissions and ensures reliable communication by combining retransmitted data with previously received data to correct errors and ultimately achieve robust and efficient data delivery in 5G networks.

3.2 Downlink Shared Channel (DL-SCH)

The Downlink Shared Channel (DL-SCH) is a crucial element of the 5G NR physical layer and is responsible for the reliable transmission of user data from the transmitter to the receiver. As the primary channel for the transmission of user plane data, broadcast messages and various control information, the DL-SCH is central to the functioning of 5G communication systems. It plays a critical role in ensuring that user data is transmitted efficiently and reliably over the wireless interface, adapting to the dynamic nature of 5G environments.

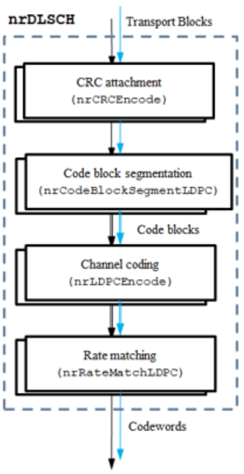

Figure 3.2: Downlink Shared Channel (DL-SCH) Process Diagram [21]

DL-SCH processing begins with the attachment of cyclic redundancy check (CRC) for error detection which is needed during the decoding process. Large transport blocks are segmented into smaller code blocks, each with its own CRC, and then encoded with low-density parity-check (LDPC) codes, which are known for their strong error correction capabilities. The *nrDLSCH* function in MATLAB handles CRC appending, code block segmentation, LDPC encoding and rate matching, ensuring that the data meets 5G NR standards, as shown in Fig. 3.2 [21].

The encoded data is adapted to the available radio resources through rate adaptation and is also supported by Hybrid Automatic Repeat reQuest (HARQ) processes to manage the retransmission of erroneous data. HARQ improves decoding accuracy by combining retransmissions with previously received data, improving reliability. These mechanisms enable the DL-SCH to meet the high data rate and low latency requirements of modern 5G applications.

3.3 Transport Block (TB) and Transport Block Size (TBS)

The transport block (TB) is the basic unit of data prepared for transmission over the air interface. It is generated at the Media Access Control (MAC) layer and processed by the physical layer within a Transmission Time Interval (TTI). Each TB goes through several important processing steps, including CRC (Cyclic Redundancy Check) application, segmentation into code blocks, and LDPC (Low-Density Parity Check) encoding for error correction. The processed TB is then mapped to physical resources for transmission over the Physical Downlink Shared Channel (PDSCH) to ensure data integrity and efficient resource utilization.

The Transport Block Size (TBS) defines the amount of data transferred in a single TB and is crucial for optimizing resource utilization in 5G NR. The TBS is determined by factors such as the number of physical resource blocks allocated for transmission, the modulation order (bits per symbol), the coding rate and additional overhead such as control information. The *nrTBS* function in MATLAB handles this calculation internally, adapting the TBS to the available resources and ensuring compliance with 5G NR standards [22]. The TBS is determined using the following equation:

$$TBS = \left\lfloor \frac{(PRB \times N_{RE}^{PRB} \times Q_m \times R) + O_H}{C} \right\rfloor \times C \tag{3.1}$$

where, *PRB* is the number of physical resource blocks allocated for transmission and can be calculated as follows:

$$PRB = \left\lfloor \frac{Bandwidth\ (Hz)}{Subcarrrier\ Spacing\ (Hz) \times N_{SC}^{PRB}} \right\rfloor \tag{3.2}$$

where N_{SC}^{PRB} is the number of subcarriers per *PRB* which corresponds to 12 subcarriers in 5G NR, N_{RE}^{PRB} is the number of resource elements (*REs*) per *PRB* available for data transfer and can be calculated as follows:

$$N_{RE}^{PRB} = N_{SC}^{PRB} \times N_{symb}^{slot} - N_{DMRS-OH}^{PRB} \tag{3.3}$$

where $N_{DMRS-OH}^{PRB}$ is the number of *REs* required for the DM-RS overhead is reserved (typically occupies 2 OFDM symbols for one transmission layer) [14], Q_m is the modulation order, i.e. the number of bits per symbol, *R* is the coding rate, which represents the proportion of useful data bits in the total number of bits transmitted, O_H is the overhead associated with the transmission, which can include various protocol-specific elements that reduce the effective data rate, *C* is a constant scaling factor that ensures that the TBS matches the granularity specified in the 5G NR standard. This constant *C* is processed internally by the *nrTBS* function.

3.4 Physical downlink shared channel (PDSCH)

The Physical Downlink Shared Channel (PDSCH) is a key element of the 5G NR physical layer and is responsible for transmitting user data and control information from the transmitter to the receiver. It transmits encoded data from the Downlink Shared Channel (DL-SCH), enabling high data rates and low latency for various 5G applications.

PDSCH uses Orthogonal Frequency Division Multiplexing (OFDM) for efficient spectrum utilization and robustness against interference. Physical resource blocks are dynamically allocated based on real-time channel conditions and user requests, optimizing network efficiency. The MATLAB function *nrPDSCHConfig* configures key PDSCH parameters such as resource block allocation, modulation schemes, and MIMO settings while supporting Hybrid Automatic Repeat reQuest (HARQ) for error correction through retransmissions [19, 20, 22].

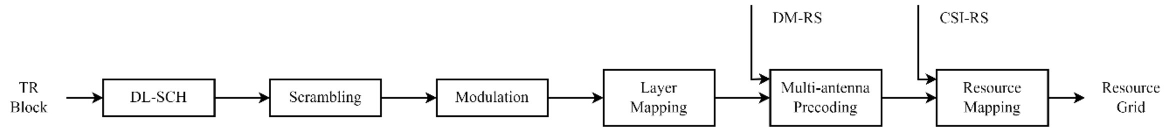

Figure 3.3: Physical downlink shared channel (PDSCH) Process Diagram

The PDSCH processing chain includes encoding, scrambling, modulation, and multi-antenna precoding as shown in Fig 3.3. The data is then mapped to the resource grid for transmission, along with reference signals such as demodulation reference signals (DM-RS) and channel state information reference signal (CSI-RS), ensuring reliable and efficient communications in a variety of 5G use cases, from Enhanced Mobile Broadband (eMBB) to Ultra-reliable Low-Latency Communications (URLLC).

3.5 Precoding

Precoding is an important step in the PDSCH transmission process. It optimizes signals for channel conditions to improve transmission quality and reduce interference. The process begins with generating PDSCH symbols from the encoded transport blocks and mapping them to modulation symbols. Precoding weights determined from channel state information are then applied to adjust the phase and amplitude of these symbols. The precoding process can be represented as follows:

$$X_{Precoded} = W_{N\_Antenna \times N\_Layer} \times X_{PDSCH\_sym} \tag{3.4}$$

where, $X_{Precoded}$ is the vector of pre-coded symbols ready for transmission, $W_{N\_Antenna \times N\_Layer}$ is the precoding matrix designed based on the channel state information, and X_{PDSCH_sym} represents the vector of PDSCH symbols [19, 20].

After precoding, demodulation reference signals (DM-RS) are generated for channel estimation at the receiver, enabling accurate decoding. In MIMO configurations, PDSCH and DM-RS symbols are assigned to resource grids based on the number of transmit antennas to ensure efficient and reliable transmission. This process ensures robust communication over the air interface, even in complex channel conditions.

3.6 CP-OFDM Modulation and Propagation Channel Model

In this phase, the modulated and precoded PDSCH symbols are converted into a time domain waveform using Orthogonal Frequency Division Multiplexing (OFDM) modulation. The process begins by applying an Inverse Fast Fourier Transform (IFFT) to the frequency domain resource grid and converts it to a time domain signal. A cyclic prefix (CP) is then added to each OFDM symbol to mitigate intersymbol interference (ISI) caused by multipath propagation in the wireless channel. The OFDM modulated waveform is represented as follows:

[txWaveform, waveformInfo] = nrOFDMModulate(carrier, pdschGrid) (3.5)

where *txWaveform* is the time domain signal carrying the data and *waveformInfo* contains details about the waveform, such as the sampling rate and the number of subcarriers [19, 20].

After modulation, the OFDM signal passes through the propagation channel, which simulates the wireless environment between transmitter and receiver and models various real-world conditions such as multipath fading, path loss and noise. In this project, we use the Tapped Delay Line (TDL) model, specifically the TDL-C channel profile, to accurately represent frequency-selective and time-varying multipath propagation.

In our transmission model, noise is introduced using Additive White Gaussian Noise (AWGN) to simulate real-world interference. To assess the quality of the transmission channel, the signal-to-noise ratio (SNR) is calculated in [20, 23]. The SNR (in dB) is related to the noise power as follows:

$$SNR = 10^{\left(\frac{SNRdB}{10}\right)} \tag{3.6}$$

$$N_0 = \frac{1}{\sqrt{N_{RxAnts} \times N_{FFT} \times SNR}} \tag{3.7}$$

where SNRdB is the signal-to-noise ratio in dB, N_{RxAnts} is the number of receive antennas, N_{FFT} is the FFT size used in OFDM demodulation, N_0 is the noise power [20].

The AWGN noise is generated as:

$$noise = N_0 \times \mathcal{N}(0,1) \tag{3.8}$$

where $\mathcal{N}(0,1)$ represents a random Gaussian distribution (mean of 0, variance of 1) applied to the waveform. This noise is then added to the received waveform:

$$rxWaveform = txWaveform + noise \tag{3.9}$$

After the signal has passed through the channel, the valid samples are available at the receiver for accurate demodulation and decoding of the transmitted data. The combination of OFDM modulation and careful handling of propagation channel effects ensures that the transmitted signal maintains its integrity and is robust to the challenges of the wireless environment, enabling reliable communication between transmitter and receiver.

3.7 Reception and Demodulation

In this phase, time synchronization is crucial to ensure that the received signal matches the correct symbol boundaries. In simulations, this can be achieved through two approaches: perfect and practical synchronization.

Perfect synchronization, performed with the *nrPerfectTimingEstimate* function, uses full channel knowledge to calculate the optimal timing offset by detecting the strongest multipath component. On the other hand, practical synchronization, implemented using the *nrTimingEstimate* function, involves cross-correlation of the received signal with PDSCH DM-RS symbols. To maintain accuracy in the presence of noise or fading, the *hSkipWeakTimingOffset* function uses previous timing estimates when the correlation is weak.

After synchronization, the signal is converted back to the frequency domain through OFDM demodulation using the *nrOFDMDemodulate* function. This process restores the resource grid containing the received symbols, which are essential for further processing and decoding. After demodulation, channel estimation is performed to compensate for distortions caused by the propagation channel. In simulations, this can be done using two methods: a perfect channel estimate via the *nrPerfectChannelEstimate* [20] function, which provides an idealized channel estimate, or a practical channel estimate using DM-RS symbols to accurately reflect real-world conditions by averaging and interpolating the channel in over time.

In our system, the signal-to-noise ratio (SNR) for each resource block slot (RB) is a crucial metric for evaluating signal quality in the presence of noise. The SNR_{RB} is calculated by determining and the average power of the received symbols for each RB (*rbSymbols*) and the noise power (*rbNoise*) [23-25]. The SNR_{RB} is then calculated as the ratio of signal power to noise power, expressed in decibels (dB) as:

$$SNR_{RB} = 10 \times log_{10}\left(\frac{P_{signal}}{P_{noise}}\right) \tag{3.10}$$

These processes ensure that the transmitted data is received and decoded accurately and high data integrity is maintained despite the challenges of the wireless environment.

3.8 Hybrid Automatic Repeat reQuest (HARQ)

Hybrid Automatic Repeat reQuest (HARQ) is an essential error correction technique in 5G NR that increases data transmission reliability by combining Forward Error Correction (FEC) with retransmission requests. If the receiver detects uncorrectable errors, it sends a negative acknowledgment (NACK) to the sender, causing the erroneous data to be retransmitted. Soft combining is used, in which incorrect data is not discarded but stored in a buffer. When retransmitted data related to the same data is received, it is combined with the previous ones, increasing the chances of successful decoding. This mechanism is known as Hybrid ARQ with soft combining.

HARQ in 5G NR utilizes incremental redundancy (IR), where each retransmission carries a different subset of information and parity bits determined by the system's rate adaptation functionality. Each retransmission is called a redundancy version (RV), where the order of RV transmissions is 0, 2, 3 and 1. The first transmission, RV0, includes all systematic bits, while RV0 and RV3 are self-decodable [26, 27].

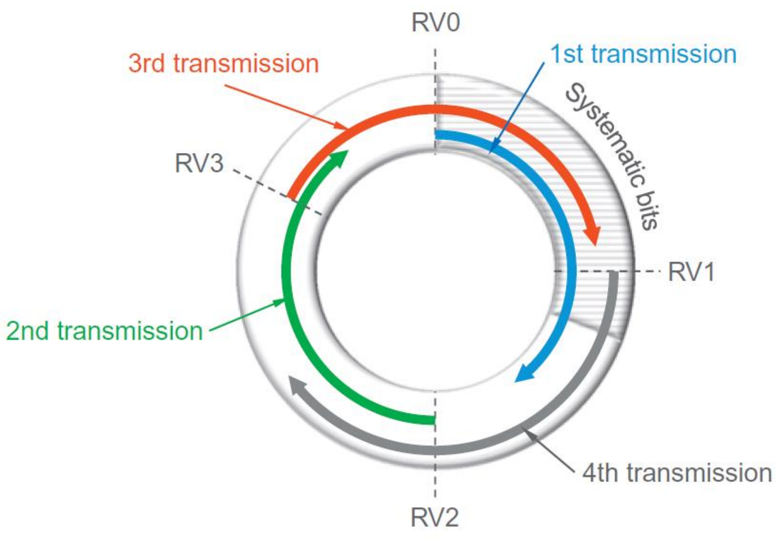

Figure 3.4: Example of 5G NR HARQ's Incremental Redundancy [26]

The error correction code used in 5G NR is Low-Density Parity Check (LDPC). The order of RV reception is important because not all RVs can be decoded independently. Multiple HARQ processes [40] run in parallel, allowing the system to handle multiple data transfers and retransmissions simultaneously, which is critical for maintaining high throughput and minimizing latency. This parallel approach ensures continuous data flow, even in the event of errors and retransmissions.

While HARQ improves transmission reliability, the challenge arises when high reliability is required under strict latency constraints. In such cases, the system may have limited capabilities for HARQ retransmissions. Therefore, each transmission must be designed to achieve high reliability from the start, reducing the need for multiple retransmissions within the latency budget.

Chapter 4

Joint Adaptive Transmission and Numerology Selection

The increasing complexity of 5G NR networks, combined with the dynamic nature of wireless environments, necessitates more intelligent and adaptive techniques for optimizing key parameters like MCS and numerology. Traditional rule-based methods often fall short of delivering optimal performance in such complex systems. Reinforcement learning (RL) presents a promising alternative, enabling systems to learn and adapt based on real-time feedback from the environment. In this chapter, we explore the application of Deep Q-Networks (DQN), a specific type of reinforcement learning algorithm, to optimize joint MCS selection and numerology configuration in 5G NR. By dynamically adjusting these parameters in response to varying channel conditions and system demands, the RL-based approach seeks to reduce latency, and ensure reliable communication for Ultra-Reliable Low Latency Communication (URLLC) use case. This chapter outlines the formulation of the RL problem, the design of the DQN agent, and the integration of this framework into our 5G NR simulation model.

4.1 Relative Work

In recent years, adaptive modulation and coding (AMC) and numerology selection in 5G NR have gained significant attention due to their critical role in optimizing wireless communication performance. Traditionally, rule-based techniques have been used to manage MCS and numerology configurations in response to different channel conditions [34–36]. However, such static approaches often have limited ability to handle the dynamic nature of modern 5G networks, especially in latency-sensitive and URLLC scenarios.

Several studies have explored the use of machine learning (ML) techniques, particularly reinforcement learning, to address these challenges. For example, [18] proposed that a deep learning-based link adaptation design is used for real-time MCS adaptation based on instantaneous channel conditions for eMBB/URLLC multiplexing in 5G NR. In another study in [28], a Deep Q-Network (DQN) was used to improve resource allocation in 5G NR systems with Automatic Repeat reQuest (ARQ) technique, dynamically adjusting parameters such as slot sizes, MCS and numerologies to minimize packet loss while maintaining latency constraints to comply.

Building on these approaches, [37] showed that RL techniques can outperform traditional link adaptation methods by making intelligent modulation and coding decisions based on historical data and real-time feedback. This work highlighted the benefits of using RL to optimize numerology selection alongside MCS, balancing throughput, latency and reliability requirements. Although these approaches are promising, many studies have primarily focused on modulation schemes and channel coding without fully integrating HARQ mechanisms, which are crucial for maintaining reliable communications in practical 5G NR deployments.

Furthermore, [35] introduced a flexible RL-based method framework that combines both explicit and implicit link adaptation strategies, demonstrating the potential of these systems to reduce transmission delays in URLLC applications. These studies laid the foundation for further exploration of RL methods to optimize resource allocation and MCS selection. However, a common limitation has been the lack of integration with real-world limitations such as HARQ mechanisms and the complexity presented by different numerologies in 5G NR systems.

In this project, we extend these advances by integrating practical channel estimation, HARQ mechanisms, MCS and numerology adjustment into a DQN-based solution to optimize MCS and numerology in 5G NR PDSCH transmissions. Our approach builds on MATLAB RL techniques and considers broader system dynamics, including HARQ, real-time SNR fluctuations, and strict latency requirements typical of URLLC applications. The aim of this work is to bridge the gap between theoretical research and practical implementation and provide a solution that can adapt to dynamic network conditions and meet the needs of modern 5G networks.

4.2 Reinforcement Learning Overview

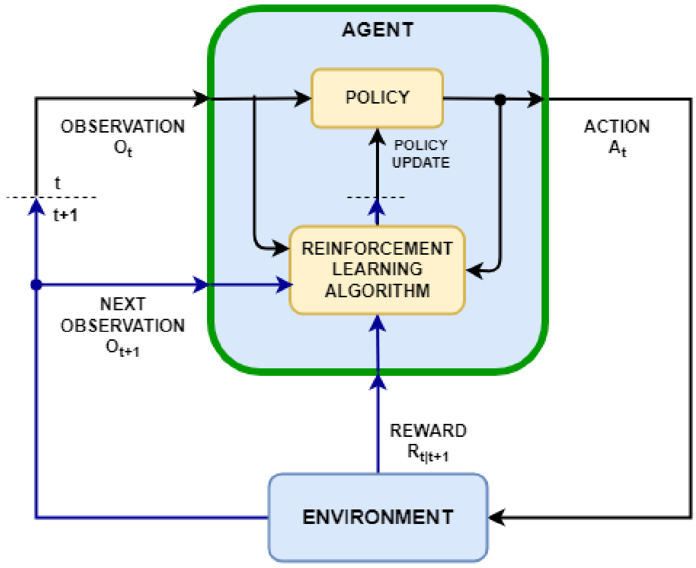

Figure 4.1: The Core Loop of Reinforcement learning [32]

Reinforcement learning is a type of machine learning technique in which an agent learns to perform a task through repeated trial-and-error interactions with a dynamic environment. This learning approach allows the agent to make a series of decisions that maximize a reward metric for the task, without the need for human intervention and without being explicitly programmed to complete the task [31, 32]. Unlike supervised learning, where the model is trained on labeled data, RL involves an agent that learns through trial and error. The agent takes actions in its environment, observes the results, and receives rewards based on the quality of its actions. Over time, the agent learns to maximize cumulative rewards by developing strategies that guide its actions in different states of the environment.

The core components of an RL system include the agent, environment, actions, states and rewards. The agent observes the current state of the environment, and takes an action based on a policy. The environment then transitions to a new state and provides feedback in the form of a reward. The agent updates its policies using a reinforcement learning algorithm that aims to improve the decisions made by maximizing the expected long-term reward.

The agent contains two components: a policy and a learning algorithm.

- The policy is a mapping from the current environment observation to a probability distribution of the actions to be taken. Within an agent, the policy is implemented by a function approximator with tunable parameters and a specific approximation model, such as a deep neural network.
- The learning algorithm continuously updates the policy learnable parameters based on the actions, observations, and rewards. The goal of the learning algorithm is to find an optimal policy that maximizes the expected discounted cumulative long-term reward received during the task.

In MATLAB's Reinforcement Learning Toolbox, this interaction loop is central to training RL agents. The toolbox provides preconfigured RL algorithms such as Deep Q-Network (DQN) and Policy Gradient methods that can be adapted to different environments, such as optimizing modulation and coding schemes and numerology in 5G NR systems [30]. Through trial and error, the agent continually improves its policy and adjusts its actions based on feedback received from the environment.

Fig. 4.1 shows the core loop of reinforcement learning, where the agent's policy is refined by the rewards received from the environment and ultimately optimizes the decision-making process for a given task.

4.3 System Model

This project focuses on point-to-point data packet transmission in a 5G NR PDSCH system. The system supports Adaptive Modulation and Coding (AMC), flexible numerology and HARQ mechanisms [14], ensuring reliable and efficient communication under strict latency constraints and varying channel conditions.

4.3.1 Channel Model and System Bandwidth

The system operates over a 25 MHz frequency bandwidth, dedicated to URLLC traffic. To simulate a realistic wireless environment, we use the Tapped Delay Line-C (TDL-C) channel model specified as Rayleigh distribution channel. The TDL-C model introduces frequency-selective fading and temporal variations in the channel gain, making the channel behavior more complex compared to basic slow frequency-flat fading models [38, 39]. This presents a more realistic challenge for maintaining reliable communication under tight URLLC latency constraints.

4.3.2 Adaptive Modulation and Coding (AMC) and Numerology

The system supports a range of modulation and coding schemes (MCS) in [9] and index them by I_{MCS} which is used to indicate the selected modulation order and coding rate that best matches the channel quality indicator (CQI) based on the signal-to-noise ratio (SNR) of the channel. In addition, the system supports multiple numerologies μ, that enable variable SCS. Numerology affects the Transmission Time Interval (TTI), with higher numerologies resulting in shorter transmission durations. The number of OFDM symbols per slot N_{sym} is set to 14 symbols in accordance with 5G NR standards.

4.3.3 Data Transmission and Subframe Calculation

Each data packet consists of H bits, and the system must ensure that the total transmission time T_{total} remains within a strict latency constraint threshold T_{th}. The transmission time depends on the modulation order, coding rate, numerology and the number of physical resource blocks (PRB) allocated for transmission. The number of subframes N_{sf} required to transmit a packet is calculated in [14] as follows:

$$N_{sf} = \left\lceil \frac{H}{Q_m \times N_{RE}^{PRB} \times R \times PRB} \right\rceil \tag{4.1}$$

where Q_m represents the modulation order (bits per symbol), N_{RE}^{PRB} denotes the number of resource elements per PRB, R is the code rate, and PRB is the number of allocated PRBs.

The duration of each transmission subframe T_{sf} (calculated at 2.1) is determined by the numerology μ, which defines the subcarrier spacing. The Transmission Time Interval (TTI) for a particular MCS and numerology is then calculated as follows:

$$t_{TTI} = N_{sf} \times T_{sf} \tag{4.2}$$

4.3.4 HARQ Mechanism and Retransmissions

To ensure reliable packet delivery, the system implements HARQ mechanism [40]. When the receiver detects errors in transport block, it immediately sends a retransmission request. Unlike initial transmissions, HARQ retransmissions do not require retransmission of the entire packet. Instead, only portions of the data, such as parity bits or selective retransmission blocks, are sent by reducing the overall retransmission time.

The total transmission time T_{total} for a packet includes both the initial transmission time t_{TTI} and the cumulative time for all necessary HARQ retransmissions. The retransmissions time t_{HARQ} is based on the number of retransmission attempts and the amount of data that needs to be retransmitted. In the case of HARQ, the total transmission time T_{total} is calculated as follows:

$$T_{total} = t_{TTI} + t_{HARQ} \tag{4.3}$$

The retransmission time t_{HARQ} is typically calculated by considering the number of retransmission attempts $N_{retries}$ and the amount of data retransmitted in each attempt. Since HARQ retransmissions only involve sending additional parity bits or selective parts of the data, the transmission time for each retransmission is based on the number of retransmitted bits rather than the full packet size. The cumulative retransmission time can be expressed as:

$$t_{HARQ} = \sum_{i=1}^{N_{retries}} (r\_N_{sf}^{(i)} \times T_{sf}) \tag{4.4}$$

where, $r\_N_{sf}^{(i)}$ is the number of subframe used for the i^{th} retransmission. The packet is considered successfully received if T_{total} remains within the latency constraint T_{th}. Exceeding this latency will result in the packet being classified as lost.

This system model provides a comprehensive framework for evaluating the performance of 5G NR PDSCH transmission under the demanding conditions of URLLC, while taking into account the complexities introduced by the TDL-C channel environment.

4.4 MDP Formulation

This project focuses on optimizing the selection of modulation and coding scheme (MCS) and numerology for 5G NR PDSCH transmissions. The goal is to achieve reliable packet delivery while meeting the strict latency requirements of URLLC applications. The problem is modeled as a finite-horizon Markov decision process (MDP) [28] and solved using a deep reinforcement learning (DRL) approach [41, 42], specifically the Deep Q-Network (DQN) algorithm using MATLAB reinforcement learning toolbox [30, 31].

Observation space: In the MDP formulation, the observation space captures the key parameters that influence the transmission process. These include the packet size (H), the channel quality (γ) represented by the Signal-to-Noise Ratio (SNR), the remaining latency budget (τ), and the number of Hybrid Automatic Repeat reQuest (HARQ) retransmissions ($q_i = 0,1,2,3,4$). At each decision point, the system's observation state is defined by the vector $S = (H, \gamma, \tau, q)$, representing the current transmission context. Initially, the remaining latency budget is set to the maximum latency threshold T_{th}, while the HARQ retransmission count starts from zero. Two terminal states are defined as “Success” if the packet was successfully delivered within the latency requirements and as “Fail” if the remaining latency budget is no longer sufficient to support further transmissions.

Action space: The action space of the MDP includes the selection of the MCS index (I_{MCS}) and numerology (μ), which determines the subcarrier spacing (SCS). These parameters are dynamically selected based on the current state to optimize the transmission rate and time interval. The action is represented by $A = (I_{MCS}, \mu)$, where the agent's task is to intelligently choose an optimal modulation and coding scheme, and numerology based on the observed channel conditions and latency requirements.

Transition dynamics: The system transitions from one state to another after each transmission attempt. Let us first consider transition between non-terminating states, i.e., from (H, γ, τ, q) to (H, γ', τ', q). Note H remains constant for the same packet transmission and q will reset to zero after each transmission attempt. After taking $A = (I_{MCS}, \mu)$, the new latency budget is updated by subtracting the time needed for the current transmission and any HARQ retransmissions as $\tau' = \tau - (t_{TTI}(I_{MCS}, \mu) + t_{HARQ})$. The new SNR γ' is determined by varying channel condition. If the remaining latency budget after a transmission attempt, including HARQ retransmission time, falls below the minimum required time for the next transmission, the system transitions to a "Fail" state. If the packet is successfully received within the latency budget, the process transitions to the terminating "Success" state.

Reward function: To guide the agent in learning effective strategies, a reward function is defined to provide feedback for each action. A large positive reward is given when a packet is successfully transmitted within the latency requirements. Conversely, a large negative reward is applied if the transmission fails, either due to packet was not transmitted successfully or does not meet the latency constraint. Additional penalties are given for incomplete transmissions at non-terminal state. These reward signals help the agent learns to prioritize actions that result in successful transmissions with minimal delay and efficient use of resources. The reward function can be expressed as:

$$R(s') = \begin{cases} C, & s' = Success, \\ -C, & s' = Fail, \\ -C/10, & s' = Non\,Terminal\,State \end{cases} \tag{4.5}$$

where C is a positive constant while different value of C may lead to different rate of convergence.

4.5 DQN-based solution

This project applies the Deep Q-Network (DQN) algorithm [30] with a focus on double DQN for optimal MCS and numerology selection in a 5G NR PDSCH transmission system. The agent is designed to maximize reliable data transmission under stringent latency constraints by learning a Q-value function $Q(S, A; \phi)$. This Q-value function estimates the expected long-term reward for taking action A (the selection of MCS and numerology) in a given observation S (channel condition, packet size, latency budget, and number of HARQ retransmissions). The learnable parameters of the Q-value function are represented by ϕ.

Our implementation of the DQN agent in MATLAB's Reinforcement Learning Toolbox [31] uses the *rlDQNAgent* object, where the Q-value function is approximated through a critic network. The network is trained to minimize the difference between predicted and actual rewards, with the agent learning optimal transmission strategies over time. In our specific case, the action space is discrete, representing combinations of MCS index and numerology values, and the observation space is continuous, consisting of various transmission metrics.

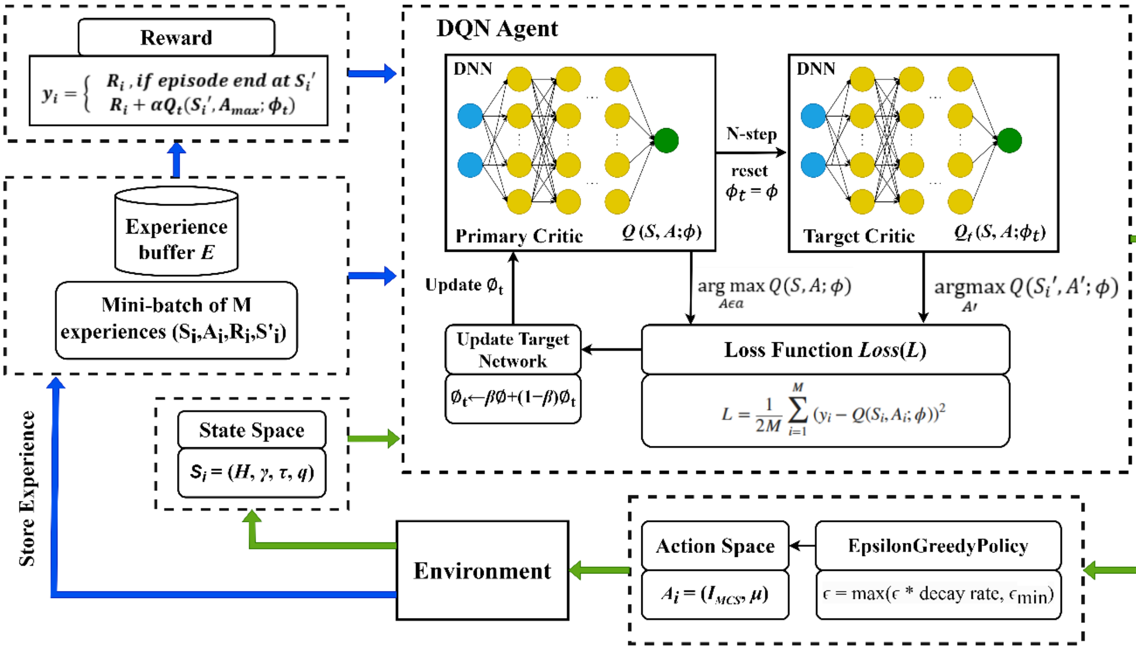

Figure 4.2: DQN Training Structure

4.5.1 Double DQN and Critic Function

To improve stability and mitigate overestimation bias, we use the Double DQN [30] approach, which involves two parametrized action-value functions (Q-value function) approximators: the Primary Critic $Q(S, A; \phi)$; given observation S and action A, this critic stores the corresponding estimate of the expected discounted cumulative long-term reward when following the optimal policy, and the Target Critic $Q_t(S, A; \phi_t)$; to improve the stability of the optimization, the agent periodically updates the target critic learnable parameters ϕ_t using the latest critic parameter values. The target critic is used to calculate the target value for the Q-function update, and it is updated periodically to prevent oscillations in learning. The key difference in double DQN is that the double DQN algorithm selects the action A that maximizes the Q-value is selected by the primary network (Primary Critic), while the normal DQN selects the value of this action is evaluated using the target network.

The critic network is responsible for selecting the best action A_{max} which can express as:

$$A_{max} = \underset{A'}{\operatorname{argmax}}\, Q(S_i', A'; \phi) \tag{4.6}$$

and observe the reward R, and the critic network provides the Q-value of that action for the next observation S'. For all experiences store in the mini-batch, if S_i' is a terminal state, set the target value y_i to R_i. Otherwise, the target value y_i for a given mini-batch experience is computed as $y_i = R_i + \alpha Q_t(S_i', A_{max}; \phi_t)$.

$$y_i = \begin{cases} R_i, if\ the\ episode\ terminates\ at\ S_i' \\ R_i + \alpha Q_t(S_i', A_{max}; \phi_t), otherwise \end{cases} \tag{4.7}$$

where α is the discount factor that balances immediate rewards with future rewards, R_i is the reward observed after taking action A', A_{max} is the action that maximizes the Q-value function for the next observation S_i', and ϕ_t represents the weights of the target critic network.

4.5.2 ϵ-Greedy Exploration

The agent explores the action space using an ϵ-greedy policy. Initially, the agent selects random actions with a high probability ϵ, which decreases over time to favor exploitation of the learned Q-values. This balance between exploration and exploitation allows the agent to efficiently explore suboptimal actions during the early stages of training while converging to an optimal policy over time. The decay rate of ϵ is controlled by the *EpsilonGreedyExploration* option in MATLAB [30].

To balance exploration (trying new actions) and exploitation (leveraging learned actions), we implement the epsilon-greedy exploration strategy. The agent begins by exploring the environment extensively, gradually shifting towards exploiting the learned policy as training progresses.

> Exploration: At the beginning of training, the agent selects actions randomly with a high probability ϵ, which encourages exploring various combinations of MCS and numerology. Exploration is essential to help the agent discover the optimal action in various states.

Exploitation: As the agent learns more about the environment, it shifts towards exploiting the learned Q-values, where the action A is chosen to maximize the Q-value:

$$A = \arg\max_{A \epsilon a} Q(S, A; \phi) \tag{4.8}$$

The probability of exploiting the best action increases over time as ϵ decays. The epsilon decay strategy is defined as:

$$\epsilon = max(\epsilon \times \text{decay rate}, \epsilon_{min}) \tag{4.9}$$

where initial ϵ is the exploration starts with a high probability (e.g., $\epsilon = 1$), decay rate is the factor gradually reduces the exploration probability as training progresses and ϵ_{min} is a minimum value for ϵ is enforced to ensure that there is always some chance of exploring new actions.

4.5.3 Experience Replay and Mini-Batch Updates

The agent stores experiences $(S, A, R, S_i{}')$ in an experience buffer (E), which is used to sample a random mini-batch of M experiences $(S_i, A_i, R_i, S_i{}')$ from the experience buffer. This technique allows the agent to break correlations between consecutive experiences and improve learning efficiency. Each mini-batch is used to update the Q-value function by minimizing the following loss L function:

$$Loss(L) = \frac{1}{2M} \sum_{i=1}^{M} (y_i - Q(S_i, A_i, \phi))^2 \tag{4.10}$$

where M is the mini-batch size and y_i is the target value calculated using the double DQN equation.

4.5.4 Target Network Updates

We employ a smoothing update method for the target critic network $Q_t(S, A; \phi_t)$, the agent updates the target parameters at every time step using smoothing factor β and the target weights ϕ_t are updated as:

$$\phi_t = \beta\phi + (1 - \beta)\phi_t \tag{4.11}$$

This method ensures gradual updates to the target network, preventing sudden shifts in policy that could destabilize the agent's learning process. The update frequency is controlled by the *TargetUpdateFrequency* parameter options in MATLAB [30].

4.5.5 DQN Training Algorithm

Algorithm 1 DQN Agent Training Algorithm

1: Input: Experience buffer *E*, reward discount factor α, mini-batch size *M*, exploration factor ϵ and ϵ decay rate

2: Initialize primary critic with random parameter value ϕ, and the target critic parameters with the same value $\phi_t = \phi$.

3: for Episode k = 1, 2, ... do:

4: Initialize packet size H, latency requirement T_{th} and HARQ retransmission $q = 0$.

5: Initialize exploration factor $\epsilon = 1$.

6: for each transmission attempt for episode k do:

7: For the current state S_i, select action A_i:

$$A_i = \begin{cases} random\ action, & with\ probability\ \epsilon \\ arg\ \max_{A \epsilon a} Q(S_i, A, \phi), & with\ probability\ 1 - \epsilon \end{cases}$$

8: Execute action A_i, observe reward R_i, and the experience buffer *E*.

9: Store experience (S_i, A_i, R_i, S_i') into the experience buffer *E*.

10: If the experience buffer *E* is full, randomly sample a mini-batch of *M* experiences (S_i, A_i, R_i, S_i') from *E*.

11: For each sampled experience, compute the target value y_i:

$$A_{max} = \underset{A'}{\operatorname{argmax}}\, Q(S_i', A'; \phi)$$

$$y_i = \begin{cases} R_i, if\ the\ episode\ terminates\ at\ S_i' \\ R_i + \alpha Q_t(S_i', A_{max}; \phi_t), otherwise \end{cases}$$

12: Update primary critic parameter by one-step minimization of the loss *L* across all sampled experiences.

$$Loss(L) = \frac{1}{2M}\sum_{i=1}^{M}(y_i - Q(S_i, A_i, \phi))^2$$

13: Update target critic parameter using Smoothing target update method:

$$\phi_t = \beta\phi + (1 - \beta)\phi_t$$

14: Update the probability threshold ϵ for selecting a random action based on the decay rate:

$$\epsilon = max(\epsilon \times decay\ rate, \epsilon_{min}).$$

15: Return *N*-step reset $\phi_t = \phi$

16: End episode if terminal state reached.

17: Return the trained primary critic parameters ϕ.

4.6 Environmental Setup

In this section, we describe the environment setup used to train, test, and deploy a Deep Q-Network (DQN) agent to optimize MCS and numerology selection in a 5G NR PDSCH transmission system. The environment is designed to simulate data packet transmission over a wireless channel, incorporating important parameters such as data packet size, SNR, latency budget, and the number of HARQ retransmissions.

The observation space for the DQN agent consists of four key variables: the data packet size, channel condition, latency budget, and the number of HARQ retransmissions. The action space is discrete and represents various combinations of modulation and coding schemes (MCS) and numerology. This setup allows the agent to select the best MCS and numerology based on the current channel conditions and latency constrain.

A custom environment is created by defining step and reset functions [42] that simulate the 5G NR transmission process. The step function evaluates the actions taken by the agent, calculates the number of bits transmitted, updates the packet size, and calculates the reward based on latency and successful transmission. The agent's reward is tuned to favor the selection of optimal MCS and numerology settings that meet latency and throughput requirements while minimizing retransmissions.

The DQN agent uses a neural network-based critic to estimate the Q-values of state-action pairs. The network consists of state space and action space, which are merged to produce a Q-value that estimates the expected reward for each action. The training process involves adjusting network parameters using stochastic gradient descent with an ϵ-greedy exploration strategy to balance exploration and exploitation.

After training, the agent can be evaluated by testing in the same environment to verify its ability to optimize 5G NR transmission. After testing, the trained agent is saved and can be deployed in real 5G systems or used in further simulations to optimize performance under different conditions.

Table 4.1: Environment Parameters

Parameter	Values
Observation Space	Packet Size, SNR, Latency, No. of HARQ Retries
Action Space	MCS and Numerology
Modulation Schemes (MCS)	QPSK, 16QAM, 64QAM, 256QAM
Subcarrier Spacings (Numerology)	15, 30, 60, 120 kHz
Max Latency	1 ms
SNR Range	[-5, 20] dB

Table 4.2: DQN Hyperparameters

Parameter	Values
Max Episodes	3000
Max Steps per Episode	60
Learning Rate (Critic)	$1x10^{-4}$
Mini-Batch Size	64
Exploration Strategy	ϵ - Greedy (ϵ = 1, Decay = 0.99999, Min = 0.1)
Target Network Update Frequency	2
Target Smoothing Factor	$1x10^{-2}$
Discount Factor (α)	0.99
Experience Buffer Length	$2x10^{6}$

This setup and parameter configuration enable the DQN agent to optimize transmission efficiency in a 5G NR system by dynamically adjusting the MCS and numerology based on real-time channel conditions and system requirements.

Chapter 5

Results and Analysis

In this chapter, we present the results of our DQN-based solution for adaptive modulation and numerology selection in 5G NR PDSCH transmission. The main goal was to train an agent capable of dynamically adjusting transmission parameters such as MCS and numerology based on channel conditions and latency requirements, ensuring efficient and reliable communication. We implemented the system using MATLAB's Reinforcement Learning Toolbox [31], with the environment modeling realistic wireless channel behaviors, including fading, noise, and latency constraints. The environment parameters and DQN hyperparameters for training are summarized in Table 4.1 and Table 4.2.

We evaluated performance based on reward history during training, selected MCS and numerology for various SNR and latency conditions, as well as key performance metrics such as transmission time and packet loss rate.

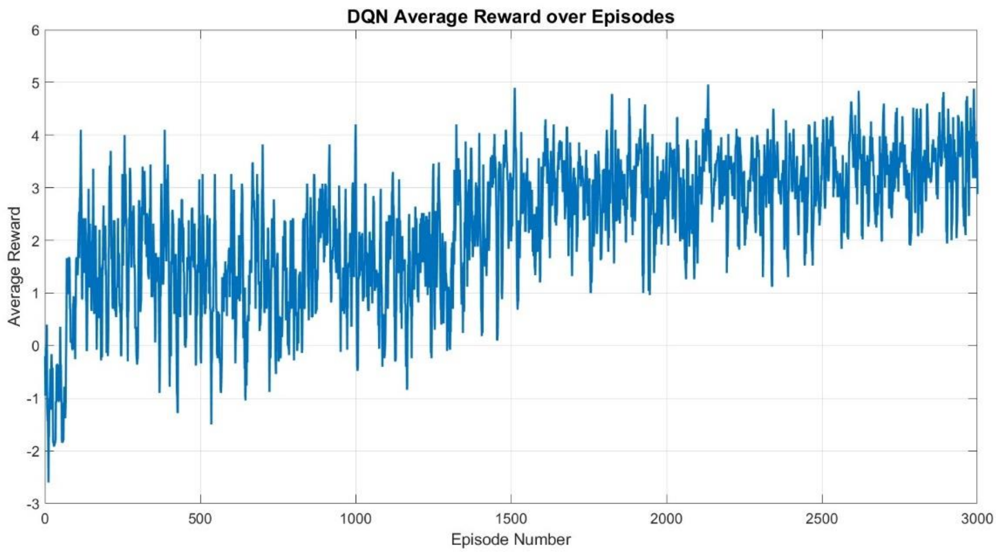

Figure 5.1: Average Reward over Episodes for DQN Agent Training

Fig. 5.1 shows the average reward history over 3000 episodes for the trained DQN agent. The average reward is computed by averaging the rewards over the most recent 5 episodes to smooth short-term variations. The episode reward provides immediate feedback on how well the agent performed during each individual episode, while the average reward provides a smoothed measure of the agent’s performance over multiple episodes, offering a clearer view of the agent’s overall learning progress. Initially, the agent explores various actions, resulting in lower rewards due to suboptimal decisions. However, as training progresses, the agent refines its decision-making process, gradually increasing the average reward. Around episode 300, the agent stabilizes, showing both an increase in rewards and less fluctuation, indicating that it has learned a more consistent and effective policy.

In this training, the reward function used is defined in equation (4.5), with a base reward constant C set to 1. This base reward forms the foundation of the agent's training, providing positive reinforcement for successfully completed transmissions within the latency budget and penalizing failures. To further guide the agent’s decisions, additional rewards were integrated to prioritize specific aspects of transmission performance. A positive reward of +3 is provided when the transmission is completed without any HARQ retransmission attempts, with each retransmission incurring a penalty of -1. This encourages the agent to minimize HARQ retransmissions and reducing the risk of block errors by selecting lower MCS and numerology index when the SNR is low.

Another additional reward is added based on the remaining latency budget, defined as $Base\ Reward + \frac{\tau}{T_{th}}$, where τ represents the remaining latency budget and T_{th} is the latency threshold, incentivizes the agent to use higher MCS and numerology index when the latency budget is tight. This enables the agent to adapt its transmission strategy, favoring faster transmission under stringent latency constraints while maintaining reliability under challenging channel conditions. Without these additional rewards, the agent may not sufficiently prioritize reducing the risk of block errors that lead to HARQ retransmissions or optimizing transmission delay under tight latency conditions, potentially resulting in suboptimal behavior. The combined reward structure thus effectively balances reliability and transmission efficiency, allowing the agent to adapt its actions to meet the low latency and high reliability based on dynamic channel conditions.

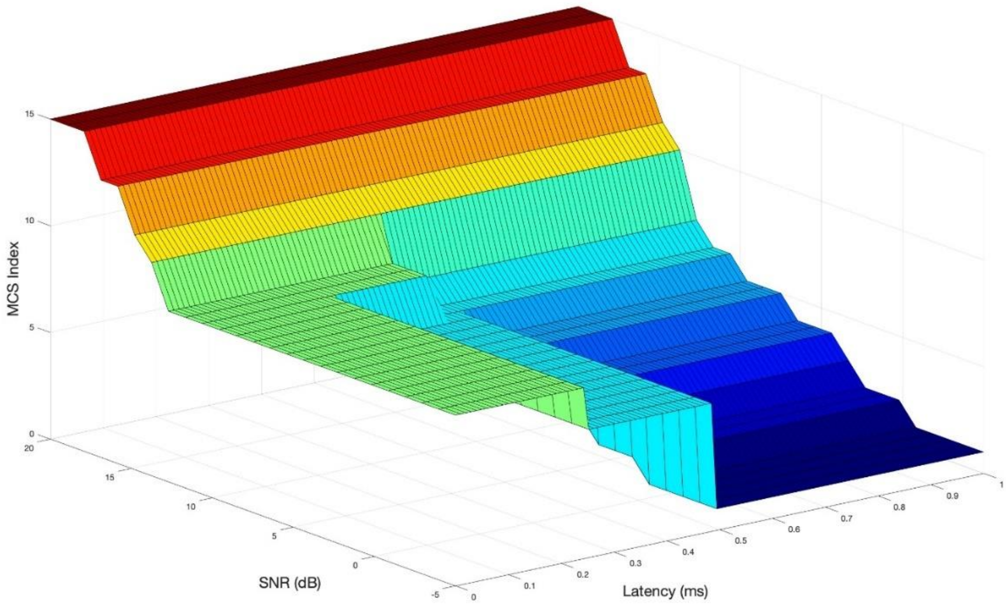

Figure 5.2: DQN agent behaviour for selecting MCS under optimal policy for different SNRs and Latency

Fig. 5.2 and Fig. 5.3 illustrate the behavior of the optimal policy obtained using the DQN algorithm for transmitting a packet with H = 1000 bits. Fig. 5.2 shows the selected modulation and coding scheme for different SNRs and latency budgets under the optimal policy. As we can see, for high SNR value, the agent selects higher MCS index, since the channel conditions are favorable. This allows the system to use higher-order modulation schemes and increase throughput. When SNR decreases, meaning poor channel condition, the agent selects a lower MCS index to improve reliability and provide better robustness against noise. However, if there's a latency constraint, the agent selects a higher MCS index even in low SNR conditions. This helps the system transmit more data to meet the latency requirements within the given time limit. So, the agent smartly balances between ensuring reliable transmission in poor signal conditions and meeting latency demands when necessary. On the other hand, when the latency budget is larger, the policy prefers a lower MCS to improve transmission reliability and ensure successful packet delivery under poor channel conditions.

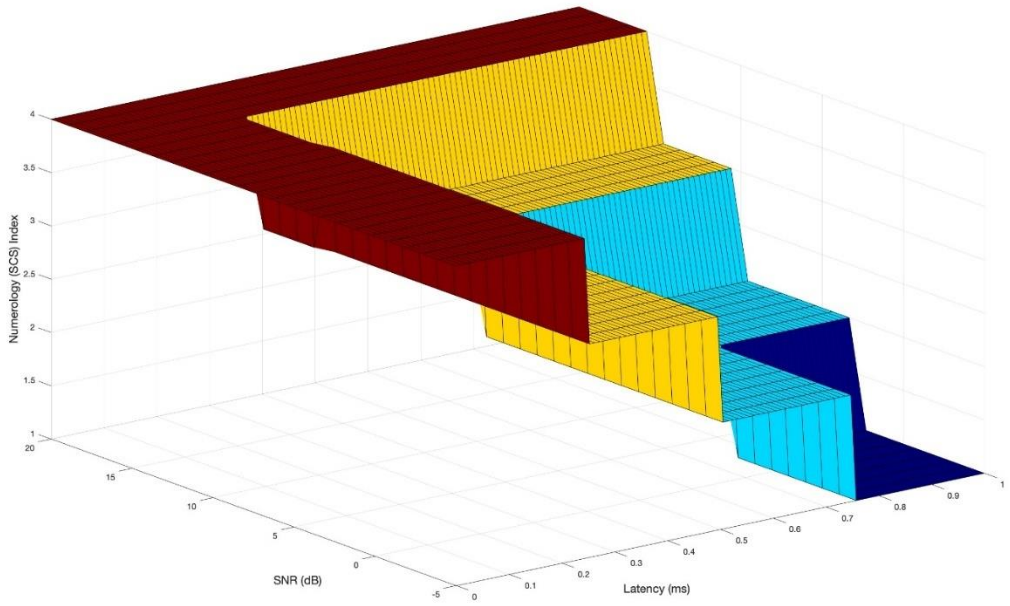

Figure 5.3: DQN agent behaviour for selecting Numerology (SCS) under optimal policy for different SNRs and Latency

Fig. 5.3 shows the DQN agent behaviour for selecting Numerology Index (SCS) for different SNRs and latency budgets under the optimal policy. When SNR values are high, the agent chooses a higher numerology index corresponding to a larger subcarrier spacing which enabling shorter transmission times. This ensures that data can be transferred quickly when channel conditions are good. However, if the SNR decreases, the agent chooses a lower numerology index with smaller subcarrier spacing to increase robustness to noise and interference. A smaller intercarrier spacing helps maintain signal reliability in these poor channel conditions. However, when there is a tight latency constraint even under low SNR conditions, the agent chooses a higher numerology index to meet the strict latency requirements by transmitting the data faster within the available time. Thus, the agent dynamically balances between maintaining reliability in poor channel conditions and meeting latency requirements by adjusting both the MCS and numerology indices based on the SNR and latency budget.

Table 5.1 System parameters for PDSCH transmission simulation

Parameters	Values
Subcarrier spacing	15, 30, 60, 120 kHz
System bandwidth	25 MHz
FFT/IFFT length	1024
Modulation schemes	QPSK, 16-QAM, 64-QAM, 256-QAM
Coding/Decoding scheme	LDPC / Min-Sum decoding
Channel model	TDL-C (Tapped Delay Line - C) with AWGN
SNR range	-4:2:16 dB
Number of packets	100000
Data packet size, *H*	1000 bits
Max latency threshold, *Tth*	1 ms

To further evaluate the performance of the trained DQN agent, we analyzed two key metrics: packet loss rate and average transmission time in the 5G NR PDSCH transmission simulation environment using the system parameters provided in Table 5.1. To enable a comparative analysis with our DQN-based adaptive transmission design, we use a traditional adaptive MCS transmission system configured with a fixed SCS. In traditional adaptive modulation and coding (AMC), the system dynamically selects the MCS based on predefined SNR-BLER curves for each CQI level (1-15), targeting a BLER value of 10^{-5}. This approach adjusts the MCS according to current SNR, aiming to balance data throughput and reliability using a fixed SCS.

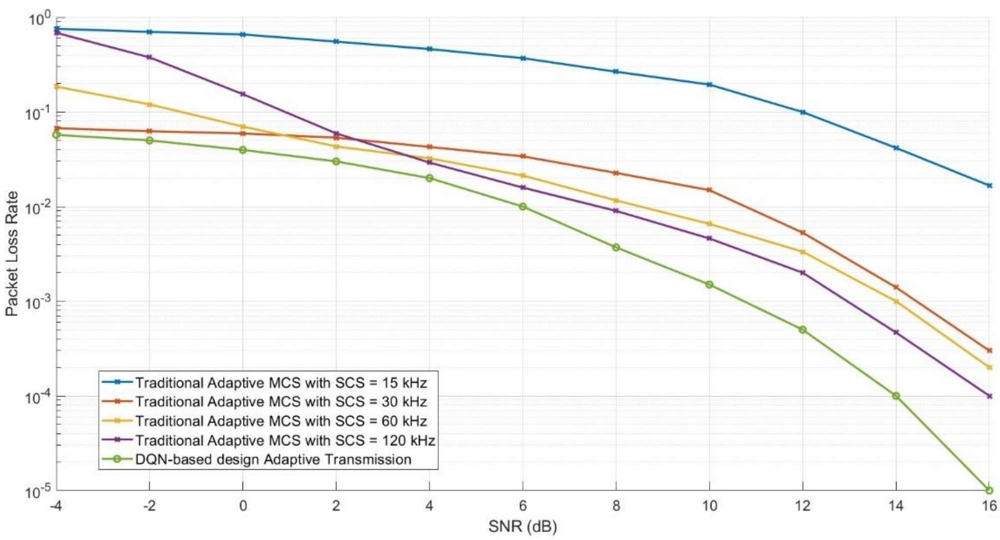

Figure 5.4: Packet Loss Rate across different SNRs in DQN-optimized 5G NR PDSCH transmission

In terms of performance comparison, Fig. 5.4 shows the packet loss rate over different channel conditions for different transmission approaches, including traditional adaptive MCS configurations with different SCS values and our DQN-based adaptive transmission. A packet is considered lost or error if it is not delivered successfully within the latency budget. It can be observed that the DQN-based adaptive transmission achieves a noticeable improvement over the traditional approaches, demonstrating lower packet loss rate over different channel conditions and significantly better performance in maintaining packet reliability.

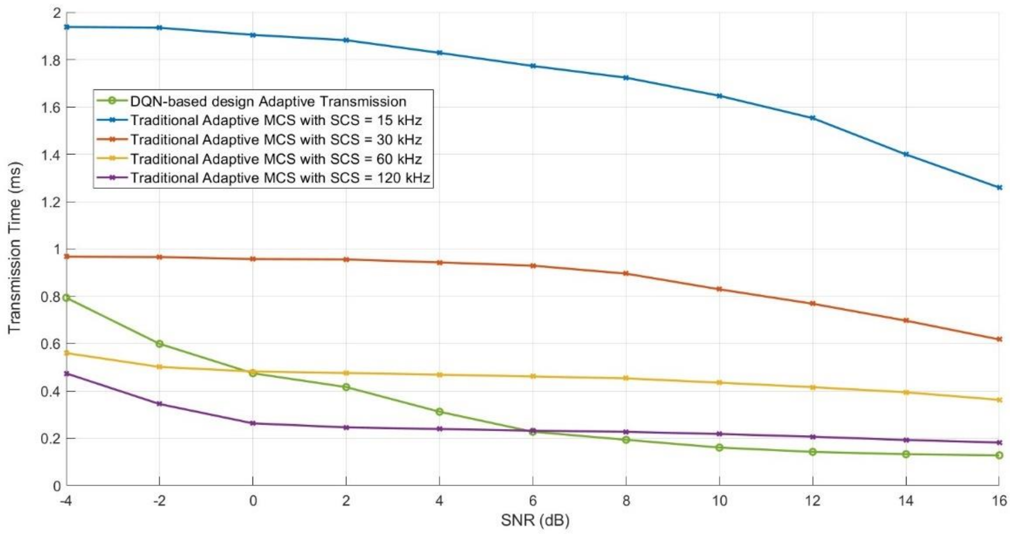

Figure 5.5: Average Transmission Time across different SNRs in DQN-optimized 5G NR PDSCH transmission

Fig. 5.5 shows the average transmission time across different SNR values, comparing DQN-based adaptive transmission with traditional MCS transmissions using fixed SCS values. Average transmission time is calculated by measuring the time required for the successful transmission of a data packet including any retransmissions and averaging this over multiple packets under the given channel conditions. For lower SNR values, the DQN agent selects optimal MCS and numerology configurations to maintain reliability and meet the latency budget. As SNR increases, the DQN agent adapts by choosing higher MCS and numerology indices, optimizing for faster transmission. In comparison, traditional methods with fixed SCS result in longer transmission times under certain conditions. But the DQN-based approach adapts more effectively, reducing transmission time.

The results demonstrate the ability of the DQN agent to effectively optimize 5G NR PDSCH transmission parameters. By dynamically adjusting the MCS and selecting numerology based on the real-time channel conditions and latency requirements, the agent significantly reduces transmission time while maintaining a near-zero packet loss rate. This adaptability is critical for URLLC, where both reliability and minimal latency are important.

Chapter 6

Conclusions and Future work

6.1 Conclusions

In this project, we developed and implemented a reinforcement learning-based solution using a Deep Q-Network (DQN) agent to optimize 5G NR PDSCH transmission parameters, with emphasis on Adaptive Modulation and Coding Scheme (MCS) and numerology selection. The DQN agent has been trained to dynamically adapt to different signal-to-noise ratios (SNR) and latency constraints, resulting in efficient performance and reliable transmissions. Simulation results show that the agent significantly reduces transmission time while maintaining low packet loss rates, which is essential for meeting the stringent requirements of URLLC applications.

The adaptability of the DQN agent ensures optimal performance under real-time channel conditions and provides a robust solution for improving transmission efficiency in 5G systems. By fine-tuning parameters such as MCS and numerology based on real-time feedback from the channel, the DQN approach provides a significant improvement over traditional static, rule-based methods, resulting in lower transmission delay, lower packet loss rates and higher reliability.

6.2 Future work

While the current implementation shows significant improvements, there are several areas for future investigation. First, incorporating more advanced channel models, such as those that take Doppler shifts and mobility into account, would enable a more realistic assessment of the DQN agent's performance in dynamic environments. Furthermore, expanding the scope of this work to consider MIMO multi-antenna configurations and beamforming techniques could further improve the efficiency and reliability of the system.

Another avenue for future work could be to explore alternative reinforcement learning algorithms such as Actor-Critic methods or Proximal Policy Optimization (PPO) to compare performance and convergence rates. Furthermore, integrating the DQN agent into a real-time 5G network testbed would enable practical evaluation and further fine-tuning under real-world conditions.

Ultimately, this work lays a solid foundation for the application of reinforcement learning in 5G NR systems. Future research can build on these findings to address more complex scenarios and drive continuous improvements in wireless communication performance.

Bibliography

[1] B. Bertenyi, S. Nagata, H. Kooropaty, X. Zhou, W. Chen, Y. Kim, X. Dai, and X. Xu, *“5g nr radio interface," Journal of ICT Standardization,* vol. 6, no. 1, pp. 31-58, 2018.

[2] B. P. Sahoo, C.-C. Chou, C.-W. Weng, and H.-Y. Wei, *“Enabling millimeter-wave 5g networks for massive iot applications: A closer look at the issues impacting millimeter-waves in consumer devices under the 5g framework," IEEE Consumer Electronics Magazine,* vol. 8, no. 1, pp. 49-54, 2018.

[3] G. J. Sutton, J. Zeng, R. P. Liu, W. Ni, D. N. Nguyen, B. A. Jayawickrama, X. Huang, M. Abolhasan, Z. Zhang, E. Dutkiewicz *et al.*, “Enabling technologies for ultra-reliable and low latency communications: From phy and mac layer perspectives,” *IEEE Communications Surveys & Tutorials*, vol. 21, no. 3, pp. 2488–2524, 2019.

[4] S.-Y. Lien, S.-L. Shieh, Y. Huang, B. Su, Y.-L. Hsu, and H.-Y. Wei, “5g new radio: Waveform, frame structure, multiple access, and initial access,” *IEEE communications magazine*, vol. 55, no. 6, pp. 64–71, 2017.

[5] E. Memisoglu, A. B. Kihero, E. Basar, and H. Arslan, “Guard band reduction for 5g and beyond multiple numerologies,” *IEEE Communications Letters*, vol. 24, no. 3, pp. 644–647, 2019.

[6] L. You, Q. Liao, N. Pappas, and D. Yuan, “Resource optimization with flexible numerology and frame structure for heterogeneous services,” *IEEE Communications Letters*, vol. 22, no. 12, pp. 2579–2582, 2018.

[7] A. Akhtar and H. Arslan, “Downlink resource allocation and packet scheduling in multi-numerology wireless systems,” in *2018 IEEE wireless communications and networking conference workshops (WCNCW).* IEEE, 2018, pp. 362–367.

[8] Liu, Baojin & Lyu, Xiaoyong & Fan, Wenbing. (2022). *Analysis of 5G Signal for Radar Application. Journal of Physics:* Conference Series. 2356. 012027. 10.1088/1742-6596/2356/1/012027.

[9] *3GPP TS 38.300 V15.7.0, NR;NR and NG-RAN Overall Description;Stage 2,* Sep 2019.

[10] *3GPP TS 38.331 V15.7.0, NR;Radio Resource Control (RRC) protocol specification,* Sep 2019.

[11] *3GPP TS 38.104 V15.7.0, NR;Base Station (BS) radio transmission and reception,* Sep 2019.

[12] *3GPP TS 38.211 V15.7.0, NR;Physical channels and modulation,* Sep 2019.

[13] *3GPP TS 38.212 V15.7.0, NR;Multiplexing and channel coding,* Sep 2019.

[14] 3GPP, "NR; Physical Layer Procedures for Data (release15)," *document 3GPPTS 38.214,* vol. v15.9.0, Apr. 2020.

[15] Yang, H.-C. (2017). *Introduction to Digital Wireless Communications*. Institution of Engineering and Technology.

[16] *3GPP, "5G; Study on New Radio (NR) access technology," document 3GPP TR 38.912 version 14.0.0 Release 14.*

[17] J. Khan and L. Jacob, "Link adaptation for multi-connectivity enabled 5g urllc: Challenges and solutions," *in 2021 International Conference on Communication Systems & NETworkS (COMSNETS).* IEEE, 2021, pp. 148–152.

[18] Y. Huang, Y. T. Hou, and W. Lou, “Deluxe: A dl-based link adaptation for urllc/embb multiplexing in 5g nr,” *IEEE Journal on Selected Areas in Communications*, vol. 40, no. 1, pp. 143–162, 2021.

[19] *5G NR Physical Downlink Shared Channel (PDSCH).* MATLAB & Simulink. (n.d.). https://www.mathworks.com/support/search.html/videos/5g-explained-downlink-data-in-5g-nr-1558600809645.html?fq%5B%5D=category%3A5g%2Findex&page=1

[20] *DL-SCH and PDSCH Transmit and Receive Processing Chain* - MATLAB & Simulink. (n.d.). https://www.mathworks.com/help/5g/gs/dl-sch-and-pdsch-transmit-and-receive-processing-chain.html

[21] *NRLDPCENCODE*. MATLAB & Simulink. (n.d.-b). https://www.mathworks.com/help/5g/gs/ldpc-processing-chain-for-dl-sch.html

[22] *Model 5G NR transport channels with HARQ*. *Model 5G NR Transport Channels with HARQ* - MATLAB & Simulink. (n.d.). https://www.mathworks.com/help/5g/gs/model-5g-nr-transport-channels-with-harq.html

[23] *AWGN*. MATLAB & Simulink. (n.d.-a). https://www.mathworks.com/help/wlan/ug/snr-definition-used-in-end-to-end-simulations.html?searchHighlight=SNR&s_tid=srchtitle_support_results_5_SNR

[24] IEEE Std 802.11 - 2020. *Telecommunications and Information Exchange between Systems Local and Metropolitan Area Networks Specific Requirements.*

[25] Perahia, Eldad, and Robert Stacey. *Next Generation Wireless LANS*: 802.11n and 802.11ac. Cambridge University Press, 2013.

[26] *Arvindpdmn. (2021, December 19). 5G NR Hybrid Arq.* Devopedia. https://devopedia.org/5g-nr-hybrid-arq

[27] Mukherjee, Amitav. 2020. *"Hybrid ARQ Schemes."* In: Wiley 5G Ref: The Essential 5G Reference Online, John Wiley & Sons, May 16. doi: 10.1002/9781119471509.w5GRef015. Accessed 2021-12-18.

[28] Saatchi, Negin Sadat & Yang, Hong-Chuan & Liang, Ying-Chang. (2022). *Novel Adaptive Transmission Scheme for Effective URLLC Support in 5G NR: A Model-based Reinforcement Learning Solution.* IEEE Wireless Communications Letters. PP. 1-1. 10.1109/LWC.2022.3218488.

[29] Mnih, Volodymyr, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. *"Playing Atari with Deep Reinforcement Learning."* ArXiv:1312.5602 [Cs], December 19, 2013. https://arxiv.org/abs/1312.5602.

[30] *RlDQNAgent*. MathWorks. (n.d.). https://www.mathworks.com/help/reinforcement-learning/ug/dqn-agents.html?searchHighlight=DQN+agent&s_tid=srchtitle_support_results_1_DQN+agent

[31] *Reinforcement learning toolbox. Reinforcement Learning Toolbox Documentation.* (n.d.). https://www.mathworks.com/help/reinforcement-learning/index.html?searchHighlight=Reinforcement+Learning&s_tid=srchtitle_support_results_1_Reinforcement+Learning

[32] *RlQAgent.* MathWorks. (n.d.-b). https://www.mathworks.com/help/reinforcement-learning/ug/create-agents-for-reinforcement-learning.html?searchHighlight=Reinforcement+Learning&s_tid=srchtitle_support_results_3_Reinforcement%2520Learning

[33] Sutton, Richard S., and Andrew G. Barto. Reinforcement Learning: An Introduction. Second edition. Adaptive Computation and Machine Learning. Cambridge, Mass: The MIT Press, 2018.

[34] M. P. Mota, D. C. Araujo, F. H. C. Neto, A. L. de Almeida, and F. R. Cavalcanti, *“Adaptive modulation and coding based on reinforcement learning for 5G networks,”* in 2019 *IEEE Globecom Workshops (GC Wkshps)*. IEEE, 2019, pp. 1–6.

[35] 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)*. IEEE, 2012, pp. 809–814.

[36] S. Praveen, J. Khan, and L. Jacob, *“Reinforcement learning based link adaptation in 5G URLLC,”* in 2021 8th *International Conference on Smart Computing and Communications (ICSCC)*. IEEE, 2021, pp. 159–163.

[37] S. Satya Sri Ganesh Seeram, *‘Link Adaptation in 5G Networks : Reinforcement Learning Framework based Approach’*, Dissertation, 2022.

[38] 3GPP TR 38.901. *“Study on channel model for frequencies from 0.5 to 100 GHz.”* 3rd Generation Partnership Project; Technical Specification Group Radio Access Network.

[39] *DelayProfile. Model TDL MIMO channel model* - MATLAB. (n.d.). https://www.mathworks.com/help/5g/ref/nrtdlchannel-system-object.html?searchHighlight=TDL-C&s_tid=srchtitle_support_results_3_TDL-C

[40] *DL-SCH HARQ modeling. DL-SCH HARQ Modeling* - MATLAB & Simulink. (n.d.). https://www.mathworks.com/help/lte/ug/dl-sch-harq-modeling.html?searchHighlight=HARQ&s_tid=srchtitle_support_results_2_HARQ

[41] H. Dong, H. Dong, Z. Ding, S. Zhang, and Chang, *Deep Reinforcement Learning*. Springer, 2020.

[42] *Environments* - MATLAB & Simulink. (n.d.). https://www.mathworks.com/help/reinforcement-learning/environments.html