ULTRA-RELIABLE LOW-LATENCY INDUSTRIAL WIRELESS COMMUNICATIONS: OPTIMIZATION AND IMPLEMENTATION

By

Litianyi Zhang

A THESIS SUBMITTED IN FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY
AT
SCHOOL OF ELECTRICAL AND INFORMATION ENGINEERING
THE UNIVERSITY OF SYDNEY

MAR 2023

To my beloved mother Qingbo Li,
my dear father Tienan Zhang in loving memory

Acknowledgements

My Ph.D. started at the Centre of IoT and Telecommunications, School of Electrical and Information Engineering, University of Sydney four years ago. I have always regarded this journey as one of the most important, precious, and memorable experiences in my life. Despite the widespread global pandemic throughout my Ph.D. life, I was able to overcome all the challenges of research thanks to the valuable help from the marvellous people I met. I would like to express my sincere gratitude to those who have kindly supported me during this extremely tough period.

First and foremost, I would like to thank Prof. Yonghui Li for his comprehensive supervision. Prof. Yonghui guided me into the palace of academic research, which significantly changed my life. I would also like to thank Prof. Branka Vucetic, who always offered me great encouragement and support whenever I met any difficulty. Also, I am grateful for her funding for my Ph.D. study.

Secondly, I would like to thank Dr. Changyang She for his supervision in writing and publishing academic papers. From him, I learned how to effectively and efficiently build applicable ideas and launch research works. I am also grateful to Dr. Ying Kai, Dr. Yifan Gu, Dr. Zhibo Pang, and Dr. Kan Yu, who have provided me with abundant professional advice on my works and thesis.

Then, I would like to express my great gratitude to all my colleagues in the Telecommunications Lab at the University of Sydney. It was great working with all of you and sharing the laboratory together. Thanks Dr. Floriana Badalotti for your editorial assistance.

Finally, thanks must go to my family, my girlfriend, and my friends, who offered me unconditional love and support all the way. I would not have such an accomplishment without their help.

Litianyi Zhang
Sydney, Australia
March, 2023

Statement of Originality

The work presented in this thesis is the result of original research carried out by myself, in collaboration with my supervisors, while enrolled in the School of Electrical and Information Engineering at the University of Sydney as a Ph.D. candidate.
These studies were conducted under the supervision of Prof. Branka Vucetic, Prof. Yonghui Li, and Dr. Changyang She. It has not been submitted for any other degree or award in any other university or educational institution.

Litianyi Zhang
School of Electrical and Information Engineering
The University of Sydney
March, 2023

Acknowledgment of Authorship

I hereby certify that the work embodied in this thesis contains published papers/ scholarly work of which I am a joint author. I have included as part of the thesis a written declaration endorsed in writing by my supervisor, attesting to my contribution to the joint publications/scholarly work.

Chapters II and III of this thesis includes the content from paper [J1] and [J2], which are under review. Chapter IV of this thesis contains materials from paper [J3], which is published. I designed each algorithm, deduced all mathematical processes, and performed all simulations and hardware experiments.

Litianyi Zhang
Mar 21, 2023

By signing below I confirm that Litianyi Zhang contributed to all publications embodied in this thesis

---

Name of Supervisor: Branka Vucetic

Signature of Supervisor:

Date: Mar 21, 2023

Abstract

Ultra-reliable and low-latency communications (URLLC), which are critical scenarios in the fifth-generation (5G) and the upcoming sixth-generation (6G) mobile networks, are crucial for mission-critical services with stringent reliability and latency requirements. However, due to the limited time and frequency resources in URLLC transmissions, the decoding packet error probability (PEP) is unavoidable, making it extremely challenging to meet the reliability constraint. One key challenge is that the current wireless systems use pilot symbols for channel estimation, and these pilot symbols share the channel resources with data symbols. In practice, a limited number of pilot symbols also leads to inevitable channel estimation errors, resulting in imperfect channel state information (CSI) at the base station and the user. Increasing the number of pilot symbols would yield more accurate channel estimation, but the remaining number of symbols for data transmissions will be reduced, thereby reducing resource utilisation efficiency. Therefore, it is essential to develop effective solutions to enhance resource utilisation efficiency while satisfying the reliability requirement in URLLC systems.

Resource allocation strategies for multiple-input single-output (MISO) URLLC systems are first introduced in this thesis to maximise resource utilisation efficiency with imperfect CSI. In the first part, I focus on the independent and identically distributed (IID) Rayleigh fading channel realisations and propose unsupervised learning algorithms to estimate the resource allocation policy, considering two types of reliability constraints: average PEP and PEP outage probability requirements. I propose a model-based unsupervised learning algorithm for the scenario where PEP is measurable. For a more practical scenario where the base station can only have discrete observations of the PEP, I also design a model-free unsupervised learning algorithm. I validate my algorithms with the maximal-ratio transmission precoding and the codebook-based precoding defined in the 5G New Radio (NR) standard. I compare

the proposed methods with the existing benchmark and observe that even with a lower achievable signal-to-interference-plus-noise ratio (SINR), my methods can still remarkably improve resource utilisation efficiency.

Due to the short time scale of URLLC transmissions, the temporal correlation of channel realisations should be addressed. In the second part, I focus on the temporally correlated channel realisations and propose deep reinforcement learning (DRL) algorithms to acquire the resource allocation policy that can maximise long-term resource utilisation efficiency. I formulate the optimisation problem as a partial observation Markov decision process (POMDP) and develop a novel cascaded-Action Twin Delayed Deep Deterministic policy (CA-TD3) to solve the POMDP problem. I propose a primal CA-TD3 algorithm and compare it with the primal-dual method. I validate the algorithm on the first-order autoregressive channel model and the clustered delay line (CDL) channel. The results show that the primal CA-TD3 can achieve a more efficient convergence performance than the primal-dual algorithm in terms of reliability and resource utilisation efficiency.

As one of the most important use cases of URLLC, factory automation aims to deploy a massive number of Industrial Internet of Things (IIoT) devices and applications, which require reliable real-time services. Wireless time-sensitive networking (WTSN) is a promising solution to support factory automation. In the third part, I aim to design a hardware platform to implement WTSN, which needs to be low-cost, scalable, compatible with existing 802.11 devices, and easily deployable. I select a commercial 802.11-based platform to support high data rates and utilise a time division multiple access (TDMA) mechanism to schedule the transmissions to achieve deterministic latency. I propose two novel schemes to improve the latency and reliability performance: real-time quality of service (RT-QoS) and fine-grained aggregation (FGA). The experimental results show the superiority of my proposed protocol compared to the existing TDMA-based 802.11 system and legacy 802.11 system.

Finally, I conclude the thesis with a summary of the results and discuss the potential future directions for improving URLLC in 6G networks.

Table of Contents

Acknowledgements iii

Statement of Originality v

Acknowledgment of Authorship vi

Abstract vii

Table of Contents ix

List of Tables xii

List of Figures xiii

List of Acronyms xvi

List of Symbols and Notations xix

List of Publications xx

1 Introduction 1

1.1 Backgrounds . . . 1

1.1.1 Challenges in URLLCs . . . 2

1.1.2 Deep Learning for URLLCs . . . 4

1.1.3 Platform for URLLCs . . . 6

1.2 Literature Review . . . 7

1.2.1 Searching-based Resource Allocation for URLLCs . . . 7

1.2.2 Learning-Based Resource Allocation for URLLCs . . . 9

1.2.3 WTSN Implementation . . . 11

1.3 Research Problems and Contributions . . . 14

1.4 Thesis Outline . . . 19

2 Unsupervised Learning for URLLC with Practical Channel Estimation 21
2.1 Introduction . . . 22
2.2 System Model . . . 23
2.2.1 Beam Training Phase . . . 24
2.2.2 Channel Estimation . . . 26
2.2.3 Data Transmission . . . 27
2.3 Problem Formulation . . . 28
2.3.1 Resource Constraint and CSI Observations . . . 28
2.3.2 Optimisation Problem Formulation . . . 29
2.3.3 Primal-dual Approach Formulation . . . 32
2.4 Unsupervised Learning Approaches . . . 32
2.4.1 Model-Based Unsupervised Learning . . . 33
2.4.2 Model-Free Unsupervised Learning . . . 36
2.4.3 Reliability Evaluation with PEP Outage Probability Constraint 41
2.4.4 Deep Transfer Learning for Dynamic Radio Resources . . . 43
2.4.5 Complexity of Cascaded DNN . . . 44
2.5 Simulation Results . . . 44
2.5.1 System Setup . . . 45
2.5.2 Benchmark . . . 46
2.5.3 Codebook-Based Precoding . . . 46
2.5.4 Hyper-Parameters of Neural Networks . . . 48
2.5.5 Performance Evaluation for Average PEP Requirement . . . 49
2.5.6 Performance Evaluation for PEP Outage Probability Requirement . . . 52
2.5.7 Performance Evaluation for Transfer Learning . . . 54
2.5.8 Performance Evaluation with Different CSI Observations . . . 61
2.6 Conclusion . . . 62

3 Reinforcement Learning for Optimal URLLC Resource Efficiency under Correlated Channel 64
3.1 Introduction . . . 65
3.2 System Model . . . 67
3.2.1 Channel Model . . . 67
3.2.2 Reliability Metric with Imperfect CSI . . . 68
3.2.3 Constrained POMDP Problem Formulation . . . 69
3.3 Deep Reinforcement Learning . . . 72
3.3.1 CA-TD3 Architecture . . . 72
3.3.2 Primal-Dual CA-TD3 . . . 74

3.3.3 Primal CA-TD3 . . . 76
3.4 Simulation Results . . . 79
3.4.1 Simulation Setup . . . 79
3.4.2 Channel Models . . . 80
3.4.3 DRL Setup . . . 81
3.4.4 Performance Evaluation . . . 82
3.5 Conclusion . . . 87

4 Enabling Real-Time Quality-of-Service and Fine-Grained Aggregation for Wireless TSN 88
4.1 Introduction . . . 89
4.2 System Design and Implementation . . . 92
4.2.1 APP-Layer Configuration and RT-QoS . . . 92
4.2.2 FGA . . . 96
4.3 FGA Analysis and Numerical Results . . . 99
4.3.1 Trade-off Analysis of FGA . . . 99
4.3.2 Numerical Results . . . 103
4.4 Experiments and Results . . . 104
4.4.1 Experiment Design . . . 106
4.4.2 MAC-Layer Performance . . . 108
4.4.3 APP-Layer Performance . . . 112
4.5 Conclusions and Future Work . . . 115

5 Conclusions and Future Work 117
5.1 Summary of Results . . . 117
5.2 Future Work . . . 119

Bibliography 122

List of Tables

2.1 System parameters for simulation setup. 45
2.2 Hyper-parameters for the DNN structures 47
2.3 Resource utilisation efficiency and Average PEP in training and testing stages 52
2.4 Resource utilisation efficiency and PEP outage probability in training and testing stages 56
2.5 Testing performance with Different CSI Observations 57

3.1 Simulation parameters 81
3.2 Test results of resource utilisation efficiency and PEP outage probability with respect to different algorithms. 87

4.1 Experiment parameters preset for RT-WiFiQA. 108
4.2 Effective reliability with a MAC delay lower than a specific deadline value for four stations 111
4.3 Effective reliability with an APP delay lower than a specific deadline value for four stations 114

List of Figures

2.1 Frame structure of a downlink MISO system. The BS transmits the CSI-RS periodically. Within one CSI-RS period, there are multiple frames transmitted. In each frame duration, T_f, DM-RS, and a data packet are transmitted to the UE. After receiving CSI-RS and DM-RS, a CSI report is sent back to the BS. 25

2.2 Model-based cascaded DNN structure, where $g_{\mathbf{h}}^{(i,m)}$ is the m-th CSI observation in the i-th iteration, $N_{\mathrm{c}}^{(i,m)}$ and $D^{(i,m)}$ are the outputs of the two DNNs. 34

2.3 Model-free Cascaded DNN structure, where $g_{\mathbf{h}}^{(i,m)}$ is the m-th CSI observation in the i-th iteration, $\mu_N^{(i,m)}$ and $\beta_N^{(i,m)}$ are the outputs of the first DNN, $\mu_D^{(i,m)}$ and $\beta_D^{(i,m)}$ are the outputs of the second DNN, $\hat{N}_c^{(i,m)}$ and $\hat{D}^{(i,m)}$ are random samples captured from $\mathcal{N}(\mu_N^{(i,m)}, \beta_N^{(i,m)})$, and $\mathcal{N}(\mu_D^{(i,m)}, \beta_D^{(i,m)})$, respectively. 38

2.4 Reliability evaluation function for the m-th channel sample in the i-th iteration. 42

2.5 Resource utilisation efficiency in the training stage, where the average PEP requirement is considered. 50

2.6 Average PEP in the training stage, where the required average PEP is 10^{-5}. 51

2.7 Number of DM-RS symbols in the training stage, where the average PEP requirement is considered. 51

2.8 CDF of SINR, where the average SNRs of "Codebook" and "Benchmark" are 11.2 dB and 11.7 dB, respectively. 53

2.9 Resource utilisation efficiency in the training stage, where the PEP outage probability requirement is considered. 53

2.10 PEP outage probability in the training stage, where the required PEP outage probability is 10^{-4}. 55

2.11 Number of DM-RS symbols in the training stage, where the PEP outage probability requirement is considered. 55
2.12 Resource utilisation efficiency when $T_f = 0.6$ ms, where the initial DNNs are trained with $T_f = 1$ ms. 58
2.13 Average PEP performance when $T_f = 0.6$ ms, where the initial DNN is trained with $T_f = 1$ ms. 58
2.14 Resource utilisation efficiency when $n_t = 6$, where the initial DNNs are trained with $n_t = 4$. 59
2.15 Average PEP when $n_t = 6$, where the initial DNNs are trained with $n_t = 4$. 59
2.16 Resource utilisation efficiency over Nakagami-m fading channels, and the initial DNNs are trained over Rayleigh fading channels. 60
2.17 Average PEP over Nakagami-m fading channels, and the initial DNNs are trained over Rayleigh fading channels. 60
2.18 Resource utilisation efficiency versus the total number of symbols. . . 61

3.1 Resource allocation in temporally correlated channel realisations . . . 66
3.2 CA-TD3 Architecture 73
3.3 Mapping of pilot and data symbols in each resource grid. 79
3.4 PEP outage probability (defined as the cost) against training episodes for the primal-dual CA-TD3. 83
3.5 Resource utilisation efficiency (defined as the reward) against training episodes for the primal-dual CA-TD3. 84
3.6 PEP outage probability (cost) versus training episodes. 85
3.7 Resource utilisation efficiency (reward) versus training episodes. . . . 86

4.1 Overview of RT-WiFiQA architecture. 93
4.2 An example of RT-WiFiQA superframe design. 96
4.3 Typical FGA frame format. 98
4.4 PER against aggregated packet size $l_{a,i}$ for different l_i, where the critical thresholds can be evaluated based on Proposition 4.3.1. 103
4.5 PER against increasing BER for different total length $l_{a,i}$ of packets that can be aggregated to the i^{th} packet with l_i of 800 bits. 105
4.6 Experimental environment, which is designed to simulate a real industrial wireless setting. One AP and four stations are placed on the ground for testing. 106

4.7 MAC-layer EPLR CCDF curves of RT-WiFiQA, RT-WiFi and WiFi. 110
4.8 APP-layer EPLR CCDF curves of RT-WiFiQA, RT-WiFi and WiFi. 113

List of Acronyms

5G	fifth-generation
6G	sixth-generation
AC	access categories
ACK	acknowledgment
APIs	application programming interfaces
APP	application
AP	access point
AWGN	additive white Gaussian noise
BER	bit error rate
BS	base station
CA-TD3	cascaded-action TD3
CDF	cumulative distribution function
CCDF	complementary cumulative distribution function
CDL	clustered delay line
COTS	commercial off-the-shelf
CRPO	constraint-rectified policy optimization
CSI	channel state information
CSI-RS	channel state information reference signal
CSMA/CA	carrier sense multiple access with collision avoidance
DM-RS	demodulation reference signal
DNN	deep neural network

DRL	deep reinforcement learning
EPLR	effective packet loss ratio
emBB	enhanced mobile broadband
FCS	frame check sequence
FIFO	first-in-first-out
IID	independent and identically distributed
IIoT	industrial internet of things
IoT	internet of things
IRS	intelligent reflecting surface
MAC	medium access control
MCS	modulation and coding scheme
MIMO	multiple-input multiple-output
MISO	multiple-input single-output
MMSE	minimum mean-square-error
mMTC	massive machine-type communication
MU-MIMO	multi-user multiple-input-multiple-output
MRT	maximal-ratio transmission
NR	New Radio
OFDMA	orthogonal frequency-division multiple access
PDF	probability density function
PEP	packet error probability
PER	packet error rate
PLCP	physical layer convergence procedure
POMDP	partially observable Markov decision process
PTP	precision time protocol
QoS	quality-of-service
SINR	signal-to-interference-plus-noise ratio
SGD	stochastic gradient descent
SDR	software-defined radio

TD3	twin delayed deep deterministic policy gradient
TDMA	time division multiple access
TDL	tapped delay line
ToS	type of service
TSF	timing synchronization function
TTI	transmission time interval
TSN	time-sensitive networking
UE	user equipment
URLLC	ultra-reliable and low-latency communications
V2X	vehicle-to-everything
VR	virtual reality
WTSN	wireless time-sensitive networking
xURLLC	extreme URLLC

List of Symbols and Notations

X	a random variable
x	values of scalar variables or the sample of random variables
$\mathbb{E}[X]$	the mean of X
σ_X^2	the variance of X
$\mathbf{A}$	a matrix
$\mathbf{A}^{\mathrm{T}}$	the transposition of matrix $\mathbf{A}$
$\mathbf{A}^{\mathrm{H}}$	the Hermitian transposition of matrix $\mathbf{A}$
$\mathcal{A}$	a set
$\mathcal{N}(\cdot, \cdot)$	the Gaussian distribution
$\mathcal{CN}(\cdot, \cdot)$	the complex Gaussian distribution
I_n	$n \times n$ identity matrix
$\lvert \cdot \rvert$	the absolute value of a complex scalar
$\lVert \cdot \rVert$	the Euclidean vector norm
$\mathbb{1}\{\cdot\}$	the indicator function
$\lfloor \cdot \rfloor$	the floor function
$\nabla_x f(x)$	the gradient vector of function $f(x)$
$Q(\cdot)$	Q-function

List of Publications

The following is a list of publications in refereed journals and patents produced during my Ph.D. candidature.

Journal Papers

[J1]. L. Zhang, C. She, K. Ying, Y. Li and B. Vucetic, "Unsupervised learning for ultra-reliable and low-latency communications with practical channel estimation", IEEE Transactions on Wireless Communications, Accepted, Aug. 2023.
[J2]. L. Zhang, C. She, K. Ying, Y. Li and B. Vucetic, "Deep reinforcement learning for improving resource utilization efficiency of URLLC with imperfect channel state information," in IEEE Wireless Communications Letters, doi: 10.1109/LWC.2023.3294910.
[J3]. L. Zhang, Y. Gu, R. Wang, K. Yu, Z. Pang, Y. Li and B. Vucetic, "Enabling real-time quality-of-service and fine-grained aggregation for wireless TSN.", in Sensors 22(10): 3901, May 2022

Patents

The following patent is not included in this thesis.

[P1]. L. Zhang, Y. Gu, R. Wang, Y. Li and B. Vucetic (2022). "Centralized TDMA-based protocol for long-range WiFi communications", International Patent WO/2023/035044 .

Chapter 1

Introduction

In this chapter, I first introduce the background of my research and provide an extensive review of related works. Then, I specify the open research problems and stress my key motivations. Finally, I summarise my contributions and present the thesis outline.

1.1 Backgrounds

The fifth generation (5G) mobile communications enable three key services, which are enhanced mobile broadband (emBB) for high data rate mobile communications, ultra-reliable low-latency communication (URLLC) for mission-critical transmissions, and massive machine-type communication (mMTC) for Industrial Internet of Things (IIoT) applications [1, 2]. In recent years, URLLC has become increasingly important for supporting emerging wireless applications with stringent requirements, such as vehicle-to-everything (V2X) services, tactile Internet, remote telesurgery in Health 4.0, and wireless factory automation [3–7]. According to the 3GPP 5G New Radio (NR) standard [8, 9], the required packet error probability (PEP) of URLLC is lower than 10^{-5}, and the latency in the air interface does not exceed 1 ms.

1.1.1 Challenges in URLLCs

For the current URLLC system design, it is still challenging to meet the stringent requirements. To achieve low latency, the time duration for URLLC transmissions is very short. The conventional Shannon capacity, which assumes low PEP obtained with very long block lengths, cannot be used to characterise URLLC transmissions accurately. Practical URLLC transmissions of short packets are operated in a finite block length regime, and the PEP is unavoidable [10]. Several works have developed some solutions to improve the URLLC performance while satisfying PEP and channel resource constraints, including the number of resource blocks and transmission power budget [11–14]. However, most of these works achieved the PEP requirement at the cost of resource utilisation efficiency, which is defined as the number of information bits transmitted within the total number of resource blocks. How to improve the resource utilisation efficiency and meet the PEP requirement simultaneously remains an open issue.

One practical challenge of optimising the URLLC system is the imperfect channel state information (CSI) due to the use of pilot signals for channel estimation [15]. In the finite block length regime, the wireless system needs to allocate a limited number of resource blocks for pilot symbols, while the remaining number of symbols will be allocated for data transmission. The limited pilot symbols lead to unavoidable channel estimation errors, leading to imperfect CSI at the base station (BS) and the user equipment (UE). If the number of pilot symbols increases, the channel estimation will be more accurate [16], and the PEP will be reduced. However, the number of data symbols will decrease, leading to a lower data rate and resource utilisation efficiency. Therefore, considering the imperfect CSI, an optimal resource allocation policy for

pilot symbols and data symbols is important for improving resource utilisation efficiency.

Another challenge in optimising the resource utilisation efficiency is the randomness and error-prone features of the wireless channel. On the one hand, some works [17, 18] assume that the channel realisations can be acknowledged at the BS and UE. However, BS and UE normally only obtain partial channel observations rather than the full CSI. The partial information can be the channel gain, channel quality information (CQI), or the received signal strength (RSS), which can be acquired from the periodic CSI report. Wireless systems need to use limited channel observations to determine the resource allocation policy. On the other hand, to combat the randomness of channel realisations and additive white Gaussian noise (AWGN), most existing works [19, 20] determine the resource allocation according to the channel's statistical character. However, the algorithms based on statistical channel characters can hardly perform well in a randomly fast-varying channel. In other words, the resource allocation policy needs to be dynamic and adaptive for different channel realisations.

Besides the limited knowledge of wireless channels, the beam training procedure in the current 3GPP NR standard also introduces a sampling error. The aforementioned pilot signals refer to the CSI reference signal (CSI-RS) and the demodulation reference signal (DM-RS). The CSI-RS is transmitted periodically for beam training, where an optimal beamforming vector is selected for future DM-RS and data transmissions. Current wireless systems use codebook-based precoding techniques [21], where the precoding matrix is selected from a pre-defined codebook. This method can achieve fast decisions of the precoding matrix but at the cost of reduced accuracy compared

to the theoretical maximal-ratio transmission (MRT) beamforming.

In addition, a common assumption is that channel realisations are independent and identically distributed (IID). However, in the mobile URLLC scenario, the channels are highly correlated in the temporal domain due to the short time scale. It is costly to execute the resource allocation algorithm as long as the channel changes, especially for latency-sensitive URLLC transmissions. Thus, it is important to dynamically optimise the resource utilisation efficiency in correlated channels.

1.1.2 Deep Learning for URLLCs

Regarding the practical implementation, the conventional resource allocation algorithms cannot fully overcome the aforementioned challenges [2, 22]. One conventional approach with low complexity is to derive the closed-form expressions of performance metrics in URLLC. However, such methods are mostly based on assumptions that are inaccurate for URLLC applications. Due to the unknown channel estimation errors, the closed-form results may not be derivable. Another approach is to apply conventional optimisation algorithms to find the optimal policy. These methods do not require strong assumptions and can obtain the optimal policy based on bisection-searching methods. However, the wireless system needs to execute the resource allocation algorithm according to the varying channels. Thus, these solutions introduce very high computing overhead, which is too complicated to be implemented in URLLC applications.

To improve the performance of URLLC systems in 6G networks, deep learning is a promising technique to obtain a near-optimal solution [2]. Primarily, I train a deep neural network (DNN) in an offline mode. The trained DNN represents the mapping

from channel observations to the near-optimal strategies in communication systems. In [23], Dong *et al.* analysed the complexity of the forward propagation algorithm, which is much lower than the conventional searching-based algorithm. Therefore, deep learning methods can be used in a real-time scenario. Additionally, unlike searching for an optimal solution for a given state in conventional optimisation algorithms, deep learning algorithms are data-driven and can explore the optimal policies numerically.

Despite the remarkable advantages of deep learning methods, how to satisfy stringent URLLC constraints and achieve optimal transmission performance in nonstationary networks is yet to be developed. Initially, supervised learning methods were applied in the wireless systems [24–26]. The idea is to generate labels through the conventional optimisation solution and train the DNN with stochastic gradient descent (SGD) to minimise the empirical mean square errors between the output of the DNN and the labels. However, in practical systems, labelled real channel data is usually unavailable, where such ideas of "learning to optimise" cannot be implemented.

To find the near-optimal policy without labelled data, there are two approaches: unsupervised learning and deep reinforcement learning (DRL). Unsupervised learning aims to solve a nondeterministic problem and obtain a real-time resource allocation strategy. The unsupervised learning model utilises batch samples of limited channel observations and estimates the policy through a DNN [27, 28]. To train the unsupervised learning model, the optimisation problem can be modified as the loss function reflecting the design goal. Then, I can use SGD algorithms to train the DNN parameters until the optimisation function converges. DRL is developed to maximise the long-term reward for a Markov decision process, which can be applied to improve resource utilisation efficiency in correlated channels [29]. Since the wireless system

can only have partial channel observations, the optimisation of URLLC is normally formulated as a partially observable Markov decision process (POMDP). During the exploration stage, conventional DRL algorithms try random actions to estimate the long-term reward. However, some of the bad actions taken by the agent can destroy the quality-of-service (QoS) requirements and result in unexpected accidents in URLLC systems. To ensure exploration safety, constrained DRL algorithms are applied to obtain the optimal policy of URLLC systems. If I use theoretical models to analyse the channel estimation errors and PEP, model-based solutions can be implemented in either unsupervised learning or DRL. Otherwise, I can apply model-free methods to train the unsupervised learning or DRL model based on the discrete observation values of SNR and PEP [28]. If the practical channel distribution is different from the training stage, deep transfer learning techniques can be applied, which improves the training efficiency. However, how to utilise learning-based data-driven methods to improve the resource utilisation efficiency of practical URLLC systems remains unclear.

1.1.3 Platform for URLLCs

While most research on URLLC systems is theoretically feasible, implementation of URLLC on a hardware platform is another important aspect of 6G networks. As one important extensional application of URLLC, real-time IIoT communication services in the factory automation and manufacturing industry have drawn great attention in recent years [30]. Different from the Internet of Things (IoT) scenario, IIoT communications also have stringent requirements of latency and reliability.

The conventional solution for real-time industrial applications is Time-Sensitive

Networking (TSN), proposed by IEEE 802.1 TSN Task Group [31]. Since the wired solution cannot meet the scalability and flexibility requirements, the development of wireless TSN (WTSN) is promoted for enabling URLLC in 6G networks. However, existing wireless solutions can hardly meet the data rate, latency, and reliability constraints of real-time services at the same time. Although recent works based on software-defined radio (SDR) [32, 33] can achieve remarkable latency and reliability performance, their systems need sophisticated modifications on the physical (PHY) layer, which is not fully compatible with existing wireless devices and is costly for large-scale deployment. Therefore, I am also motivated to provide such a hardware-based platform for WTSN.

1.2 Literature Review

In this section, I first specify the resource allocation problems in URLLC systems and introduce the existing works, including conventional methods based on exhaustive search and emerging learning-based solutions. Then, I focus on the scenario of factory automation and the manufacturing industry and provide a comprehensive review of practical solutions for WTSN.

1.2.1 Searching-based Resource Allocation for URLLCs

Resource allocation of URLLC systems has been extensively investigated in the existing literature. Sun *et al.* [13] developed an optimisation algorithm to find the global optimal resource allocation in the finite blocklength regime, where the optimisation problem is non-convex. Ren *et al.* [17] proposed an optimisation algorithm

that jointly optimises the blocklength and power allocation to minimise the decoding error probability in factory automation. Salah *et al.* [12] designed a retransmission scheme and optimised radio resource allocation to meet the QoS requirements of URLLC systems. Walid *et al.* [18] designed a resource allocation algorithm for a downlink multiple-input single-output (MISO) URLLC system with multiple UEs. In terms of formulating deterministic optimisation problems, the above work relies on the assumption that perfect CSI is available at the transmitter and receiver.

Considering the overhead for channel estimation, Zeng *et al.* [20] optimised the pilot length to minimise the PEP in multi-user multiple-input-multiple-output (MU-MIMO) uplink communications. Schiessl *et al.* [34] investigated the delay performance of MISO systems in the finite blocklength regime with imperfect CSI. Lin *et al.* [19] developed a low-complexity algorithm to optimise resource allocation for channel estimation and data transmission in URLLC systems. Since channel estimation errors are unknown to the communication systems, the relationship between imperfect CSI observation and PEP does not have a closed-form expression. In order to design an optimisation algorithm, some assumptions and theoretical models on the distribution of channel estimation errors are needed, which, however, may not hold in practical systems.

Existing works [12, 13, 17, 18] mostly assume that the channel realisations are IID. However, due to the short time scale of URLLC, the coherent channel realisations show a strong correlation [35], which cannot be neglected. Moreover, the distribution of correlated channels may change during one frame duration. As a result, the resource allocation policy obtained from IID channels cannot guarantee optimal performance within the coherence time. Therefore, how to design a resource

allocation policy that can be effective for temporally correlated channels is still an unsolved issue. Librino *et al.* [35] investigated the impact of channel time correlation in the uplink scenario with a massive number of sensors, where the CSI is imperfect and updated less frequently. Ren *et al.* [36] analysed the decoding error probability and data rate of correlated channels in an intelligent reflecting surface (IRS). Cao *et al.* [37] proposed a joint design of resource allocation with respect to pilot symbols and data symbols for multi-device URLLC systems in temporally correlated channels. From these works, we can observe the importance of channel correlation, and also, the first-order autoregressive channel model is widely used as a simple temporally correlated channel model. However, how to optimise the resource allocation efficiency in practical correlated channel realisations was not fully explored in these works.

1.2.2 Learning-Based Resource Allocation for URLLCs

Deep learning has been considered as a promising method for real-time resource allocation in wireless systems. The idea is to approximate the resource allocation policy by using a DNN. Dong *et al.* [23] proposed a cascaded structure of neural networks with deep transfer learning to meet diverse QoS requirements in 5G communication systems. Sun *et al.* [24] used a DNN to approximate a signal processing task over interference-limited channels. Liu *et al.* [25] developed a learning-based approach for the constrained energy minimisation problem in generic multi-dimensional networks. However, such supervised learning algorithms cannot be applied to practical wireless systems without labelled data.

To combat the need for real data, unsupervised learning methods have attracted wide attention. Liang *et al.* [27] conducted an overall review of applying deep learning

methods for resource allocation in vehicular networks. Initially, deep supervised learning was applied to wireless systems. Sun *et al.* [24] proposed a learning-based solution to find a mapping relationship from the environmental parameters to the optimal decision using unsupervised deep learning. Li *et al.* [14] developed a learning-based power control policy for securing transmissions of short packets in URLLC. For a general optimal resource allocation problem, Eisen *et al.* [28] designed a primal-dual training method to train a DNN, which can be either model-based or model-free. Xia *et al.* [38] constructed beamforming neural networks for different optimisation problems in multi-user MISO systems. Mismar *et al.* [39] applied deep reinforcement learning to maximise the signal-to-interference-plus-noise ratio (SINR) by jointly designing beamforming, power control, and interference coordination. Nevertheless, the impact of channel estimation errors on the PEP of URLLC is not investigated in the above works. How to guarantee the QoS of URLLC with deep learning in the presence of channel estimation errors and signalling overhead remains unclear.

On the other hand, many optimisation problems in URLLC systems are sequential decision-making problems, such as resource allocation in temporally correlated channels. These problems can be solved through DRL algorithms, which do not require labelled data. Dong *et al.* [40] adopts deep Q-learning [41] to improve mobile edge computing system performance through its digital twin. In [42], the authors proposed a DRL-based intelligent link adaptation in a time-correlated and fast-fading channel. Saatchi *et al.* [43] proposed a joint design of reliability and latency to maximise the successful packet ratio by using a model-based DRL technique. Alsenwi *et al.* [44] developed a DRL algorithm for resource allocation between enhanced Mobile Broad Band (emBB) and URLLC traffic, with the target of maximising the average data rate

of emBB users. However, in these works, no URLLC system constraints were considered in the optimisation problems. Considering the URLLC constraints, Meng *et al.* [45] built a constrained DRL framework based on primal-dual methods for reducing the tracking error between a robotic system and its digital model in the metaverse, which further proves the effectiveness of DRL in solving optimisation problems with constraints. Li *et al.* [46] designed a constrained DRL framework for low-latency wireless virtual reality (VR) applications. Nevertheless, the primal-dual-based DRL reveals significant limitations, including slow convergence of the constraint condition and difficult parameter tuning.

1.2.3 WTSN Implementation

Some recent wireless protocols have been carried out to meet the stringent latency and reliability requirements of WTSN. WirelessHP [33] and w-SHARP [32] were proposed based on software-defined radio (SDR). Both the physical (PHY) and medium access control (MAC) layers were optimised to achieve μs-level latency and packet loss ratio lower than 10^{-6}. However, these solutions have low compatibility with existing wireless standards and are very costly to be implemented in practical systems. There are also some works improving the MAC layer of the existing IEEE 802.15.4 protocol, such as WirelessHART [47] and ISA100.11a [48]. However, the data rate of IEEE 802.15.4 is only up to 250 kb/s and cannot satisfy the requirements of high-rate transmissions in many IIoT applications. In addition, reconfigurable intelligent surface and satellite-terrestrial networks are also investigated to improve the transmission reliability in IIoT networks [49, 50]. However, these solutions may require additional hardware equipment, which can increase the cost and system complexity.

Recently, URLLC has been proposed by 3GPP [51, 52]. However, the existing TSN is established on the 802 link layers, which is not fully compatible with the 3GPP-based 5G standard [53].

Due to the advantages of compatibility, cost, high rate, etc., numerous research works focused on the modification of IEEE 802.11 protocols based on the commercial off-the-shelf (COTS) network interfaces toward WTSN. Several studies focused on designing QoS schemes to support real-time data delivery. For instance, SchedWiFi, introduced in [54], is a novel traffic classification system based on access categories (AC) for ad-hoc IEEE 802.11 networks. SchedWiFi utilises a window mechanism to minimise interference between scheduled traffic and others. Similarly, in [55], multiple MAC schemes that support traffic with varying time and safety requirements were proposed. Real-time traffic is isolated from other traffic and transmitted within specific periods. However, the proposed QoS-based protocols in [54, 55] cannot provide a deterministic communication pattern as in TDMA-based systems.

In order to improve the reliability of industrial Wi-Fi networks, the authors in [56] proposed Wi-Fi Redundancy (Wi-Red) solution to offer seamless link-level redundancy. However, each independent Wi-Fi network in Wi-Red still uses legacy carrier sense multiple access with collision avoidance (CSMA/CA), which cannot guarantee deterministic latency. In [57], the authors proposed the RT-WiFi protocol, which designed a scheduler to allocate a sequence of time slots for each station and can achieve a sampling rate of 6 kHz. The authors in [58] proposed the Soft-TDMAC based on time division multiple access (TDMA) protocols to improve synchronisation precision. The authors in [59] conducted TDMA scheduling implementation on COTS hardware with support for multi-hop networks, namely Det-WiFi. These protocols

utilised TDMA to guarantee latency without considering efficiency or reliability optimisation. Very recently, the authors in [60] designed HAR^{2}D-Fi to provide reliable and deterministic communication based on the latest IEEE 802.11ax protocol. Unlike the studies above, we focus on the implementation of QoS and aggregation mechanisms for WTSN. The proposed schemes are validated on COTS hardware platforms through a real channel environment, while HAR^{2}D-Fi was only validated through simulation.

To improve the transmission efficiency in WTSN, applying aggregation schemes can be an effective and promising solution. Aggregation schemes were initially proposed in the conventional wired TSN, namely the Link Aggregation Control Protocol [61], but it cannot be extended to WTSN directly because of the error-prone features of wireless channels. In 802.11 n, A-MPDU and A-MSDU schemes can aggregate packets towards a single destination and are designed for throughput maximisation. However, A-MPDU and A-MSDU schemes cannot achieve low latency because of the time required for generating the aggregated packet [62]. Moreover, the packet aggregation mechanism proposed for WirelessHART focused on the 802.15.4 ad-hoc mode and cannot be implemented on the considered 802.11 infrastructure mode [63–65]. Additionally, WIA-FA proposed a similar aggregation mechanism for 802.11 interfaces [66]. However, the detailed implementation with application programming interfaces (APIs) is not designed and discussed. Besides, the aforementioned critical trade-off in determining whether or not to use aggregation is not analysed.

1.3 Research Problems and Contributions

In the previous section, I presented a review of resource allocation problems in URLLC. It's evident that the performance of URLLC systems is still limited in terms of resource utilisation efficiency and training complexity. Therefore, how to achieve optimal resource utilisation efficiency in URLLC while meeting the reliability requirements remains an unresolved issue, especially for a practical communication system adhering to the 5G NR design. To answer this question, I leverage the advantages of deep learning and reinforcement learning algorithms and investigate different resource allocation policies considering various scenarios. I also illustrated existing implementation solutions for achieving WTSN in factory automation and the manufacturing industry, which is an essential aspect of URLLC. Providing a reliable and scalable testbed that can efficiently schedule transmissions in a short time scale remains challenging. Hence, I also provided a hardware-based solution for WTSN, functioning as a testbed for URLLC.

In the first research problem (Chapter 2), I focus on designing a resource allocation policy for practical URLLC systems based on unsupervised learning. Most of the existing resource allocation policies for URLLC were derived from theoretical analysis and optimisation. Given inaccurate channel estimations, the PEP does not generally have a closed-form expression. To derive analytical results, I require certain assumptions and approximations, which could significantly impact the reliability of URLLC. To avoid unrealistic assumptions, I account for practical channel estimation and aim to maximise the number of bits that can be transmitted in one codeword, while adhering to a PEP constraint. The contributions are summarised as follows:

- I propose a framework for optimising the resource allocation policy under a PEP

constraint in URLLC systems, considering different types of CSI observations: 1) perfect channel gain, 2) estimated channel gain, 3) received signal strength. To improve resource utilisation efficiency, I maximise the number of bits that can be transmitted over a given amount of time and frequency resources by optimising resource allocation and packet size. Specifically, I consider two types of reliability requirements, i.e., average PEP and PEP outage probability, for different URLLC applications. Additionally, both MRT and codebook-based precoding techniques are taken into account.

- I develop model-based and model-free unsupervised learning algorithms to solve the problem. The model-based algorithm evaluates the PEP using the theoretical results in [15]. The model-free algorithm solely relies on practical observations of PEP and doesn't require any model. In both unsupervised learning methods, I design a cascaded DNN structure to approximate the optimal policy. The first DNN maps the CSI observation to the number of symbols for channel estimation. The second DNN maps the output of the first DNN to the number of bits that can be transmitted in the block, guiding the design of modulation and coding scheme. For practical URLLC systems with dynamic blocklengths, transfer learning is applied to fine-tune DNN parameters.

- In the scenario with a PEP outage probability constraint, I use an indicator to represent whether the PEP is satisfied. Since the indicator function is binary, with values of zero or one, updating the training parameters of the cascaded DNN via gradient descent isn't stable, and numerous samples are required for reliable evaluation. To address this issue, I develop a DNN for reliability evaluation, integrating its training into unsupervised learning algorithms. This DNN

takes CSI observation and the cascaded DNN output as inputs, producing the PEP outage probability. This approach allows me to compute the gradient of the PEP outage probability with respect to cascaded DNN parameters using back-propagation.

- Valuable insights are gained from my simulation results. I first demonstrate a significant performance gap between MRT and codebook-based precoding techniques (a 40% difference in resource utilisation efficiency). This suggests the potential to enhance resource utilisation efficiency by 40% through improved codebook design. My testing results show that the model-free algorithm can achieve near-optimal resource utilisation efficiency compared to the model-based algorithm, although the latter requires more training samples. Additionally, by replacing received signal strength with estimated channel gain, it's possible to increase resource utilisation efficiency by 10%. I also verify the effectiveness of transfer learning in scenarios with varying blocklengths, indicating that transfer learning reduces convergence time by 70%. Finally, I compare the learning methods with a benchmark method that maximises the number of symbols for data transmission. My methods improve resource utilisation efficiency by three to eight times, as the SINR of the benchmark exhibits a much longer tail distribution than my methods.

In the second research problem (Chapter 3), I aim to find a resource allocation policy for correlated fading channels such that the wireless system can achieve optimal resource utilisation efficiency and satisfy a PEP requirement. The contributions are summarised as follows:

- Due to the unknown channel realisation and imperfect CSI, the BS can only

obtain limited observations of the CSI. Therefore, I formulate the sequential decision problem with channel variation as a POMDP. My focus is on enhancing resource utilisation efficiency under the finite blocklength regime, considering instant channel estimation errors and random AWGN. I define the PEP outage probability as the reliability performance constraint.

- I design a novel DRL framework for the optimisation problem. Specifically, I utilise the Twin Delayed Deep Deterministic policy (TD3) structure to explore actions and estimate long-term reward and cost. Since I need to determine dual actions (i.e., the policy of resource allocation and packet size), I develop a cascaded DNN structure for action selection, referred to as cascaded-action TD3 (CA-TD3).

- I propose two DRL training algorithms: primal-dual CA-TD3 and primal CA-TD3. The primal-dual CA-TD3 involves a Lagrangian multiplier to combine the constraint and objective function into a single optimisation problem. However, the conventional primal-dual method has limitations, including slow convergence, instability, and challenging parameter tuning. To overcome these challenges, I first enhance the primal-dual algorithm with normalisation coefficients. Furthermore, I develop primal CA-TD3 based on the constraint-rectified policy optimisation (CRPO) method [67].

- I compare the primal-type algorithm with the primal-dual method using the first-order autoregressive channel model [68] and the CDL channel model [69]. The first-order autoregressive channel is a conventional theoretical model for temporally correlated channel realisations. The CDL channel is a link-level

channel generator defined in 3GPP TR 38.901, widely used for practical correlated channel simulation. My numerical results demonstrate that both primal-dual CA-TD3 and primal CA-TD3 can satisfy the constraint-based policy. Compared to the enhanced primal-dual CA-TD3, primal CA-TD3 achieves faster convergence in terms of PEP outage probability and resource utilisation efficiency. Finally, I present the testing results based on trained models, which significantly outperform the existing benchmark.

In the third research problem (Chapter 4), I propose the real-time WiFi protocol with Quality of Service and aggregation (RT-WiFiQA) by introducing two novel schemes to enhance the performance of the TDMA-based 802.11 systems: real-time Quality of Service (RT-QoS) and fine-grained aggregation (FGA). The contributions of this work are summarised as follows:

- I propose the RT-WiFiQA protocol with RT-QoS and FGA mechanisms to improve the performance in terms of latency and reliability on the TDMA-based 802.11 system. I also implement the developed schemes on COTS 802.11 interfaces. A detailed implementation with APIs is also provided.

- I analytically demonstrate that the FGA mechanism can outperform the system without aggregation in terms of latency and reliability when the FGA packet size is smaller than a critical threshold. Numerical simulations are also conducted to validate my theoretical analysis, and the evaluated critical threshold is applied in the practical FGA implementation.

- I perform extensive experiments to measure the Application-Layer (APP-layer) and the Medium Access Control Layer (MAC-layer) latency and reliability on

my hardware platform. The experimental results demonstrate the superiority of the proposed RT-WiFiQA protocol compared to the existing TDMA-based 802.11 protocol and the conventional 802.11 protocol.

1.4 Thesis Outline

The rest of the thesis is organised as follows. Chapter 2 describes a resource allocation method for URLLC based on unsupervised learning, where practical beam training and channel estimation defined in the 5G NR standard are considered. I aim to obtain optimal resource utilisation efficiency while meeting the reliability requirement. Both model-based and model-free unsupervised learning algorithms are introduced. In the numerical results, I comprehensively compare the model-based method, model-free method, and existing benchmark.

Chapter 3 further extends the resource allocation problem to the correlated-channel scenario, where I investigate the time-varying fading features and aim to explore the optimal resource utilisation efficiency during the coherence time. I adopt the DRL analysis frameworks, which are primal CA-TD3 and primal-dual CA-TD3. The simulation results show that both algorithms can achieve constraint-satisfying performance. However, the primal CA-TD3 has faster convergence and better control of constraints than the primal-dual method.

In Chapter 4, I develop a hardware-based testbed to implement a URLLC practical system under the scope of factory automation. I take cost, implementation complexity, scalability, and compatibility into consideration, then design a real-time 802.11-based system with COTS hardware. The proposed platform takes advantage of TDMA to guarantee the deterministic transmission pattern and integrates FGA

and RT-QoS schemes to improve reliability and latency performance.

Finally, Chapter 5 provides the conclusion of the thesis and presents some future directions.

Chapter 2

Unsupervised Learning for URLLC with Practical Channel Estimation

In this chapter, I optimise the resource allocation for channel estimation and data transmission, as well as the packet size, to maximise the resource utilisation efficiency while adhering to the constraints of URLLC. With practical channel estimation, the packet error probability (PEP) does not possess a closed-form expression. To address this issue, I develop novel model-based and model-free unsupervised deep learning algorithms to train a deep neural network for resource allocation and data transmission. Two types of reliability constraints are considered over a wireless link: 1) an average PEP constraint; 2) a constraint on the probability that PEP exceeds a certain threshold. The simulation results demonstrate that the learning algorithms can satisfy both types of reliability constraints. When compared with a benchmark approach that maximises the number of symbols for data transmission and employs maximum ratio transmission precoding, the learning method utilizing codebook-based precoding

achieves a lower average signal-to-interference-plus-noise ratio (SINR) while improving the resource utilisation efficiency by a factor of three. This disparity arises because the resource utilisation efficiency of URLLC is dominated by the tail distribution of SINR, rather than the average SINR. Furthermore, the benchmark's SINR exhibits a much longer tail distribution compared to the learning method.

2.1 Introduction

To guarantee the latency and reliability requirements of URLLC, I need to sacrifice the resource utilisation efficiency by encoding less information in each codeword. The fundamental trade-offs between reliability, latency, and resource utilisation efficiency were first obtained in [10]. Specifically, given the time and frequency resources (the number of modulation symbols [70]) and the target PEP, the maximum achievable data rate over an additive white Gaussian noise (AWGN) channel was derived. The achievable data rate in the finite blocklength regime was further extended into multi-antenna systems [15]. Based on these results, radio resource management has been widely investigated in different communication systems [11–13, 17, 18].

Different from most high data rate services, the block lengths of URLLC are short, e.g., a few hundred symbols. In the short block length regime, the overhead for channel estimation is not negligible [16, 19]. Nevertheless, to obtain analytical results, most of the existing works assumed that perfect channel state information (CSI) is available at the transmitter and the receiver. As defined in the 5G NR standard, a channel state information reference signal (CSI-RS) and a demodulation reference signal (DM-RS) are transmitted to the user equipment (UE) for channel estimation. Once the UE has the estimated CSI, a channel report is sent back to the base station

(BS). Due to estimation and quantisation errors, the BS only has limited observations of CSI [71]. To maximise resource utilisation efficiency, I need to optimise the number of symbols allocated for channel estimation and data transmission.

With traditional optimisation algorithms, the BS adjusts resource allocation according to the observation of CSI, which is updated every few milliseconds. In other words, the BS executes optimisation algorithms once the CSI changes. As a result, the computing overhead is exceptionally high. A novel approach is to approximate the mapping from the CSI observation to the resource allocation policy by a deep neural network (DNN) [24]. After offline training, the BS only needs to execute the forward propagation algorithm to obtain the resource allocation. Nevertheless, there are no labeled training samples in practical communication systems in general. To address this issue, I adopt unsupervised deep learning to optimise wireless communication systems [28]. This approach works well in deterministic optimisation problems with perfect CSI. When the CSI observation is inaccurate, the optimisation problems are non-deterministic. Achieving the target PEP of URLLC with inaccurate CSI observation remains a challenging issue.

2.2 System Model

I consider a downlink MISO system, where a BS with n_t transmit antennas serves a single-antenna UE. It can be easily extended to multi-user scenarios by using orthogonal frequency division multiple access. The channel between the BS and UE is assumed to be quasi-static since the transmission duration of each packet is much smaller than the channel coherence time in URLLC. I denote $\mathbf{h} \in \mathbb{C}^{n_t \times 1}$ as the small-scale channel coefficients, which follow the circularly symmetric complex Gaussian

distribution, i.e., $\mathbf{h} \sim \mathcal{CN}(0, \sigma_{\mathbf{h}}^2)$, where $\sigma_{\mathbf{h}}^2$ is the variance. The large-scale channel gain is denoted by α, which varies slowly and is known at the BS and the UE. I assume that there is a maximum transmit power constraint. In order to maximise the time and frequency resource utilisation efficiency, the maximum transmission power is used in both channel estimation and data transmission. When the BS transmits signal $\mathbf{x}$ with the maximum transmission power $p \in \mathbb{R}^+$, the received signal $\mathbf{y}$ at the UE can be expressed as

$$\mathbf{y} = \sqrt{\alpha p}\mathbf{h}^H\mathbf{w}\mathbf{x} + \mathbf{z}, \tag{2.2.1}$$

where $\mathbf{z} \sim \mathcal{CN}(0, \sigma_{\mathbf{z}}^2)$ is the AWGN, and $\mathbf{w}$ is the unit-norm beamforming vectors. With perfect CSI, I know that MRT, i.e., $\mathbf{w_m} = \frac{\mathbf{h}}{\|\mathbf{h}\|} \in \mathbb{C}^{n_t \times 1}$, is the optimal beamforming vector that maximises the received signal power. However, to reduce the CSI report's overhead, the 5G NR introduces codebook-based precoding vectors [72], and the BS chooses the best beamforming vector from the codebook that achieves the highest SINR.

In my system model, I follow the Physical Downlink Shared Channel procedure specified in 5G NR standard [21]. The channel estimation and data transmission include three phases: 1) the transmitter sends a CSI-RS periodically over multiple beams for precoding vector selection; 2) with the selected precoding vector, a DM-RS is transmitted for channel estimation; 3) with the selected beamforming vector, the BS transmits data symbols to the UE. The system model is shown in Fig. 2.1.

2.2.1 Beam Training Phase

In the beam training phase, the CSI-RS is transmitted periodically, and the beamforming vector can be updated at the beginning of each CSI-RS period. A CSI-RS

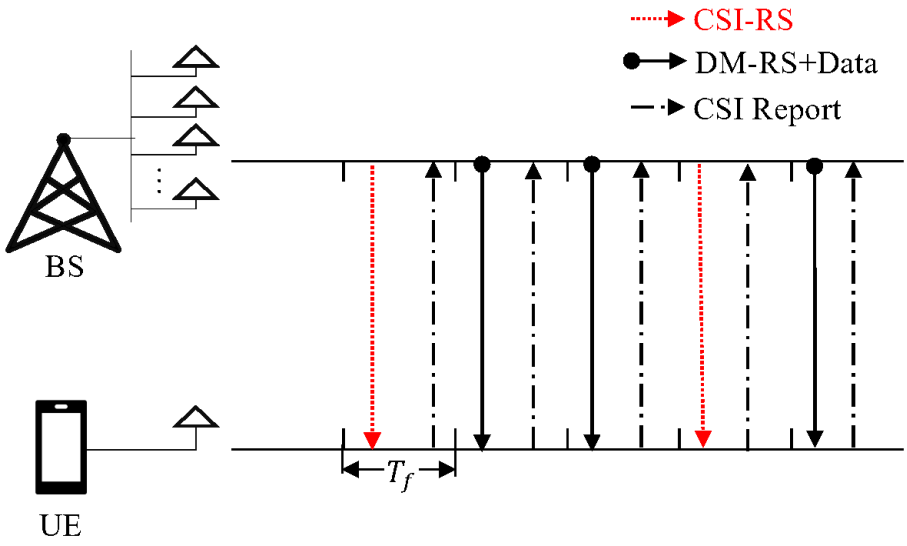


Figure 2.1: Frame structure of a downlink MISO system. The BS transmits the CSI-RS periodically. Within one CSI-RS period, there are multiple frames transmitted. In each frame duration, T_f, DM-RS, and a data packet are transmitted to the UE. After receiving CSI-RS and DM-RS, a CSI report is sent back to the BS.

period consists of multiple frames with duration T_f. Within each frame, a DM-RS and a data packet are transmitted to the UE. The beamforming vector is selected from a pre-defined codebook. The CSI-RS sequence and the codebook are known to the BS and the UE. The beam training procedure consists of three steps [72]:

1. *Beam Sweeping:* CSI-RS sequence, denoted as $\mathbf{x}_{\mathrm{b}}$, is transmitted multiple times using different beamforming vectors in the codebook with the maximum transmit power p. Let $\mathcal{S}_{\mathrm{b}}$ denote the set of beamforming vectors. When the beamforming vector $\mathbf{w}_i$ is used, the received signal $\mathbf{y}_{\mathrm{b,i}}$ at the UE can be obtained from (2.2.1).

2. *Beam Selection:* After the BS scans multiple beamforming vectors in the codebook, the UE selects the best beamforming vector that can achieve the highest SINR [73]. The selected beamforming vector, $\mathbf{w_f}$, can be expressed as

$$\mathbf{w_f} \triangleq \arg\max_{i \in \mathcal{S}_\mathrm{b}} \frac{\left|\mathbf{x}_\mathrm{b}^H \mathbf{y}_{\mathrm{b,i}}\right|^2}{\sigma_\mathbf{z}^2 \left\|\mathbf{x}_\mathrm{b}\right\|_2^2}. \tag{2.2.2}$$

3. *Beam Report:* The CSI report includes a channel quality indicator and a precoding matrix indicator of $\mathbf{w_f}$. Based on the channel quality, the BS adjusts the resource allocation policy and the modulation and coding scheme (MCS).

2.2.2 Channel Estimation

DM-RS, denoted by $\mathbf{x}_\mathrm{c}$, is a sequence of symbols, known by the BS and the UE, for channel estimation. I assume that the DM-RS is also transmitted with the maximum transmit power p. With the selected precoding vector $\mathbf{w_f}$, the received signal $\mathbf{y}_\mathrm{c}$ of DM-RS can be obtained from (2.2.1). Then, the UE performs channel estimation based on the received signal and the DM-RS. In this chapter, I consider the minimum mean-square-error (MMSE) channel estimation [16]. The estimated channel coefficient and the channel estimation error are denoted by $\hat{\mathbf{h}} \in \mathbb{C}^{n_t \times 1}$ and $\mathbf{e} \in \mathbb{C}^{n_t \times 1}$, respectively. Their relationship is given by $\mathbf{e} = \mathbf{h} - \hat{\mathbf{h}}$ [16], where $\mathbf{e}$ and $\hat{\mathbf{h}}$ follow circularly symmetric complex Gaussian distributions, i.e., $\mathbf{e} \sim \mathcal{CN}(0, \sigma_e^2)$ and $\hat{\mathbf{h}} \sim \mathcal{CN}(0, \sigma_{\hat{h}}^2)$. With the MMSE channel estimation, the variance of the channel estimation error is given by [16]

$$\sigma_\mathbf{e}^2 = (\frac{1}{\sigma_\mathbf{h}^2} + \frac{N_\mathrm{c} p}{\sigma_\mathbf{z}^2 n_t})^{-1}, \tag{2.2.3}$$

where N_c is the number of symbols allocated for DM-RS. N_c is required to be larger than n_t for a reliable channel estimation [16].

2.2.3 Data Transmission

The BS encodes D bits of data into N_{d} symbols by using a certain MCS. With the selected precoding matrix, $\mathbf{w_f}$, the received data symbols $\mathbf{y}_{\mathrm{d}}$ can be expressed as follows:

$$\mathbf{y}_{\mathrm{d}} = \sqrt{\alpha p}\hat{\mathbf{h}}^H \mathbf{w_f} \mathbf{x}_{\mathrm{d}} + \sqrt{\alpha p}\mathbf{e}^H \mathbf{w_f} \mathbf{x}_{\mathrm{d}} + \mathbf{z}, \tag{2.2.4}$$

where $\mathbf{x}_{\mathrm{d}}$ denotes normalised data symbols with $\|\mathbf{x}_{\mathrm{d}}\|_2^2 = 1$. Finally, the UE will decode the data symbols and recover the original information bits.

By adjusting the MCS, it is possible to increase the data rate at the cost of a higher PEP. In the following, I introduce a theoretical method for evaluating the PEP. Essentially, there is a trade-off between the number of bits transmitted in one packet and the PEP. Given the SINR, γ, the achievable PEP in the finite blocklength regime can be approximated by [2, 15]

$$\epsilon \approx Q\left([C(\gamma) - R]\sqrt{\frac{N_{\mathrm{d}}}{V(\gamma)}}\right), \tag{2.2.5}$$

where $R = D/N_{\mathrm{d}}$ is the achievable data rate, $C(\gamma) = \log_2(1+\gamma)$ is the channel capacity, $V(\gamma) = (1-(1+\gamma)^{-2})\log_2^2 e$ is the channel dispersion, $Q(\cdot)$ is the Q-function, i.e., $Q(x) = \frac{1}{\sqrt{2\pi}}\int_x^\infty e^{-\frac{t^2}{2}}dt$.

From (2.2.4), I can derive the SINR with imperfect CSI. The first term is the useful signal, and the second and third terms are unknown Gaussian variables. Therefore, the received SINR with channel estimation errors can be derived as follows,

$$\gamma = \frac{\alpha p|\hat{\mathbf{h}}^H \mathbf{w_f}|^2}{\alpha p|\mathbf{e}^H \mathbf{w_f}|^2 + \sigma_{\mathbf{z}}^2}. \tag{2.2.6}$$

Then, I can obtain the corresponding PEP by substituting (2.2.6) into (2.2.5). In the rest of this work, I assume that the SINR in (2.2.6) can be estimated by the UE, but the interference power and the noise power are unknown.

2.3 Problem Formulation

In this section, I formulate the problem to maximise the resource utilisation efficiency, i.e., optimising the number of symbols for DM-RS and the packet sise given the time and frequency resources and the PEP constraint. I assume that a specific beamforming vector is selected in the beam training phase and focus on the resource allocation for DM-RS and data transmission.

2.3.1 Resource Constraint and CSI Observations

Given the time and frequency resources for a packet, the total number of symbols for channel estimation and data transmission is fixed. Let us denote the total number of symbols by $N_{\max}$. The constraint, i.e., $N_{\mathrm{c}} + N_{\mathrm{d}} = N_{\max}$, should be satisfied. If more symbols are allocated for DM-RS, the channel estimation will be more accurate. However, better channel estimation may not lead to higher resource utilisation efficiency since the number of symbols for data transmission decreases. Therefore, I aim to find the optimal resource allocation policy for DM-RS and data transmission. I assume that the transmission power is fixed for both DM-RS and data transmission.

I denote the CSI observation of a channel realisation $\mathbf{h}$ by $g_{\mathbf{h}}$. According to the CSI report from the UE, the BS can acknowledge the SINR. I consider the following three types of observations at the BS:

Perfect Channel Gain

To evaluate the upper bound of the resource utilisation efficiency, I assume that the BS has the perfect channel gain, i.e., $g_{\mathbf{h}} = |\mathbf{h}|^2$. Please note that the UE does not have the perfect CSI and still needs to execute the channel estimation through MMSE.

Estimated Channel Gain

Once the UE receives the DM-RS, it estimates the channel coefficient and sends a CSI report back to the BS. The BS can obtain the estimated channel gain from the CSI report. Due to the channel estimation errors, the estimated channel gain is different from the perfect channel gain. In this case, the CSI observation is given by $g_{\mathbf{h}} = |\hat{\mathbf{h}}|^2$, which is obtained from the estimated channel coefficient $\hat{\mathbf{h}}$.

Received Signal Strength

In a practical system, estimating received signal strength is much easier than estimating the channel coefficient $\hat{\mathbf{h}}$. The received signal strength obtained from CSI-RS or DM-RS can be used as the CSI observation, i.e., $g_{\mathbf{h}} = \|\mathbf{h}\mathbf{w_f}\mathbf{x_j} + \mathbf{z}\|^2$, where $\mathbf{x_j} \triangleq \{\mathbf{x_b}, \mathbf{x_c}\}$ is the CSI-RS or DM-RS.

2.3.2 Optimisation Problem Formulation

Resource allocation policy is a mapping from $g_{\mathbf{h}}$ to the number of symbols for channel estimation, i.e., length of DM-RS. I denote the policy by $N_{\mathrm{c}}(g_{\mathbf{h}})$. Resource utilisation efficiency is the number of bits that can be transmitted within given time and frequency resources. Therefore, I further optimise the packet size, which depends on the estimated channel gain, $g_{\mathbf{h}}$, and resource allocation policy, $N_{\mathrm{c}}(g_{\mathbf{h}})$. Hence, the number of bits in one packet is denoted by $D(g_{\mathbf{h}}, N_{\mathrm{c}}(g_{\mathbf{h}}))$. Although the 5G standard has specified the PEP requirement of URLLC, i.e., below 10^{-5}, it is not possible to achieve a deterministic PEP requirement with probability one in the presence of channel fading and channel estimation errors. To characterise the reliability of URLLC in practical wireless systems, I consider two types of PEP constraints: 1) an average

PEP constraint; 2) a PEP outage probability constraint.

Average PEP Constraint

Average PEP is defined as the average of ϵ in (2.2.5), where the average is taken over the CSI observation and the channel estimation error. Given the resource allocation policy, the packet size, and the CSI observation, the BS can obtain the SINR from the CSI report and estimate the PEP in (2.2.5). Thus, the optimisation problem can be formulated as

$$\max_{N_{\mathrm{c}}(g_{\mathbf{h}}), D(g_{\mathbf{h}}, N_{\mathrm{c}}(g_{\mathbf{h}}))} \quad \mathbb{E}_{g_{\mathbf{h}}}[D(g_{\mathbf{h}}, N_{\mathrm{c}}(g_{\mathbf{h}}))] \tag{2.3.1}$$

$$\text{s.t.} \quad \mathbb{E}_{g_{\mathbf{h}},\mathbf{e}}[\epsilon(N_{\mathrm{c}}(g_{\mathbf{h}}), D(g_{\mathbf{h}}, N_{\mathrm{c}}(g_{\mathbf{h}})), \gamma)] \leq \bar{\epsilon}, \tag{2.3.1a}$$

$$n_t \leq N_{\mathrm{c}}(g_{\mathbf{h}}) < N_{\max}, \tag{2.3.1b}$$

$$D(g_{\mathbf{h}}, N_{\mathrm{c}}(g_{\mathbf{h}})), N_{\mathrm{c}}(g_{\mathbf{h}}) \in \mathbb{Z}^{+}, \tag{2.3.1c}$$

where $\bar{\epsilon}$ is the maximum tolerable average PEP of a service, (2.3.1a) specifies the average PEP requirement, (2.3.1b) pertains to the number of DM-RS symbols to be greater than or equal to the number of transmitting antennas [15], and (2.3.1c) indicates that the number of DM-RS symbols and the data packet size are positive integers.

PEP Outage Probability Constraint

PEP outage probability is defined as the probability that the PEP is higher than a required threshold, ϵ_q. Given the resource allocation policy, the packet size, and the unknown channel estimation error, I define an indicator function, denoted by $\mathbb{1}\{\epsilon(N_{\mathrm{c}}(g_{\mathbf{h}}), D(g_{\mathbf{h}}, N_{\mathrm{c}}(g_{\mathbf{h}})), \gamma) > \epsilon_q\}$, to describe the PEP outage. If the PEP is higher

than ϵ_q, then $\mathbb{1}\{\epsilon(N_{\mathrm{c}}(g_{\mathbf{h}}), D(g_{\mathbf{h}}, N_{\mathrm{c}}(g_{\mathbf{h}})), \gamma) > \epsilon_q\} = 1$. Otherwise, $\mathbb{1}\{\epsilon(N_{\mathrm{c}}(g_{\mathbf{h}}), D(g_{\mathbf{h}}, N_{\mathrm{c}}(g_{\mathbf{h}})), \gamma) > \epsilon_q\} = 0$. Then, the PEP outage probability is the average of the indicator, where the average is taken over the CSI observation and the channel estimation error. It is worth noting that the expectation of the indicator is the probability that $\epsilon(N_{\mathrm{c}}(g_{\mathbf{h}}), D(g_{\mathbf{h}}, N_{\mathrm{c}}(g_{\mathbf{h}})), \gamma) > \epsilon_q$.

According to the above definition, the PEP outage probability can be obtained from

$$\Pr\{\epsilon(N_{\mathrm{c}}(g_{\mathbf{h}}), D(g_{\mathbf{h}}, N_{\mathrm{c}}(g_{\mathbf{h}})), \gamma) > \epsilon_q\} = \mathbb{E}_{g_{\mathbf{h}},\mathbf{e}}[\mathbb{1}\{\epsilon(N_{\mathrm{c}}(g_{\mathbf{h}}), D(g_{\mathbf{h}}, N_{\mathrm{c}}(g_{\mathbf{h}})), \gamma) > \epsilon_q\}].$$

The optimisation problem can be formulated as follows,

$$\max_{N_{\mathrm{c}}(g_{\mathbf{h}}), D(g_{\mathbf{h}}, N_{\mathrm{c}}(g_{\mathbf{h}}))} \quad \mathbb{E}_{g_{\mathbf{h}}}[D(g_{\mathbf{h}}, N_{\mathrm{c}}(g_{\mathbf{h}}))] \tag{2.3.2}$$

$$\text{s.t.} \quad \mathbb{E}_{g_{\mathbf{h}},\mathbf{e}}[\mathbb{1}\{\epsilon(N_{\mathrm{c}}(g_{\mathbf{h}}), D(g_{\mathbf{h}}, N_{\mathrm{c}}(g_{\mathbf{h}})), \gamma) > \epsilon_q\}] \leq \Upsilon \tag{2.3.2a}$$

$$n_t \leq N_{\mathrm{c}}(g_{\mathbf{h}}) < N_{\max}, \tag{2.3.2b}$$

$$D(g_{\mathbf{h}}, N_{\mathrm{c}}(g_{\mathbf{h}})), N_{\mathrm{c}}(g_{\mathbf{h}}) \in \mathbb{Z}^+, \tag{2.3.2c}$$

where (2.3.2a) is the constraint of PEP outage probability, and Υ is the maximum tolerable PEP outage probability of the URLLC service.

To solve the problem in (2.3.1) and (2.3.2), I need to find the optimal functions $D(g_{\mathbf{h}}, N_{\mathrm{c}}(g_{\mathbf{h}}))$ and $N_{\mathrm{c}}(g_{\mathbf{h}})$. These problems are functional optimisation and cannot be solved by using traditional optimisation algorithms.

2.3.3 Primal-dual Approach Formulation

I can first use primal-dual and Lagrangian multiplier method [28, 74] to solve the optimisation problems (2.3.1) and (2.3.2), which can be converted to a general formulation as

$$\max_{\lambda} \min_{N_{\mathrm{c}}(g_{\mathbf{h}}), D(g_{\mathbf{h}}, N_{\mathrm{c}}(g_{\mathbf{h}}))} \mathcal{L} \quad = -\mathbb{E}_{g_{\mathbf{h}}}[D(g_{\mathbf{h}}, N_{\mathrm{c}}(g_{\mathbf{h}}))] + \tag{2.3.3}$$

$$\mathbb{E}_{g_{\mathbf{h}},\mathbf{e}}(\lambda \Omega(N_{\mathrm{c}}(g_{\mathbf{h}}), D(g_{\mathbf{h}}, N_{\mathrm{c}}(g_{\mathbf{h}})), \gamma) - 1) \tag{2.3.4}$$

$$\text{s.t.} \quad \lambda \geq 0, \tag{2.3.4a}$$

$$n_t \leq N_{\mathrm{c}}(g_{\mathbf{h}}) < N_{\max}, \tag{2.3.4b}$$

$$D(g_{\mathbf{h}}, N_{\mathrm{c}}(g_{\mathbf{h}})), N_{\mathrm{c}}(g_{\mathbf{h}}) \in \mathbb{Z}^{+}, \tag{2.3.4c}$$

where $\mathcal{L}$ is the Lagrangian function, λ is the Lagrangian multiplier, $\Omega(N_{\mathrm{c}}(g_{\mathbf{h}}), D(g_{\mathbf{h}}), N_{\mathrm{c}}(g_{\mathbf{h}}), \gamma)$ is the normalised PEP constraint, i.e., $\Omega(N_{\mathrm{c}}(g_{\mathbf{h}}), D(g_{\mathbf{h}}, N_{\mathrm{c}}(g_{\mathbf{h}})), \gamma) = \frac{\epsilon(N_{\mathrm{c}}(g_{\mathbf{h}}), D(g_{\mathbf{h}}, N_{\mathrm{c}}(g_{\mathbf{h}})), \gamma)}{\bar{\epsilon}}$ for problem (2.3.1), and $\Omega(N_{\mathrm{c}}(g_{\mathbf{h}}), D(g_{\mathbf{h}}, N_{\mathrm{c}}(g_{\mathbf{h}})), \gamma) = \frac{\mathbb{1}\{\epsilon(N_{\mathrm{c}}(g_{\mathbf{h}}), D(g_{\mathbf{h}}, N_{\mathrm{c}}(g_{\mathbf{h}})), \gamma) > \epsilon_q\}}{\Upsilon}$ for problem (2.3.2).

2.4 Unsupervised Learning Approaches

In general cases, there are no closed-form solutions for problems expressed in (2.3.1) and (2.3.2). Since I only have the estimated channel but do not have the true CSI, channel estimation errors are not available for evaluating the average PEP or PEP outage probability. Moreover, the Q-function in (2.2.5) is an integration, making it impossible to obtain closed-form solutions for (2.3.1) and (2.3.2). To overcome this

difficulty, I use DNNs to approximate the functions to be optimised and use unsupervised learning in [28] to optimise the parameters of the DNNs. I design a cascaded DNN structure and apply either model-based or model-free training techniques to optimise the parameters of the DNNs. The model-based training is applicable when the SINR in (2.2.6) is available. The model-free training can be applied in more practical scenarios where theoretical models are not available. The underlying idea is to use the primal-dual method in (2.3.3) as the loss function and use stochastic gradient descent (SGD) to update the parameters of the cascaded DNN and the Lagrangian multiplier. After the training stage, the DNNs can be used for decision-making in wireless systems with low inference complexity [23].

2.4.1 Model-Based Unsupervised Learning

From the received SINR, it is possible to evaluate the PEP according to the model in (2.2.5). With the help of the theoretical model, I can use the model-based unsupervised learning method.

Cascaded DNN Design

Given the total number of symbols, if I allocate more symbols for channel estimation, I can obtain more accurate CSI, but fewer symbols are available for data transmission. The packet size depends on the CSI and the number of symbols for data transmission. Thus, the policy first determines the number of symbols for channel estimation and then determines the packet size based on the estimated CSI and the available symbols for data transmission. In addition, my preliminary results in [23] show that if I

represent a policy by a fully connected DNN without exploiting the relationship of different optimisation variables, it is difficult to obtain a policy with good performance. With this domain knowledge, I design a cascaded DNN architecture accordingly. The input of the first DNN is the channel observation $g_{\mathbf{h}}$, and the output is the number of symbols allocated for DM-RS transmissions. The first DNN with parameters, θ_N, is denoted as $\phi_N(g_{\mathbf{h}}|\theta_N)$. The number of symbols allocated for DM-RS can be obtained from the output of the $\phi_N(g_{\mathbf{h}}|\theta_N)$. Then, I can then obtain the number of symbols for data transmission, i.e., $N_{\mathrm{d}} = N_{\max} - N_{\mathrm{c}}$. The second DNN, denoted by $\phi_D(g_{\mathbf{h}}, \phi_N(g_{\mathbf{h}}|\theta_N)|\theta_D)$, takes N_{c} and $g_{\mathbf{h}}$ as its input and outputs the packet size, where θ_D is the parameters. In order to train the cascaded DNN, the Lagrangian function in (2.3.3) is used as the loss function. Lagrangian multipliers and the parameters of the cascaded DNN are updated by using the primal-dual method [28].

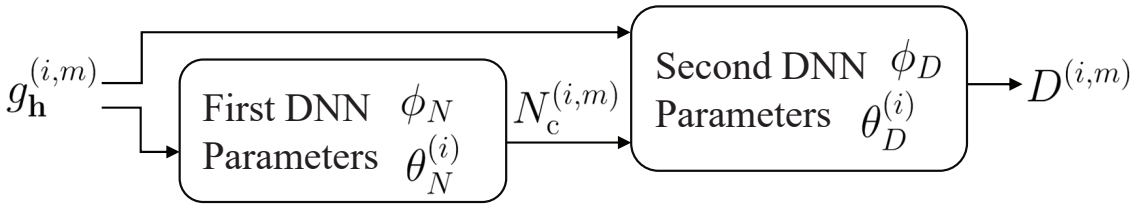


Figure 2.2: Model-based cascaded DNN structure, where $g_{\mathbf{h}}^{(i,m)}$ is the m-th CSI observation in the i-th iteration, $N_{\mathrm{c}}^{(i,m)}$ and $D^{(i,m)}$ are the outputs of the two DNNs.

Model-Based Training Procedure

In the i-th iteration of the primal-dual algorithm, I generate M channel realisations according to a stochastic channel model. The m-th sample in the i-th iteration is denoted by $\mathbf{h}^{(i,m)}$, where $m \in \{1, 2, \cdots, M\}$. The CSI observation is given by $g_{\mathbf{h}}^{(i,m)}$. The cascaded DNN is shown in Fig. 2.2, where $N_{\mathrm{c}}^{(i,m)}$ and $D^{(i,m)}$ are

obtained from the cascaded DNN. Since the statistical distribution of channel estimation errors is derived from (2.2.3), I can also generate M channel estimation errors in my simulation, $\mathbf{e}^{(i,m)}$. I assume $\mathbf{h}^{(i,m)}$ and $\mathbf{e}^{(i,m)}$ are unknown to the BS and the UE, but the received SINR in (2.2.6), $\gamma^{(i,m)}$, is available. Based on the outputs of the cascaded DNN and the received SINR, the BS can evaluate the PEP, $\epsilon(\phi_N(g_{\mathbf{h}}^{(i,m)}|\theta_N), \phi_D(g_{\mathbf{h}}^{(i,m)}, \phi_N(g_{\mathbf{h}}^{(i,m)}|\theta_N)|\theta_D), \gamma^{(i,m)})$. The normalised average PEP estimated by the m-th sample in the i-th iteration is given by

$$\Omega^{(i,m)} = \frac{\epsilon(\phi_N(g_{\mathbf{h}}^{(i,m)}|\theta_N), \phi_D(g_{\mathbf{h}}^{(i,m)}, \phi_N(g_{\mathbf{h}}^{(i,m)}|\theta_N)|\theta_D), \gamma^{(i,m)})}{\bar{\epsilon}}. \tag{2.4.1}$$

The normalised PEP outage probability is given by

$$\Omega^{(i,m)} = \frac{\mathbb{1}\{\epsilon(\phi_N(g_{\mathbf{h}}^{(i,m)}|\theta_N), \phi_D(g_{\mathbf{h}}^{(i,m)}, \phi_N(g_{\mathbf{h}}^{(i,m)}|\theta_N)|\theta_D), \gamma^{(i,m)}) > \epsilon_q\}}{\Upsilon}. \tag{2.4.2}$$

The loss function in the i-th iteration can be estimated by the M samples according to

$$\hat{\mathcal{L}}^{(i)} = \frac{1}{M}\sum_{m=1}^{M}[-\phi_D(g_{\mathbf{h}}^{(i,m)}, \phi_N(g_{\mathbf{h}}^{(i,m)}|\theta_N^{(i)})|\theta_D^{(i)}) + \lambda^{(i)}(\Omega^{(i,m)} - 1)]. \tag{2.4.3}$$

By using the stochastic gradient ascent algorithm, I update $\lambda^{(i)}$ according to

$$\lambda^{(i+1)} = \left[\lambda^{(i)} + \eta_\lambda^{(i)}\frac{\partial\hat{\mathcal{L}}^{(i)}}{\partial\lambda^{(i)}}\right]^+ = \left[\lambda^{(i)} + \eta_\lambda^{(i)}\frac{1}{M}\sum_{m=1}^{M}(\Omega^{(i,m)} - 1)\right]^+, \tag{2.4.4}$$

where $[x]^+ \triangleq \max\{x, 0\}$ and $\eta_\lambda^{(i)}$ is the learning rate of the dual variable. In the primal domain, I apply the SGD algorithm to train the parameters of the cascaded DNN. The gradient of $\theta^{(i)} \triangleq \left\{\theta_N^{(i)}, \theta_D^{(i)}\right\}$ is denoted by $\nabla_\theta\hat{\mathcal{L}}^{(i)}$. To use SGD for training DNNs, the loss function needs to be differentiable. Therefore, I relax $N_{\mathrm{c}}^{(i,m)}$ and $D^{(i,m)}$ as continuous variables. As $N_{\mathrm{c}}^{(i,m)}$ and $D^{(i,m)}$ can be up to a few hundred, the relaxation has minimal impact on both the PEP and the resource utilisation efficiency. The

parameters are updated by

$$\theta^{(i+1)} = \theta^{(i)} - \eta^{(i)} \nabla_{\theta} \hat{\mathcal{L}}^{(i)}, \tag{2.4.5}$$

where $\eta^{(i)} \triangleq \left\{ \eta_N^{(i)}, \eta_D^{(i)} \right\}$, $\eta_N^{(i)}$ and $\eta_D^{(i)}$ are the learning rate of ϕ_N and ϕ_D, respectively. The model-based unsupervised learning algorithm is summarised in Algorithm 1.

Algorithm 1 Model-Based Unsupervised Learning

Require: Initial parameter $\theta^{(0)}$, $\lambda^{(0)}$

1: **for** $i = 0, 1, 2, \cdots$ **do**
2: Generate M random samples of $\mathbf{h}^{(i,m)}, m \triangleq \{1, 2, \cdots, M\}$
3: Obtain channel observations $g_{\mathbf{h}}^{(i,m)}$
4: Obtain $N_{\mathrm{c}}^{(i,m)}$ through $\phi_N(g_{\mathbf{h}}^{(i,m)} | \theta_N^{(i)})$
5: Generate $\mathbf{e}^{(i,m)}$ according to the circularly symmetric complex Gaussian distribution with variance in (2.2.3)
6: Obtain $D^{(i,m)}$ through $\phi_D(g_{\mathbf{h}}^{(i,m)}, \phi_N(g_{\mathbf{h}}^{(i,m)} | \theta_N^{(i)}) | \theta_D^{(i)})$
7: Evaluate $\hat{\mathcal{L}}^{(i)}$
8: Evaluate gradients $\nabla_{\theta_N} \hat{\mathcal{L}}^{(i)}$ and $\nabla_{\theta_D} \hat{\mathcal{L}}^{(i)}$
9: Update parameters $\lambda^{(i)}$, $\theta_N^{(i)}$, and $\theta_D^{(i)}$ according to (2.4.4) and (2.4.5).

2.4.2 Model-Free Unsupervised Learning

The model-based solution relies on the theoretical model in (2.2.5). However, with practical MCS, there is no closed-form expression of PEP. Therefore, I cannot derive the expression of constraints (2.3.1a) and (2.3.2a). To train the cascaded DNN, I propose a model-free unsupervised learning algorithm.

Cascaded DNN Design with Stochastic Policies

Since there is no closed-form expression of PEP, I cannot obtain the gradient in (2.4.5). To address this issue, I use the *policy gradient estimation* method from [28] to approximate the gradient in the model-free learning method. To apply the *policy gradient estimation*, I replace the deterministic policies in the cascaded DNN structure with two stochastic policies [75], where the number of symbols for channel estimation and the packet size obtained from the stochastic policies are denoted by $\hat{N}_{\mathrm{c}}$ and $\hat{D}$, respectively. Since $\hat{N}_{\mathrm{c}}$ depends on the channel observation $g_{\mathbf{h}}$ and trainable parameters of the first DNN θ_N, I denote the distribution function of $\hat{N}_{\mathrm{c}}$ by $\pi_{\hat{N}_{\mathrm{c}}}(z_1|g_{\mathbf{h}},\theta_N)$, where z_1 is a random variable following the distribution $\pi_{\hat{N}_{\mathrm{c}}}(\cdot|g_{\mathbf{h}},\theta_N)$. The packet size $\hat{D}$ relies on the trainable parameters of the second DNN θ_D, and the input of the second DNN, $\hat{N}_{\mathrm{c}}$, and the CSI observation $g_{\mathbf{h}}$. Therefore, the distribution function of $\hat{D}$ is denoted by $\pi_{\hat{D}}(z_2|g_{\mathbf{h}},\hat{N}_{\mathrm{c}},\theta_D)$, where z_2 is a random variable following the distribution $\pi_{\hat{D}}(\cdot|g_{\mathbf{h}},\hat{N}_{\mathrm{c}},\theta_D)$. When the density functions approach the impulse functions, the stochastic policies become deterministic policies, i.e., $\pi_{\hat{N}_{\mathrm{c}}}(z_1|g_{\mathbf{h}},\theta_N)=\delta(z_1-\hat{N}_{\mathrm{c}})$ and $\pi_{\hat{D}}(z_2|g_{\mathbf{h}},\hat{N}_{\mathrm{c}},\theta_D)=\delta(z_2-\hat{D})$, respectively. However, the impulse function is non-differentiable, and I utilise a differentiable distribution function, i.e., truncated Gaussian distribution, to approximate the impulse function [28]. Thus, the distributions $\pi_{\hat{N}_{\mathrm{c}}}(z_1|g_{\mathbf{h}},\theta_N)$ and $\pi_{\hat{D}}(z_2|g_{\mathbf{h}},z_1,\theta_D)$ can be represented by $\mathcal{N}(\mu_N,\beta_N)$, and $\mathcal{N}(\mu_D,\beta_D)$, respectively.

I design another cascaded DNN structure to represent the stochastic policies in the model-free unsupervised learning algorithm. The first DNN, denoted by $\omega_N(g_{\mathbf{h}}|\theta_N)$, represents the mapping from the CSI observation to $\{\mu_N,\beta_N\}$. From the expectation and variance of the truncated Gaussian distribution, I can generate one realisation of

z_1 by using the reparameterisation trick [76], $\hat{N}_{\rm c} = \mu_N + \sqrt{\beta_N}\xi$, where $\xi \sim \mathcal{N}(0,1)$. The second DNN is denoted by $\omega_D(g_{\mathbf{h}}, \hat{N}_{\rm c}|\theta_D)$. It takes the realisation of the first stochastic policy and the CSI observation as its input and outputs the mean and variance of the packet size, i.e., $\{\mu_D, \beta_D\}$. By applying the reparameterisation trick, I obtain a realisation of z_2, $\hat{D} = \mu_D + \sqrt{\beta_D}\xi$.

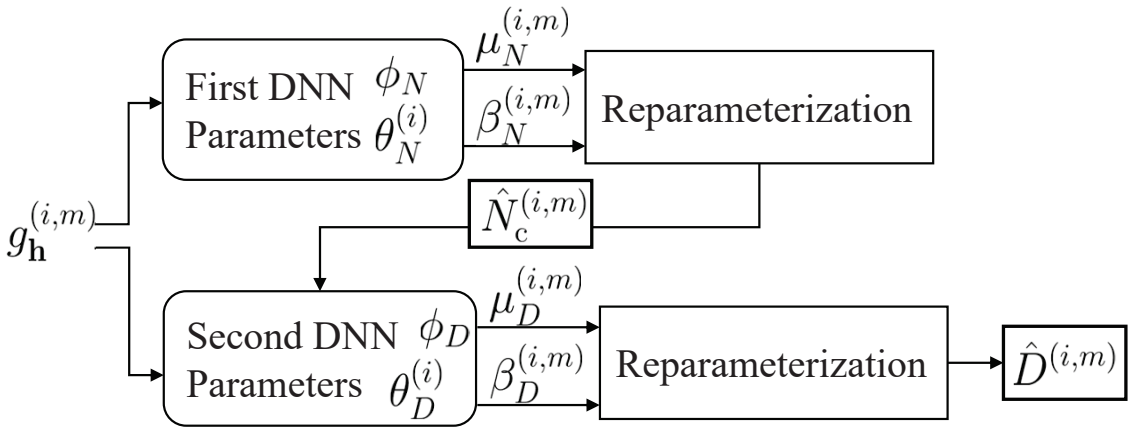


Figure 2.3: Model-free Cascaded DNN structure, where $g_{\mathbf{h}}^{(i,m)}$ is the m-th CSI observation in the i-th iteration, $\mu_N^{(i,m)}$ and $\beta_N^{(i,m)}$ are the outputs of the first DNN, $\mu_D^{(i,m)}$ and $\beta_D^{(i,m)}$ are the outputs of the second DNN, $\hat{N}_c^{(i,m)}$ and $\hat{D}^{(i,m)}$ are random samples captured from $\mathcal{N}(\mu_N^{(i,m)}, \beta_N^{(i,m)})$, and $\mathcal{N}(\mu_D^{(i,m)}, \beta_D^{(i,m)})$, respectively.

Model-Free Training Procedure

In the i-th iteration, I generate M channel realisations, $\mathbf{h}^{(i,m)}$, where $m \in \{1, 2, \cdots, M\}$. For each channel realisation, the CSI observation at the BS is $g_{\mathbf{h}}^{(i,m)}$. Based on $g_{\mathbf{h}}^{(i,m)}$, I can obtain $\{\mu_N^{(i,m)}, \beta_N^{(i,m)}\}$ and $\{\mu_D^{(i,m)}, \beta_D^{(i,m)}\}$ from the cascaded DNN, as shown in Fig. 2.3. Then, I generate one realisation of $z_1^{(i,m)}$ from

$\mathcal{N}(\mu_N^{(i,m)}, \beta_N^{(i,m)})$ and one realisation of $z_2^{(i,m)}$ from $\mathcal{N}(\mu_D^{(i,m)}, \beta_D^{(i,m)})$ by the reparameterisation trick, denoted by $\hat{N}_{\mathrm{c}}^{(i,m)}$ and $\hat{D}^{(i,m)}$, respectively. I then generate M channel estimation errors $\mathbf{e}^{(i,m)}$ according to its distribution in (2.2.3). Although $\mathbf{e}^{(i,m)}$ is unknown to the BS and the UE, I assume that the BS can obtain the receive SINR $\gamma^{(i,m)}$ from the CSI report. The PEP achieved by this sample is given by $\hat{\epsilon}(\hat{N}_{\mathrm{c}}^{(i,m)}, \hat{D}^{(i,m)}, \gamma^{(i,m)})$. The normalised PEP constraint estimated by the samples is denoted by $\hat{\Omega}^{(i,m)}$. The normalised PEP can be estimated from

$$\hat{\Omega}^{(i,m)} = \frac{\epsilon(\hat{N}_{\mathrm{c}}^{(i,m)}, \hat{D}^{(i,m)}, \gamma^{(i,m)})}{\bar{\epsilon}}. \tag{2.4.6}$$

The normalised PEP outage is given by

$$\hat{\Omega}^{(i,m)} = \frac{\mathbb{1}\{\epsilon(\hat{N}_{\mathrm{c}}^{(i,m)}, \hat{D}^{(i,m)}, \gamma^{(i,m)}) > \epsilon_q\}}{\Upsilon}. \tag{2.4.7}$$

In the i-th iteration, the estimated loss function is given by

$$\hat{\mathcal{L}}^{(i)} = \frac{1}{M}\sum_{m=1}^{M}[-\hat{D}^{(i,m)} + \lambda^{(i)}(\hat{\Omega}^{(i,m)} - 1)]. \tag{2.4.8}$$

The Lagrangian multiplier $\lambda^{(i)}$ is updated according to

$$\lambda^{(i+1)} = [\lambda^{(i)} + \eta_\lambda^{(i)}\frac{\partial\hat{\mathcal{L}}^{(i)}}{\partial\lambda^{(i)}}]^+ = \left[\lambda^{(i)} + \eta_\lambda^{(i)}\frac{1}{M}\sum_{m=1}^{M}(\hat{\Omega}^{(i,m)} - 1)\right]^+. \tag{2.4.9}$$

Similar to the model-based training, I relax $\hat{N}_{\mathrm{c}}^{(i,m)}$ and $\hat{D}^{(i,m)}$ as continuous variables in model-free training. The parameters of the cascaded DNN can be updated according to

$$\theta^{(i+1)} = \theta^{(i)} - \eta^{(i)}\nabla_\theta\hat{\mathcal{L}}^{(i)}, \tag{2.4.10}$$

where $\theta^{(i)} \triangleq \left\{\theta_N^{(i)}, \theta_D^{(i)}\right\}$, $\eta^{(i)} \triangleq \left\{\eta_N^{(i)}, \eta_D^{(i)}\right\}$, and the gradient $\nabla_\theta \hat{\mathcal{L}}^{(i)}$ can be estimated by [28, 75]

$$\nabla_{\theta_N} \hat{\mathcal{L}}^{(i)} = \frac{1}{M} \sum_{m=1}^{M} \left\{ \left[-\hat{D}^{(i,m)} + \lambda^{(i)} (\hat{\Omega}^{(i,m)} - 1) \right] \nabla_{\theta_N} \left[\log(\pi_{\hat{N}_\mathrm{c}}(\hat{N}_\mathrm{c}^{(i,m)} | g_{\mathbf{h}}^{(i,m)}, \theta_N^{(i)})) \right] \right\}, \tag{2.4.11}$$

$$\nabla_{\theta_D} \hat{\mathcal{L}}^{(i)} = \frac{1}{M} \sum_{m=1}^{M} \left\{ \left[-\hat{D}^{(i,m)} + \lambda^{(i)} (\hat{\Omega}^{(i,m)} - 1) \right] \nabla_{\theta_D} \left[\log(\pi_{\hat{D}}(\hat{D}^{(i,m)} | g_{\mathbf{h}}^{(i,m)}, \hat{N}_\mathrm{c}^{(i,m)}, \theta_D^{(i)})) \right] \right\}. \tag{2.4.12}$$

The model-free unsupervised learning algorithm is summarised in Algorithm 2.

Algorithm 2 Model-Free Unsupervised Learning

Require: initial parameter $\theta^{(0)}$, $\lambda^{(0)}$

1: **for** $i = 0, 1, 2, \cdots$ in iterations **do**
2: Generate M random samples of $\mathbf{h}^{(i,m)}, m \triangleq \{1, 2, \cdots, M\}$
3: Obtain channel observations $g_{\mathbf{h}}^{(i,m)}$
4: Obtain $\mu_N^{(i,m)}$, and $\beta_N^{(i,m)}$ through $\phi_N(g_{\mathbf{h}}^{(i,m)} | \theta_N^{(i)})$
5: Obtain $\hat{N}_c^{(i,m)}$ from the reparameterisation trick.
6: Generate samples of $\mathbf{e}^{(i,m)}$ according to its distribution function with variance of (2.2.3)
7: Obtain $\mu_D^{(i,m)}$, and $\beta_D^{(i,m)}$ through $\phi_D(g_{\mathbf{h}}^{(i,m)}, \phi_N(g_{\mathbf{h}}^{(i,m)} | \theta_N^{(i)}) | \theta_D^{(i)})$
8: Obtain $\hat{D}^{(i,m)}$ from the reparameterisation trick.
9: Compute $\hat{\mathcal{L}}^{(i)}$
10: Estimate the gradients $\nabla_{\theta_N} \hat{\mathcal{L}}^{(i)}$ and $\nabla_{\theta_D} \hat{\mathcal{L}}^{(i)}$
11: Update parameters $\lambda^{(i)}$, $\theta_N^{(i)}$, and $\theta_D^{(i)}$ for backward-propagation

2.4.3 Reliability Evaluation with PEP Outage Probability Constraint

For the services with the PEP outage probability constraint, the model-based and model-free unsupervised learning methods can hardly meet the reliability requirements. This is because these methods use SGD to update the parameters, where the loss function needs to be differentiable. For the average PEP constraint, the PEP is a continuous and differentiable function. However, the PEP outage probability is the expectation of the indicator function, which is not continuous. When estimating the gradient with a batch of samples, the gradient is unbounded when the indicator switches between zero and one. To improve the stability of SGD, I estimate the normalised PEP of the policies N_{c} and D by another DNN. As shown in Fig. 2.4, I take the model-based unsupervised learning algorithm as an example to illustrate reliability evaluation.

In the i-th iteration, I generate M channel realisations and channel estimation errors, i.e., $\mathbf{h}^{(i,m)}$ and $\mathbf{e}^{(i,m)}$, where $m \in \{1, 2, \cdots, M\}$. From the cascaded DNN, I can obtain the number of symbols allocated for DM-RS, $N_{\mathrm{c}}^{(i,m)}$, and the packet size, $D^{(i,m)}$. Then, the value of the indicator $\mathbb{1}\{\epsilon(N_{\mathrm{c}}^{(i,m)}, D^{(i,m)}, \gamma^{(i,m)}) > \epsilon_q\}$ can be obtained from the theoretical model in (2.2.5). I denote the value of the indicator by $\upsilon^{(i,m)}$, which is either zero or one. Thus, the indicator function is not differentiable. To obtain a differentiable reliability evaluation function, I use a DNN to approximate the mapping from each sample $\{g_{\mathbf{h}}^{(i,m)}, N_{\mathrm{c}}^{(i,m)}, D^{(i,m)}\}$ to $\upsilon^{(i,m)}$. The activation function in the output layer of the DNN is the sigmoid function ranging from zero to one, and thus the DNN is continuous and differentiable.

The DNN is denoted by ϕ_r, and the parameters of the DNN in the i-th iteration

are denoted by $\theta_r^{(i)}$. To train the DNN, I use $\{g_{\mathbf{h}}^{(i,m)}, N_{\mathrm{c}}^{(i,m)}, D^{(i,m)}\}$ and $\upsilon^{(i,m)}$ as the training samples and labels. Since the label is either zero or one, the cross entropy loss is applied, i.e.,

$$\hat{\mathcal{L}}_r^{(i)} = -\frac{1}{M}\sum_{m=1}^{M}\{\upsilon^{(i,m)} \ln \hat{\upsilon}^{(i,m)} + (1-\upsilon^{(i,m)})(\ln(1-\hat{\upsilon}^{(i,m)}))\}, \tag{2.4.13}$$

where $\hat{\upsilon}^{(i,m)}$ is the output of the DNN. It is worth noting that the reliability evaluation function can work with both model-based and model-free algorithms. In the model-free algorithm, the indicator of PEP outage is evaluated from a batch of samples $\{g_{\mathbf{h}}^{(i,m)}, \hat{N}_{\mathrm{c}}^{(i,m)}, \hat{D}^{(i,m)}\}$ in practical systems. Therefore, I can also use the indicator as the label to train the DNN for reliability evaluation. The algorithm is shown in Algorithm 3.

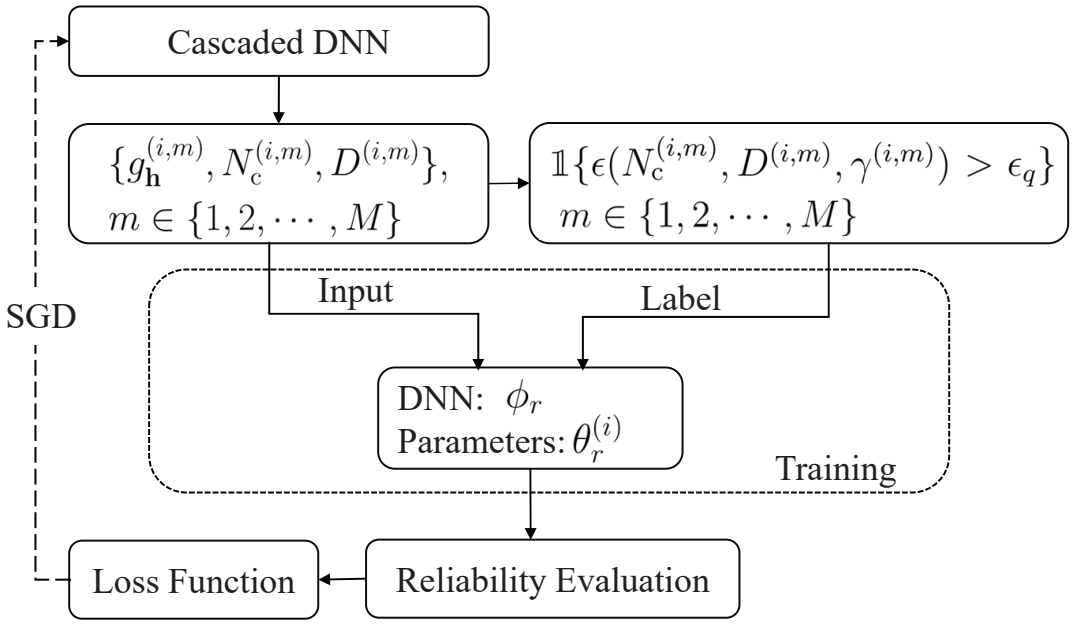


Figure 2.4: Reliability evaluation function for the m-th channel sample in the i-th iteration.

Algorithm 3 Reliability Evaluation Function

Require: initial parameter $\theta^{(0)}$, $\lambda^{(0)}$

1: **for** $i = 0, 1, 2, \cdots$ in iterations **do**

2: Generate M random samples of $\mathbf{h}^{(i,m)}$ and $\mathbf{e}^{(i,m)}$, $m \triangleq \{1, 2, \cdots, M\}$

3: Obtain channel observations $g_{\mathbf{h}}^{(i,m)}$

4: Obtain $N_{\mathrm{c}}^{(i,m)}$ and $D^{(i,m)}$ from the cascaded DNN

5: Compute the PEP outage probability based on $\{g_{\mathbf{h}}^{(i,m)}, N_{\mathrm{c}}^{(i,m)}, D^{(i,m)}\}$

6: Train a DNN to evaluate the PEP outage probability

7: Compute the loss function

8: Continue the back-propagation of unsupervised learning algorithms

2.4.4 Deep Transfer Learning for Dynamic Radio Resources

The unsupervised learning algorithms discussed above are trained offline with a fixed number of antennas, n_t, a given amount of time and frequency resources, $N_{\max}$, and a certain channel distribution. In practice, the BS may switch on/off some antennas and adjust the time and frequency resource allocation for each user. A mobile user may experience different channel distributions. To apply the DNN in dynamic wireless communication systems, I need to train the parameters of the DNN in different scenarios. To improve sample efficiency, I adopt deep transfer learning to fine-tune the pre-trained DNN [23]. Specifically, I first train the cascaded DNN using unsupervised learning algorithms with n_t antennas and a fixed value of $N_{\max}$ under Rayleigh fading channels. The pre-trained parameters are denoted by θ_{off}. When n_t, $N_{\max}$, or the channel distribution changes, I initialize the cascaded DNN with θ_{off} and fine-tune it with a few new samples. The performance of transfer learning will be provided in Section 2.5.7, with specific examples.

2.4.5 Complexity of Cascaded DNN

After the offline training stage, I can apply the trained cascaded DNN structure for resource allocation in practical URLLC systems. The computational complexity of the cascaded DNN is primarily due to the forward propagation algorithm, which comprises mathematical operations such as addition, multiplication, and activation functions. It is worth noting that multiplication operations require the most computation resources among these operations. I denote the number of multiplication operations of the two neural networks, ϕ_N and ϕ_D, in the cascaded neural network architecture by $\mathrm{N}(\phi_N, \phi_D)$, which is given by [23]

$$\mathrm{N}(\phi_N, \phi_D) = \sum_{l_{\phi_N}=0}^{L_{\phi_N}-1} \kappa_{\phi_N}^{[l_{\phi_N}]} \times \kappa_{\phi_N}^{[l_{\phi_N}+1]} + \sum_{l_{\phi_D}=0}^{L_{\phi_D}-1} \kappa_{\phi_D}^{[l_{\phi_D}]} \times \kappa_{\phi_D}^{[l_{\phi_D}+1]}, \tag{2.4.14}$$

where l_{ϕ_N} and l_{ϕ_D} are the indices of hidden layers of ϕ_N and ϕ_D, respectively, L_{ϕ_N} and L_{ϕ_D} are the number of layers of ϕ_N and ϕ_D, respectively, $\kappa_{\phi_N}^{[l_{\phi_N}]}$ and $\kappa_{\phi_D}^{[l_{\phi_D}]}$ are the number of neurons in the l_{ϕ_N}-th layer of ϕ_N and the l_{ϕ_D}-th layer of ϕ_D, respectively. Given that numbers of other operations, such as addition and activation functions, are considerably smaller than $\mathrm{N}(\phi_N, \phi_D)$, the computational complexity of the cascaded DNN is $\mathcal{O}(\mathrm{N}(\phi_N, \phi_D))$. As demonstrated in [23], the complexity is low enough to be implemented in practical systems in real time.

2.5 Simulation Results

In this section, I validate the reliability and resource utilisation efficiency of the policies obtained from the unsupervised learning algorithms.

2.5.1 System Setup

I consider a downlink MISO system, where the BS equipped with four antennas serves a single-antenna UE. The time and frequency resources allocated for one packet are $N_{\max} = T_f B$, where B is the available bandwidth, and T_f is the duration of one frame. The variance of noise power is given by $\sigma_{\mathbf{z}}^2 = N_0 B$, where N_0 is the noise spectral density. The large-scale channel gain is obtained from the path loss model, i.e., $\alpha = -35.3 - 37.6 \log_{10}(d)$, where d (m) is the distance between the BS and the UE. The parameters are summarised in Table 2.1.

Table 2.1: System parameters for simulation setup.

Parameters	Notations	Values
BS antenna	n_t	4
Frame duration	T_f	1 ms
Bandwidth	B	1 MHz
Distance	d	300 m
Transmission power	p	23 dBm
Noise spectral density	N_0	173 dBm/Hz
Average PEP requirement	$\bar{\epsilon}$	10^{-5}
Required PEP threshold	ϵ_q	10^{-5}
PEP outage probability requirement	Υ	10^{-4}

2.5.2 Benchmark

I compare my policies with a benchmark method modified from an existing resource allocation policy in [19], where MRT is applied. Given an energy budget constraint, the authors of [19] proved that the optimal number of symbols for channel estimation is equal to the number of antennas n_t, where the average PEP is obtained from the average SINR. In the benchmark method, I set N_{c} to n_t and maximise the resource utilisation efficiency by optimising the packet size D. In order to find the optimal packet size D, the unsupervised learning algorithm is applied to train the second DNN in the cascaded DNN, where the output of the first DNN is fixed as $N_{\mathrm{c}} = n_t$. Different from [19], the energy budget constraint is replaced by a maximum power constraint in my work. In order to maximise the number of bits that can be transmitted in one packet, the maximum transmission power is used in both channel estimation and data transmission. Thus, there is no need to optimise the transmit power in my work.

2.5.3 Codebook-Based Precoding

I use the 'Type I Single-Panel' codebook in my simulation [21]. The key parameters for developing the codebook include (N_1, N_2) and (O_1, O_2). N_1 and N_2 denote the numbers of horizontal and vertical antenna ports, respectively. O_1 and O_2 are the oversampling factors in the horizontal and vertical directions, respectively. The possible configurations of (N_1, N_2) and (O_1, O_2) are specified in [21]. The layout of the antenna array in my simulation is $(N_1, N_2) = (4, 1)$, and the oversampling factor is set to $(O_1, O_2) = (4, 1)$. Based on (N_1, N_2) and (O_1, O_2), I can generate the codebook according to the table of 'CodebookMode=2' for 1-layer CSI reporting. Specifically,

Table 2.2: Hyper-parameters for the DNN structures

Hyper-parameters	Model-Based cascaded DNN		Model-Free cascaded DNN		Embedded DNN
	1st DNN	2nd DNN	1st DNN	2nd DNN	
Learning rate	0.0005	0.00005	0.0005	0.00005	0.0005
Number of hidden layers	4	3	4	3	3
Number of neurons in different hidden layers	64/4/4/2	16/16/8	64/4/4/2	16/16/8	32/32/4
Number of neurons in the output layer	1	1	2	2	1
Batch size	500000				2048
Iterations	10000				
Activation function in hidden layers	Leaky ReLu (slope coefficient: 0.1)				ReLu
Activation function in the output layer	tanh	ReLu	tanh	ReLu	sigmoid

the precoding matrices $\mathbf{W}$ in the codebook are obtained from

$$\mathbf{W} = \mathbf{W}_1\mathbf{W}_2, \mathbf{W}_1 = \begin{bmatrix} \mathbf{b} & 0 \\ 0 & \mathbf{b} \end{bmatrix}, \mathbf{W}_2 = \frac{1}{\sqrt{2N_1N_2}} \begin{bmatrix} 1 \\ \varphi_n \end{bmatrix}, \tag{2.5.1}$$

where $\mathbf{b} = [1, e^{j2\pi l/N_1O_1}, \cdots, e^{j2\pi l(N_1-1)/N_1O_1}], l \triangleq \{0, 1, \cdots, (N_1O_1 - 1)\}$ is a single beam of antenna elements, and $\varphi_n = e^{j\pi n/2}, n \triangleq \{0, 1, 2, 3\}$ is the co-phasing coefficient [21].

2.5.4 Hyper-Parameters of Neural Networks

I use fully connected neural networks in unsupervised learning algorithms. The hyper-parameters and design of all DNNs are presented in Table 2.2 unless specified otherwise. For example, the first DNN in the model-based cascaded DNN has four hidden layers, comprising 64, 4, 4, and 2 neurons in the first, second, third, and fourth layers, respectively. Therefore, the number of neurons in different hidden layers is represented by "64/4/4/2". In addition, the channel observation is normalised by the number of antennas, i.e., $g_{\mathbf{h}}/n_t$, while the number of symbols for channel estimation is normalised by the total amount of time and frequency resources, i.e., $N_{\mathrm{c}}/N_{\max}$. I define the resource utilisation efficiency as $D/N_{\max}$. The initial weights in the cascaded DNN are initialised with Gaussian random variables with zero mean and unit variance, and the initial bias is fixed at 0.1. The initial value of the Lagrangian multiplier is set to 0. I utilise the hyperbolic tangent (tanh) function as the activation function for the DNN used to obtain N_{c}. I guarantee the range of N_{c} by mapping the outputs of tanh to the values between n_t and $N_{\max}$. Then, I use the floor function to round the obtained value of N_{c} to the nearest integer.

2.5.5 Performance Evaluation for Average PEP Requirement

I initially compare the performance of model-based and model-free unsupervised learning algorithms in terms of the average PEP requirement. Fig. 2.5 depicts the optimal resource utilisation efficiency for both MRT and codebook-based precoding techniques. The results indicate that the model-free algorithm achieves nearly the same resource utilisation efficiency as the model-based algorithm, with both algorithms utilizing the same precoding technique. However, the convergence time of the model-free algorithm is slightly longer than that of the model-based algorithm. Thus, the model-free unsupervised learning algorithm can achieve a nearly optimal policy in practical systems without any theoretical models, but it comes at the cost of a slightly longer convergence time. Additionally, the resource utilisation efficiency of codebook-based precoding is found to have a performance loss of 30% compared to MRT. It is worth noting that the comparison does not take the overhead of the channel report into account. Further study is needed to enhance the resource utilisation efficiency of codebook-based precoding with limited channel feedback. Fig. 2.6 shows that my proposed algorithms can ensure the required constraint after a small number of training iterations. The reason why the curves fluctuate around the required PEP is that I only have a limited number of samples (i.e., a batch size of 500000) to estimate the average PEP. Finally, Fig. 2.7 presents the optimal resource allocation policy. As the codebook-based precoding results in lower SINR values when compared to MRT, more symbols are allocated to DM-RS when codebook-based precoding is used.

In Table 2.3, I test the resource utilisation efficiency and the average PEP, where 5×10^8 channel realisations and channel estimation errors are randomly generated for performance evaluation. The results show that the average PEP requirement can

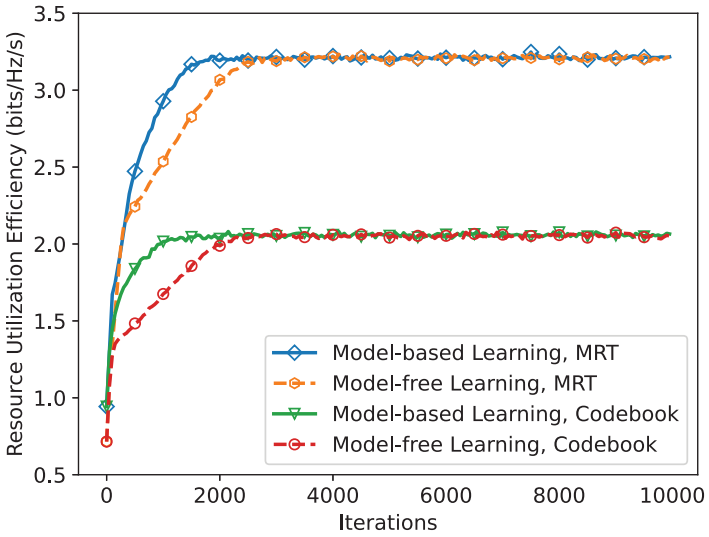


Figure 2.5: Resource utilisation efficiency in the training stage, where the average PEP requirement is considered.

be satisfied in both the training and testing stages. Compared with the benchmark, the resource utilisation efficiency can be improved by three times with the codebook-based precoding and five times with MRT. The results show that by optimizing the resource allocation for channel estimation and data transmission, I can increase the resource utilisation efficiency significantly.

To better understand the performance gap between the benchmark and the unsupervised learning methods in Table 2.3, I provide the cumulative distribution functions (CDFs) of the SINR achieved by different policies in Fig. 2.8: 1) model-based unsupervised learning with codebook-based precoding (with legend "Codebook"), 2) benchmark with MRT (with legend "Benchmark"). The average SINR achieved by the two policies are 11.2 dB and 11.7 dB, respectively. It is worth noting that the average SINR achieved by the benchmark is higher than that achieved by the learning

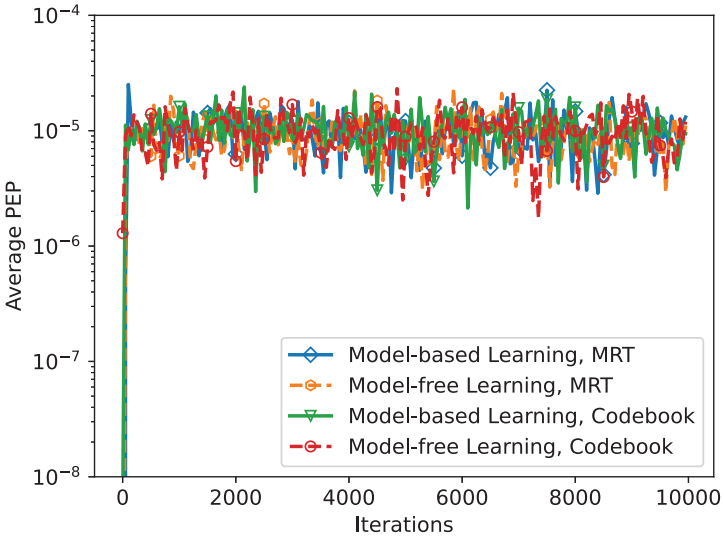


Figure 2.6: Average PEP in the training stage, where the required average PEP is 10^{-5}.

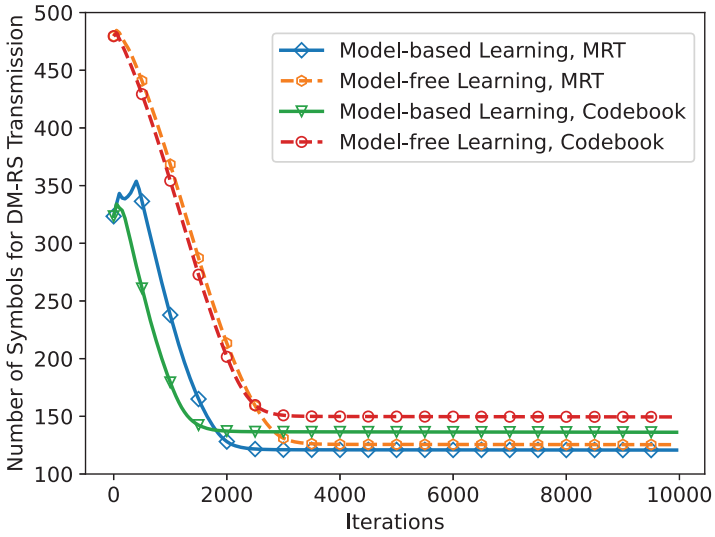


Figure 2.7: Number of DM-RS symbols in the training stage, where the average PEP requirement is considered.

Table 2.3: Resource utilisation efficiency and Average PEP in training and testing stages

Learning Algorithm		Model-Based		Model-Free		Existing Policy
Beamforming		MRT	Codebook	MRT	Codebook	MRT
Resource Utilisation Efficiency (bits/Hz/s)	Training	3.21	2.06	3.21	2.05	-
	Testing	3.21	2.06	3.21	2.05	0.54
Average PEP	Training	1.00e-5	1.00e-5	9.99e-6	1.00e-5	-
	Testing	1.00e-5	1.00e-5	1.00e-5	9.99e-6	9.99e-6

method with the codebook-based precoding. Nevertheless, the CDF obtained from the benchmark has a much longer tail distribution than the other two policies. The resource utilisation efficiency of URLLC is dominated by the tail distribution of the SINR, not the average SINR. Therefore, the resource utilisation efficiency of the learning methods with codebook-based precoding is 3 times higher than the benchmark.

2.5.6 Performance Evaluation for PEP Outage Probability Requirement

In this subsection, I demonstrate the performance of unsupervised learning algorithms regarding the PEP outage probability requirement. Fig. 2.9 depicts the resource allocation efficiency achieved by various algorithms using different precoding techniques. With the PEP outage probability requirement, the learning algorithms need more time to converge than the average PEP requirement. This is because I need to train

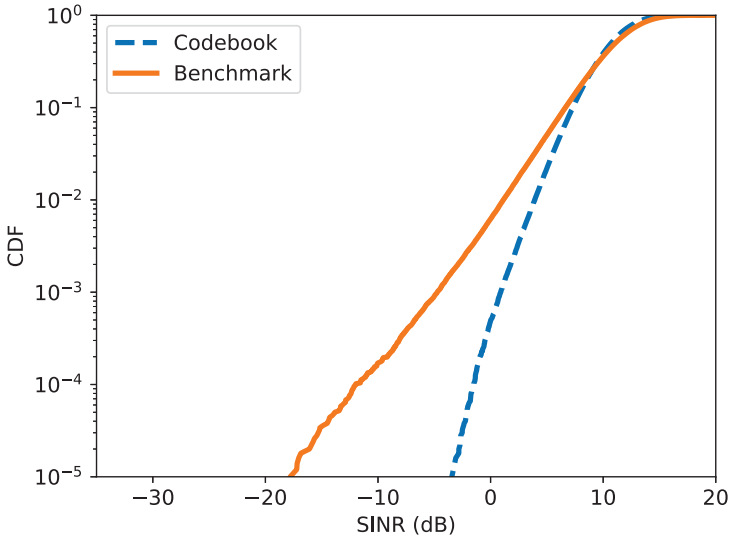


Figure 2.8: CDF of SINR, where the average SNRs of "Codebook" and "Benchmark" are 11.2 dB and 11.7 dB, respectively.

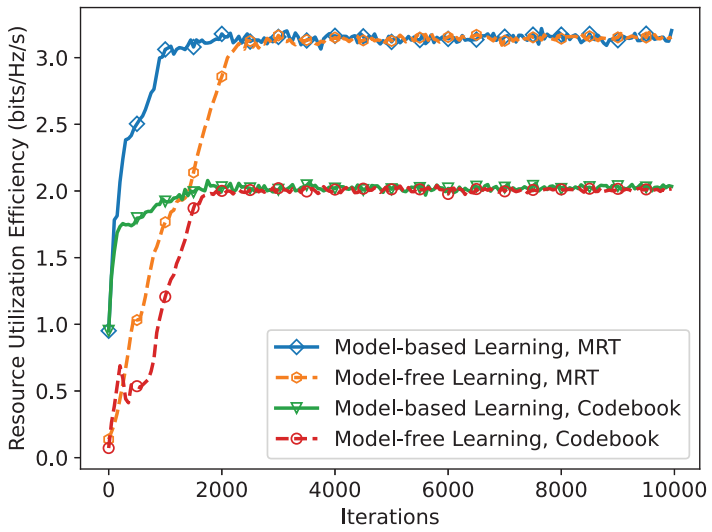


Figure 2.9: Resource utilisation efficiency in the training stage, where the PEP outage probability requirement is considered.

an additional DNN for reliability evaluation. Without the reliability evaluation function, the unsupervised learning algorithm cannot converge within $10,000$ iterations. In Fig. 2.10, I evaluate the PEP outage probabilities achieved by different algorithms with different precoding techniques. The results show that both model-based and model-free learning algorithms can ensure the PEP outage probability with both MRT and codebook-based precoding. To put it differently, my approach can achieve an air-interface latency of 1 ms and a PEP of 10^{-5} with a probability of 99.99%, i.e., $1-\Upsilon$. Fig. 2.11 presents the symbols allocated for DM-RS transmission, and the outcomes are similar to those in Fig. 2.7.

In Table 2.4, I evaluate the resource utilisation efficiency and PEP outage probabilities in both training and testing stages, where 5×10^8 samples of channel realisations and channel estimation errors are used. The results show that in both the training and testing stages, the PEP outage probabilities can meet the PEP outage probability requirement. Compared with the benchmark, the unsupervised learning algorithms can improve the resource utilisation efficiency by three times with the codebook-based precoding and five times with MRT.

2.5.7 Performance Evaluation for Transfer Learning

As I have shown in the previous two subsections, the results with the PEP outage probability requirement are similar. Thus, I take the average PEP requirement as an example to show the performance of transfer learning. I fix the first three layers and fine-tune the last two layers by using the model-based or model-free unsupervised learning algorithm. I first apply transfer learning to dynamic channel resources.

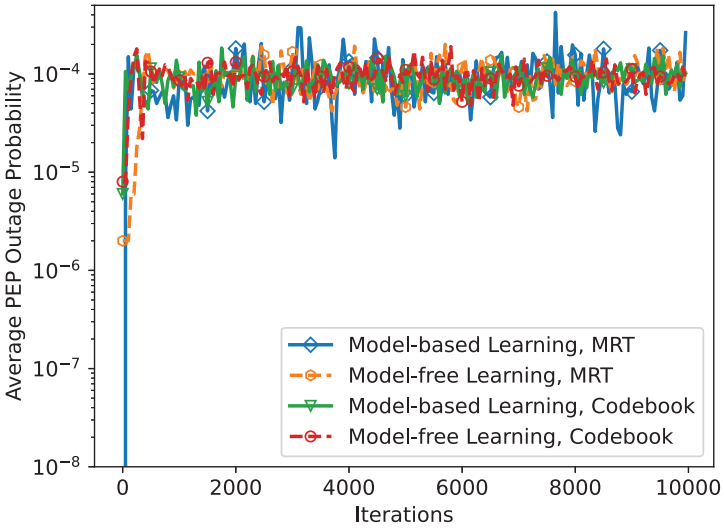


Figure 2.10: PEP outage probability in the training stage, where the required PEP outage probability is 10^{-4}.

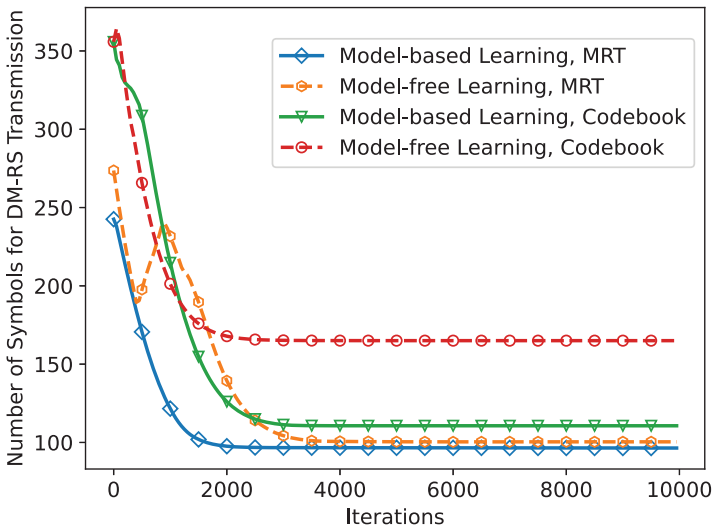


Figure 2.11: Number of DM-RS symbols in the training stage, where the PEP outage probability requirement is considered.

Table 2.4: Resource utilisation efficiency and PEP outage probability in training and testing stages

Learning Algorithm		Model-Based		Model-Free		Existing Policy Model-Based
Beamforming		MRT	Codebook	MRT	Codebook	MRT
Resources utilisation Efficiency (bits/Hz/s)	Training	3.15	2.02	3.15	2.00	-
	Testing	3.15	2.02	3.15	2.00	0.52
PEP Outage Probability	Training	9.89e-5	9.30e-5	9.95e-5	9.69e-5	-
	Testing	9.73e-5	9.37e-5	1.01e-4	9.99e-5	9.95e-05

Specifically, when the time and frequency resources, $N_{\max}$, are dynamic, I use transfer learning to fine-tune the cascaded DNN that is trained with $B = 1$ MHz and $T_f = 1$ ms. Specifically, I fix the bandwidth $B = 1$ MHz and change the frame duration, T_f, from 0.2 ms to 0.8 ms. In Figs. 2.12 and 2.13, I provide the resource utilisation efficiency and the average PEP in the training stage, where $T_f = 0.6$ ms. Without transfer learning, the model-based (model-free) unsupervised learning algorithm converges after 1500 (2000) iterations. With transfer learning, only 600 iterations are needed to fine-tune the cascaded DNN. Thus, the convergence time can be reduced by 70%. The results in Fig. 2.13 show that the average PEP requirement can be satisfied with or without transfer learning.

I further implement transfer learning for various numbers of transmitting antennas and different channel statistics. Specifically, I modified the small-scale channel from Rayleigh fading to Nakagami-m fading [77]. The average PEP requirement is used as an example to demonstrate the training performance of using transfer learning.

The initial DNNs are trained with 4 transmission antennas under the Rayleigh channel. Fig. 2.14 illustrates that with the increase of transmitting antennas to 6, both model-based and model-free algorithms achieve rapid convergence within 500 iterations using transfer learning. In contrast, randomly initialised parameters require over 2000 iterations for the model-based algorithm and 2500 iterations for the model-free algorithm to converge. Furthermore, Fig. 2.15 validates that all algorithms can meet the reliability requirement. Moreover, I have evaluated the resource utilisation efficiency of transfer learning for Nakagami-m fading channels with $m = 3$, as illustrated in Fig. 2.16. For both model-based and model-free algorithms, transfer learning leads to convergence at around 800 iterations, while without transfer learning, convergence requires over 3000 iterations. Fig. 2.17 also verifies that all algorithms satisfy the average PEP requirement. Therefore, my results suggest that transfer learning is also an effective approach for URLLC systems with varying numbers of antennas and different channel statistics.

Table 2.5: Testing performance with Different CSI Observations

CSI Observation	Average PEP		Resource Utilisation Efficiency (bits/Hz/s)		Number of symbols for DM-RS	
	Model-Based	Model-Free	Model-Based	Model-Free	Model-Based	Model-Free
Perfect Channel Gain	1.06e-5	6.73e-6	3.21	3.21	101.21	108.87
Estimated Channel Gain	6.89-6	5.93e-6	3.11	3.13	102.41	115.87
Received Signal Strength	5.43e-6	4.36e-6	2.84	2.83	115.17	121.97

I test the resource utilisation efficiency of model-based and model-free algorithms with different values of $N_{\max}$ in Fig. 2.18. The results show that, as the total number of symbols increases, the resource utilisation efficiency increases, and the gap between the model-based and model-free algorithms decreases. However, the gap between

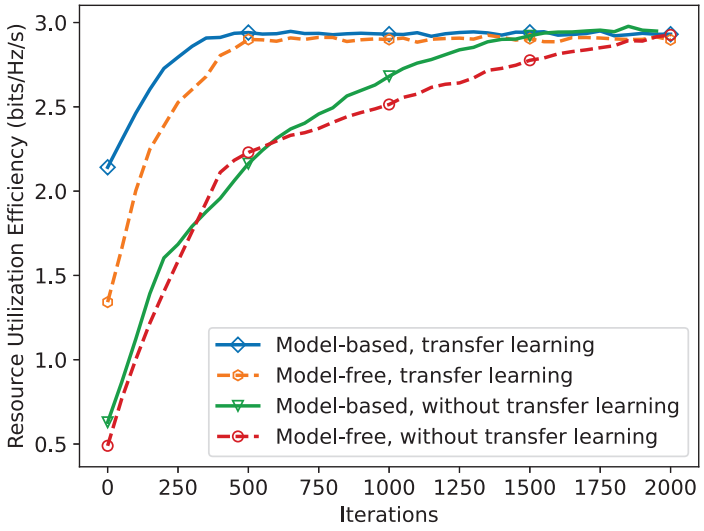


Figure 2.12: Resource utilisation efficiency when $T_f = 0.6$ ms, where the initial DNNs are trained with $T_f = 1$ ms.

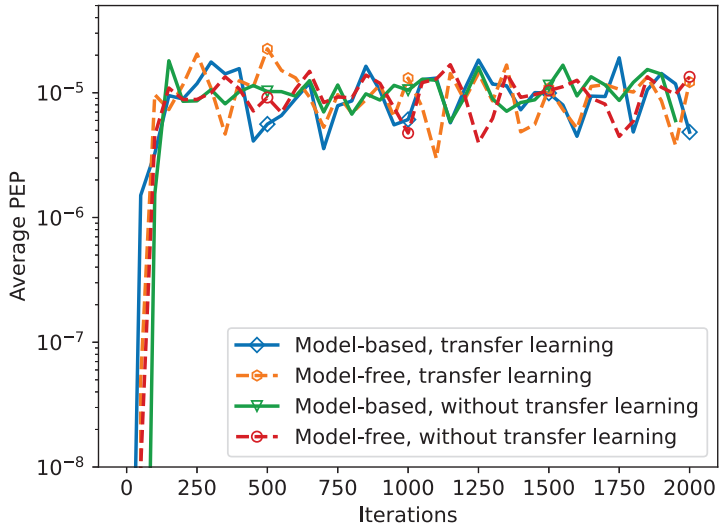


Figure 2.13: Average PEP performance when $T_f = 0.6$ ms, where the initial DNN is trained with $T_f = 1$ ms.

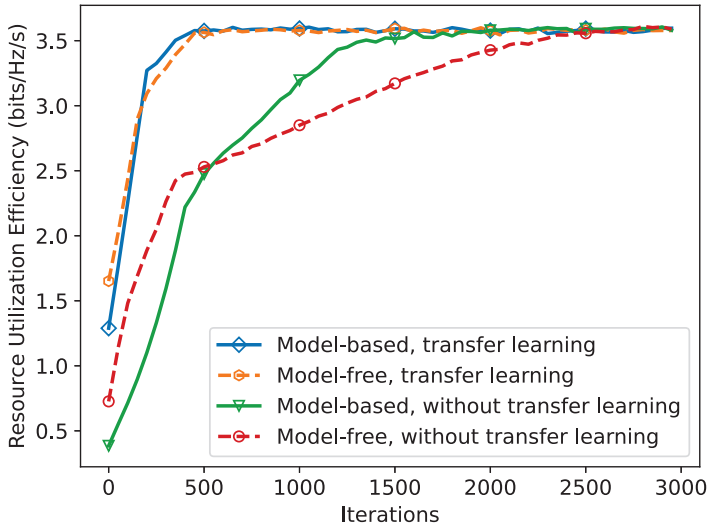


Figure 2.14: Resource utilisation efficiency when $n_t = 6$, where the initial DNNs are trained with $n_t = 4$.

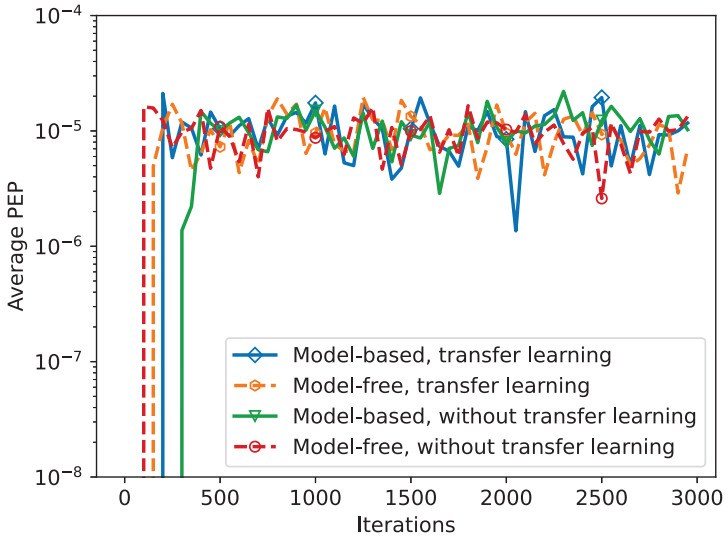


Figure 2.15: Average PEP when $n_t = 6$, where the initial DNNs are trained with $n_t = 4$.

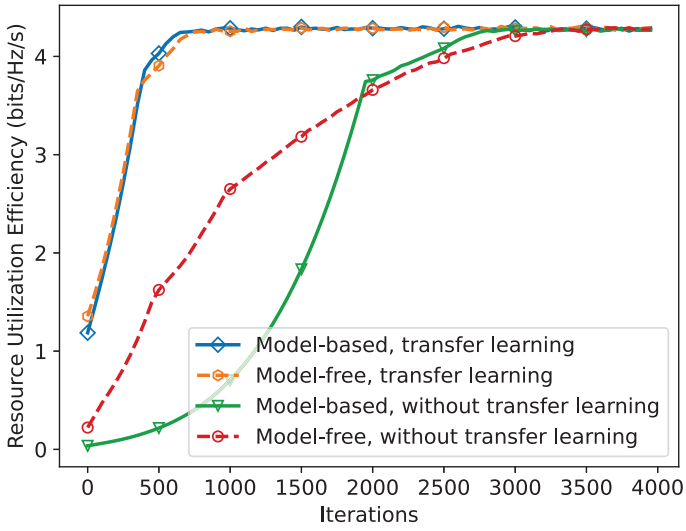


Figure 2.16: Resource utilisation efficiency over Nakagami-m fading channels, and the initial DNNs are trained over Rayleigh fading channels.

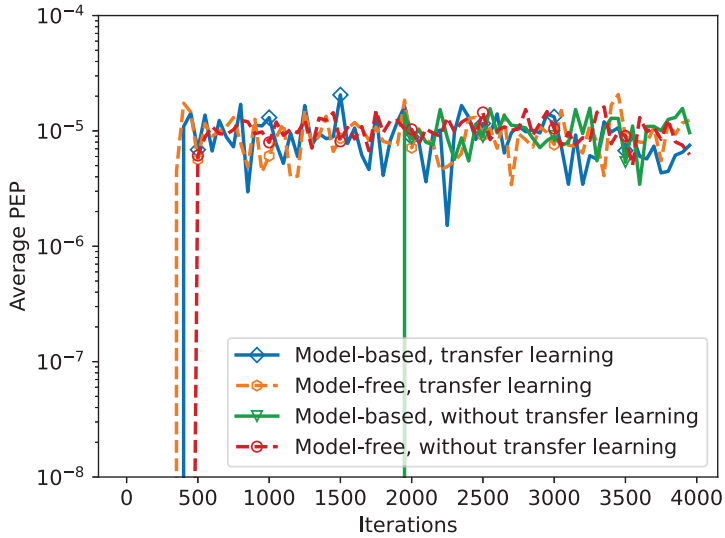


Figure 2.17: Average PEP over Nakagami-m fading channels, and the initial DNNs are trained over Rayleigh fading channels.

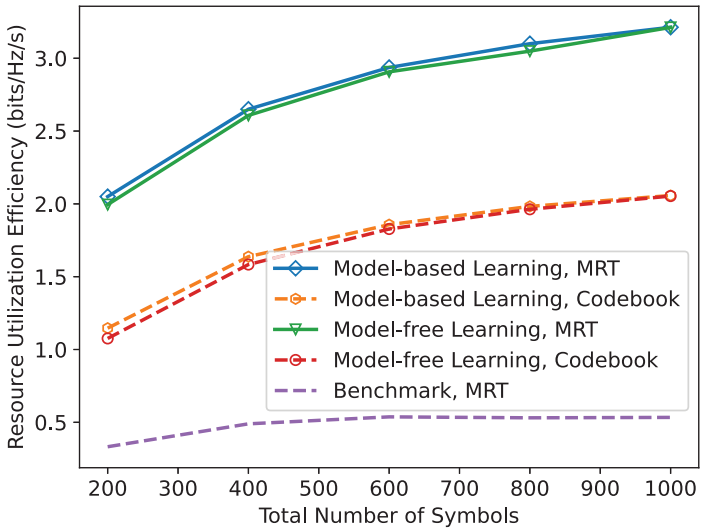


Figure 2.18: Resource utilisation efficiency versus the total number of symbols.

MRT and codebook-based precoding does not change significantly. Compared with the benchmark that uses MRT, my methods can improve the resource utilisation efficiency by up to eight times with MRT and up to five times with the codebook-based precoding.

2.5.8 Performance Evaluation with Different CSI Observations

In this subsection, I evaluate the impact of CSI observations on reliability, resource utilisation efficiency, and resource allocation policy. Since the results are similar to the two types of reliability constraints, I only provide the results under the average PEP requirement. As shown in Table 2.5, the model-based and model-free unsupervised learning algorithms can meet the average PEP requirement with different CSI

observations. By replacing the perfect channel gain with the estimated channel gain, there is only around 2% performance loss in terms of resource utilisation efficiency. The performance loss will be 15%, if the BS only has the received signal strength. By replacing the received signal strength with the estimated channel gain, it is possible to achieve a 10% performance gain in terms of resource utilisation efficiency. It is because in the scenario with the received signal strength, more symbols are needed for DM-RS than the other two kinds of CSI observations.

2.6 Conclusion

In this chapter, I optimised the resource allocation and packet size to maximise the resource utilisation efficiency of URLLC. I developed model-based and model-free unsupervised learning algorithms to find the optimal solutions with different CSI observations, precoding techniques, and reliability requirements. My results showed that both model-based and model-free algorithms could meet the reliability requirements, and the resource utilisation efficiency achieved by the two algorithms is nearly the same. The codebook-based precoding can reduce the overhead of the CSI report, but the resource utilisation efficiency achieved by codebook-based precoding is 40% lower than that achieved by MRT. By evaluating the resource utilisation with different CSI observations, I found that the BS only needs the estimated channel gain to achieve the nearly optimal resource utilisation efficiency, i.e., 2% lower than the scenario with the perfect channel gain. If the BS only has the received signal strength, the resource utilisation efficiency will be 10% lower than that with the estimated channel gain. My results indicated that transfer learning could reduce the convergence time of unsupervised learning methods by 70% when the total amount of resources allocated to

the UE is dynamic.

Chapter 3

Reinforcement Learning for Optimal URLLC Resource Efficiency under Correlated Channel

This chapter considers the optimisation of blocklength allocation for channel estimation and data transmission in MISO ultra-reliable low-latency communication systems. Specifically, I aim to determine a resource allocation strategy to optimise resource utilisation efficiency under a constraint of reliability. I investigate the optimisation problem in correlated channel realisations and formulate the sequential decision-making problem as a partial observation Markov decision process (POMDP). I utilise deep reinforcement learning (DRL) and develop a novel Cascaded-Action Twin Delayed Deep Deterministic policy (CA-TD3) to solve the POMDP problem. I propose

the primal CA-TD3 algorithm and compare its performance with the primal-dual CA-TD3. I validate my model on two channel models: the first-order autoregressive channel model and the clustered delay line (CDL) channel model. The simulation results show that both the primal-type algorithm and the primal-dual one can acquire the optimal strategy on either channel model. However, the primal CA-TD3 can converge remarkably faster than the primal-dual CA-TD3 in terms of reliability.

3.1 Introduction

To meet the reliability requirement in URLLC, the current wireless system uses pilot signals for channel estimation. How to allocate the channel resources for channel estimation and data transmission remains a challenging issue. The limited number of pilot symbols will lead to imperfect channel state information (CSI) and unavoidable decoding packet error probability (PEP). Moreover, satisfying the stringent reliability requirement results in reduced resource utilisation efficiency (i.e., transmitted information bits per time and frequency resource block). The URLLC system needs an intelligent resource allocation policy that can improve resource utilisation efficiency and meet a reliability requirement simultaneously [19]. However, the channel realisations are mostly assumed to be independent and identically distributed (IID) for different frames. Due to the short time scale, the channel coefficient is highly correlated in the temporal domain. Therefore, existing works *et al.* [35, 37] proposed resource allocation strategies based on the first-order autoregressive model [68], which is used to simulate the temporally correlated channel. However, how to optimise the resource utilisation efficiency in practical correlated channel realisations remains unclear.

Resource allocation for time-varying correlated channels is a sequential decision-making problem, which can be solved through deep reinforcement learning (DRL) algorithms [44, 78]. However, the unconstrained DRL algorithms will try some bad actions, which will eliminate the stringent reliability requirements in URLLC systems. To improve the exploration safety of DRL algorithms, the authors in [45, 46] built constrained DRL frameworks using the primal-dual algorithm. Nevertheless, the primal-dual DRL algorithms reveal that the convergence of the constraint condition is slow. Moreover, the performance of the primal-dual algorithms highly relies on the parameter tuning of the Lagrangian multiplier, which significantly increases the training difficulty.

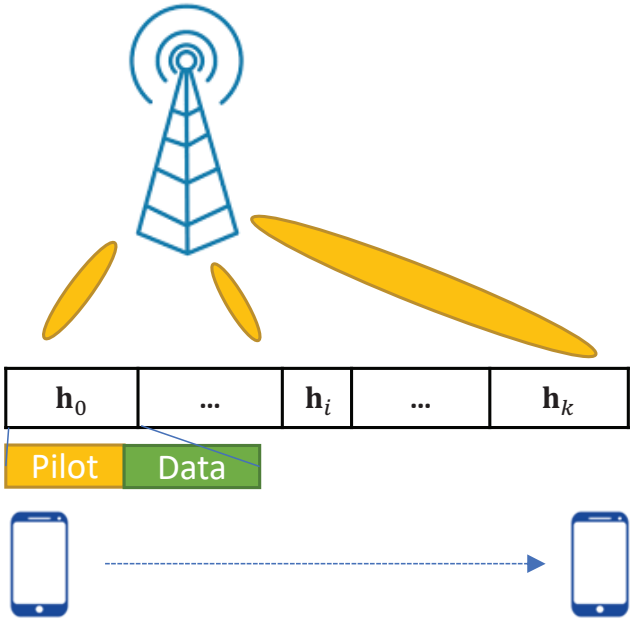


Figure 3.1: Resource allocation in temporally correlated channel realisations

In this chapter, I aim to optimise a resource allocation policy over temporally correlated fading channels, where resource utilisation efficiency is maximised subject

to a PEP constraint. Due to the unknown channel realisation and imperfect CSI, the BS can only have limited channel observations. Thus, I formulate the problem as a partially observable Markov decision process (POMDP). I design a novel constrained DRL framework to solve the POMDP problem based on Twin Delayed Deep Deterministic policy (TD3)[79], i.e., the cascaded-action TD3 (CA-TD3). I develop a cascaded deep neural network (DNN) structure to determine the actions. Inspired by *et al.* [67], I train the policy using a primal domain approach and compare it with the primal-dual DRL algorithm[80]. I validate my algorithm on the first-order autoregressive channel model [68] and the clustered delay line (CDL) channel model [69]. My results show the superiority of the primal CA-TD3 in terms of the convergence time compared with the primal-dual CA-TD3.

3.2 System Model

In this section, I first present my channel model. Then, I provide the reliability metric with imperfect CSI. Finally, I formulate the problem as a POMDP problem.

3.2.1 Channel Model

I consider a downlink MISO system, where a base station (BS) with T transmit antennas serves a mobile single-antenna user equipment (UE). I define k as the number of frames transmitted in time-correlated wireless channels, as shown in Fig. 3.1. The i-th frame, $i \triangleq \{0, 1, \cdots, k\}$, is transmitted in the i-th channel realisation, which is considered to be quasi-static as the frame duration is smaller than the coherence time [15]. For each frame, I assume that the total time-frequency resources are

N orthogonal frequency-division multiplexing (OFDM) symbols, which are used to transmit pilot symbols and data symbols. I denote the number of pilot symbols as m_i, and the number of data symbols as $n_i = N - m_i$ for the i-th frame. The received symbols $\mathbf{y}_i$ at the UE is expressed as

$$\mathbf{y}_i = \sqrt{\alpha p}\mathbf{h}_i^H \mathbf{w}_i \mathbf{x}_i + \mathbf{z}_i, \tag{3.2.1}$$

where $\alpha \in \mathbb{R}^+$ is the constant large-scale channel fading, $\mathbf{h}_i \in \mathbb{C}^{\mathrm{T}\times 1}$ is the small-scale channel coefficient, $p \in \mathbb{R}^+$ is the transmit power, $\mathbf{w}_i = \mathbf{h}_i/|\mathbf{h}_i|$ is the maximum ratio transmission (MRT) precoding vector, $\mathbf{x}_i$ is the transmitted symbols, and $\mathbf{z}_i \sim \mathcal{CN}(0, \sigma^2_{\mathbf{z}_i})$ is the Additive white Gaussian noise (AWGN). Sequential small-scale channel coefficients, e.g., $\mathbf{h}_{i+1}$ and $\mathbf{h}_i$, are temporally correlated.

3.2.2 Reliability Metric with Imperfect CSI

Due to the limited time-frequency resources allocated for pilot symbols, there are unavoidable channel estimation errors, denoted by $\mathbf{e}_i \in \mathbb{C}^{\mathrm{T}\times 1}$. I denote the estimated channel of $\mathbf{h}_i$ as $\hat{\mathbf{h}}_i$. The channel estimation errors are defined as the difference between the real channel realisation and the estimated one following

$$\mathbf{e}_i = \mathbf{h}_i - \hat{\mathbf{h}}_i. \tag{3.2.2}$$

With the minimum mean-square-error (MMSE) channel estimation in Rayleigh fading channels, $\mathbf{e}_i$ follows circularly symmetric complex Gaussian distributions, i.e., $\mathbf{e}_i \sim \mathcal{CN}(0, \sigma^2_{\mathbf{e}_i})$, where $\sigma^2_{\mathbf{e}_i} = (\frac{1}{\sigma^2_{\mathbf{h}_i}} + \frac{m_i \alpha p}{\sigma^2_{\mathbf{z}}\mathrm{T}})^{-1}$ [16].

Based on the channel estimation errors, I can derive the signal-to-interference-plus-noise ratio (SINR) for $\mathbf{h}_i$ as

$$\gamma_i = \frac{\alpha p |\hat{\mathbf{h}}_i^H \mathbf{w}_i|^2}{\alpha p |\mathbf{e}_i^H \mathbf{w}_i|^2 + |\mathbf{z}_i|^2}. \tag{3.2.3}$$

Then, if D_i bits are encoded into n_i symbols, the achievable PEP at $\mathbf{h}_i$ is approximated by [10] as

$$\epsilon(\gamma_i) \approx Q\left([C(\gamma_i) - R_i]\sqrt{\frac{n_i}{V(\gamma_i)}}\right), \tag{3.2.4}$$

where $R_i = D_i/n_i$ is the achievable data rate, $C(\gamma_i) = \log_2(1+\gamma_i)$ is the channel capacity, $V(\gamma_i) = (1-(1+\gamma_i)^{-2})\log_2^2 e$ is the channel dispersion, $Q(\cdot)$ is the Q-function, i.e., $Q(x) = \frac{1}{\sqrt{2\pi}}\int_x^\infty e^{-\frac{t^2}{2}}\mathrm{d}t$. Therefore, if the number of pilot symbols increases, the channel estimation errors decrease, the SINR rises, and the PEP is reduced. To manage the PEP constraint, it is crucial to control the number of pilot symbols.

Due to channel fading, it is difficult to achieve target PEP with probability one. Therefore, I define PEP outage probability, which represents the percentage of packets that cannot meet the PEP requirement, to evaluate the reliability. I denote the PEP requirement of each packet as ϵ_{q}. If a packet transmission satisfies the PEP requirement, I set an indicator function equal to one, denoted by $\mathbb{1}\{\epsilon(\gamma_i) > \epsilon_{\mathrm{q}}\} = 1$. Otherwise, $\mathbb{1}\{\epsilon(\gamma_i) > \epsilon_{\mathrm{q}}\} = 0$. Thus, the PEP outage probability is defined as $\mathbb{E}_{\gamma_i}(\mathbb{1}\{\epsilon(\gamma_i) > \epsilon_{\mathrm{q}}\})$.

3.2.3 Constrained POMDP Problem Formulation

Since the BS cannot learn full CSI, the sequential decision problem can be formulated as a POMDP problem, defined by a tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{O}, \mathcal{R}, \mathcal{C}, \mathcal{T}\rangle$, where $\mathcal{S}$ is the state environment, $\mathcal{A}$ is the action space taken by the agent, $\mathcal{O}$ is the partial observation of the unknown state environment, $\mathcal{R}$ is the instantaneous reward, $\mathcal{C}$ is the instantaneous cost, and $\mathcal{T}$ is the transition. The detailed definition is as follows:

State, Observation, and Transition

The state for the i-th frame, denoted by $\mathbf{s}_i \in \mathcal{S}$, refers to the channel realisation $\mathbf{h}_i$. I assume the BS can obtain an estimation of $\mathbf{h}_{i-1}$ at the beginning of the i-th frame. Since $\mathbf{h}_{i-1}$ and $\mathbf{h}_i$ are highly correlated, $\hat{\mathbf{h}}_{i-1}$ serves as the observation in the i-th frame, denoted by o_i. The BS will use o_i for sequential decisions. The transition $\mathrm{t}_i \in \mathcal{T}$ refers to the change from $\mathbf{s}_i$ to $\mathbf{s}_{i+1}$.

Action

Two actions need to be decided by the BS: the packet size, D_i, and the number of data symbols, n_i. Therefore, I denote the actions as $a_i = \{D_i, n_i\}, a_i \in \mathcal{A}$. The packet size D_i is in continuous infinite action space. In practice, the number of data symbols n_i ought to be in discrete finite action space in terms of the subcarrier frequency space and transmission time interval (TTI) defined in the 5G NR standard. However, to provide a more accurate estimation of the resource allocation policy without discrete sampling errors, I assume that n_i is in the continuous finite action space.

Instantaneous Reward and Cost

I define the instantaneous reward at the $\mathbf{h}_{i+1}$ as the instantaneous resource utilisation efficiency in the i-th frame, i.e., $r_i = D_i/N$. The cost is defined as the indicator value at the $\mathbf{h}_{i+1}$, i.e., $c_i = \mathbb{1}\{\epsilon(\gamma_i) > \epsilon_{\mathrm{q}}\}$.

Policy

A policy π is defined as the mapping from the observation to the actions. With policy π, the long-term discounted reward is denoted as

$$R^{\pi} = \mathbb{E}_{\pi}[\sum_{i=0}^{k} \Gamma^{i} r_i], \tag{3.2.5}$$

where $\Gamma \in (0, 1]$ is the discount factor. Similarly, the long-term discounted cost with π is given by

$$C^{\pi} = \mathbb{E}_{\pi}[\sum_{i=0}^{k} \Gamma^{i} c_i]. \tag{3.2.6}$$

POMDP Problem

I aim to find the optimal policy π^* such that I can maximise the long-term reward while controlling the long-term cost not exceeding the constraint. Therefore, the optimisation problem is formulated as follows:

$$\pi^* = \arg\max_{\pi} R^{\pi}, \tag{3.2.7}$$

$$\text{s.t.} \quad C^{\pi} \leq \frac{\Upsilon}{1-\Gamma}, \tag{3.2.7a}$$

$$N_i \leq N - \mathrm{T}, \tag{3.2.7b}$$

where Υ is the maximum tolerable PEP outage probability, and (3.2.7b) guarantee the number of symbols for channel estimation is equal to or larger than the number of antennas [15].

The problem can be solved through primal-dual and Lagrangian multiplier methods [74]. The Lagrangian function can be defined as follows:

$$L(\pi, \lambda) = R^{\pi} - \lambda(\frac{C^{\pi}(1-\Gamma)}{\Upsilon} - 1), \tag{3.2.8}$$

where λ is the Lagrangian multiplier. The constrained POMDP problem is converted to an unconstrained min-max problem as follows:

$$\pi^*, \lambda^* = \arg\min_{\lambda} \max_{\pi} L(\pi, \lambda). \tag{3.2.9}$$
$$\text{s.t.} \quad (3.2.7b).$$

3.3 Deep Reinforcement Learning

In this section, I introduce my DRL framework developed based on TD3. The conventional Deep Deterministic Policy Gradient (DDPG) algorithm [81] overestimates the long-term reward and underestimates the long-term cost, which results in an unstable update of actor networks. To overcome this drawback, DDPG is extended to TD3 by introducing twin critic deep neural networks (DNNs). In TD3, the agent can choose the preferable output from the two DNNs as the critic value, which leads to a more accurate estimation of long-term rewards and costs. Moreover, since I have two actions to be determined, I develop the Cascaded-Action TD3 (CA-TD3) architecture. I first consider the primal-dual methods to solve the problem (3.2.7). To improve the training stability and control the constraint efficiently, I further develop a primal algorithm based on the Constraint-Rectified Policy Optimisation (CRPO) method [67].

3.3.1 CA-TD3 Architecture

The TD3 follows the actor-critic structure. The actor networks include the DNNs, which represent the policy. Since the packet size depends on the number of symbols

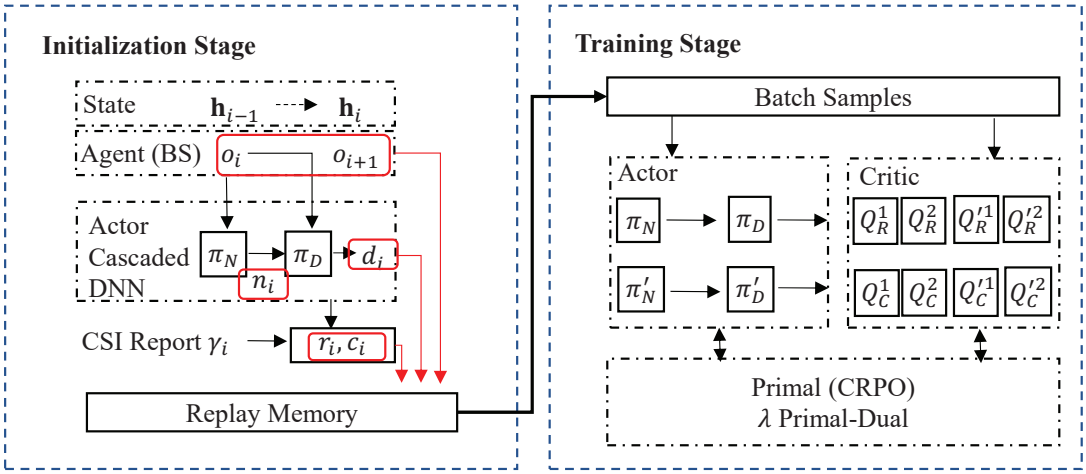


Figure 3.2: CA-TD3 Architecture

for data transmission, I design a cascaded DNN structure in the actor network. The first DNN in the cascaded DNN represents the mapping from channel observations to the number of data symbols and is denoted by $\pi_N(o_i|\theta_{\pi_N})$. The channel observations and the number of data symbols are further used as the input of the second DNN, which is denoted as $\pi_D(o_i, \pi_N(o_i|\theta_{\pi_N})|\theta_{\pi_D})$. The output of the second DNN is the packet size. The corresponding target networks are denoted by $\pi'_N(o_i|\theta'_{\pi_N})$ and $\pi'_D(o_i, \pi'_N(o_i|\theta'_{\pi_N})|\theta'_{\pi_D})$. θ_{π_N},θ_{π_D}, θ'_{π_N}, and θ'_{π_D} are the parameters.

The TD3 is extended from the DDPG algorithm by introducing twin critic DNNs. The twin critic networks are used to evaluate the long-term reward, denoted by $Q_R^{\phi}(o_i, \pi_N(o_i|\theta_{\pi_N}), \pi_D(o_i, \pi_N(o_i|\theta_{\pi_N})|\theta_{\pi_D})|\theta_{Q_R}^{\phi})$, and the long-term cost, denoted by $Q_C^{\phi}(o_i, \pi_N(o_i|\theta_{\pi_N}), \pi_D(o_i, \pi_N(o_i|\theta_{\pi_N})|\theta_{\pi_D})|\theta_{Q_C}^{\phi})$, where $\phi \triangleq \{1, 2\}$. The corresponding target critic networks are denoted by $Q_R'^{\phi}(o_i, \pi'_N(o_i|\theta'_{\pi_N}), \pi'_D(o_i, \pi'_N(o_i|\theta'_{\pi_N})|\theta'_{\pi_D})|\theta'^{\phi}_{Q_R})$ and $Q_C'^{\phi}(o_i, \pi'_N(o_i|\theta'_{\pi_N}), \pi'_D(o_i, \pi'_N(o_i|\theta'_{\pi_N})|\theta'_{\pi_D})|\theta'^{\phi}_{Q_C})$. $\theta_{Q_R}^{\phi}$, $\theta_{Q_C}^{\phi}$, $\theta'^{\phi}_{Q_R}$, and $\theta'^{\phi}_{Q_C}$ are the parameters. The overall CA-TD3 architecture is shown in Fig. 3.2.

3.3.2 Primal-Dual CA-TD3

During the initialisation procedure, the agent will first generate the actions, i.e., the packet size and the number of data symbols, through the actor networks. The packet size and the number of data symbols are generated along with exploration:

$$n_i = \pi_N(o_i|\theta_{\pi_N}) + \omega_N, \tag{3.3.1}$$

$$D_i = \pi_N(o_i, N_i|\theta_{\pi_D}) + \omega_D, \tag{3.3.2}$$

where ω_N and ω_D are the exploration noise, following normal distributions with variance $\sigma^2_{\omega_N}$ and $\sigma^2_{\omega_D}$, which will decay by a ratio $\tau_\omega \in (0, 1)$ against training episodes.

TD3 is an off-policy algorithm and uses the experience replay buffer to save historical transitions $\langle o_i, D_i, n_i, r_i, c_i, o_{i+1}\rangle$. The training stage starts when the replay buffer is full. In each training step, the agent will select a random batch of M samples from the replay memory. The batch of samples is denoted by $\langle o_m, D_m, N_m, r_m, c_m, o_{m+1}\rangle$, $m \triangleq \{1, 2, \cdots, M\}$, which can be used to compute the target reward and cost values through the Bellman equation:

$$\begin{aligned} R_m = r_m + \Gamma \min_{\phi} Q'^{\phi}_R(o_{m+1}, \pi'_N(o_{m+1}|\theta'_{\pi_N}), \\ \pi'_D(o_{m+1}, \pi'_N(o_{m+1}|\theta'_{\pi_N})|\theta'_{\pi_D})|\theta'^{\phi}_{Q_R}) \end{aligned} \tag{3.3.3}$$

$$\begin{aligned} C_m = c_m + \Gamma \max_{\phi} Q'^{\phi}_C(o_{m+1}, \pi'_N(o_{m+1}|\theta'_{\pi_N}), \\ \pi'_D(o_{m+1}, \pi'_N(o_{m+1}|\theta'_{\pi_N})|\theta'_{\pi_D})|\theta'^{\phi}_{Q_C}). \end{aligned} \tag{3.3.4}$$

The critic networks can be trained by minimising the mean square temporal difference

(TD) errors, which are defined as:

$$L_{Q_R^\phi} = \frac{1}{M}\sum_{m=1}^{M}\Big[R_m - Q_R^\phi(o_m, \pi_N(o_m|\theta_{\pi_N}), \pi_D(o_m, \pi_N(o_m|\theta_{\pi_N})|\theta_{\pi_D})|\theta_{Q_R}^\phi)\Big]^2, \tag{3.3.5}$$

$$L_{Q_C^\phi} = \frac{1}{M}\sum_{m=1}^{M}\Big[C_m - Q_C^\phi(o_m, \pi_N(o_m|\theta_{\pi_N}), \pi_D(o_m, \pi_N(o_m|\theta_{\pi_N})|\theta_{\pi_D})|\theta_{Q_C}^\phi)\Big]^2. \tag{3.3.6}$$

The actor networks are updated by maximising the Lagrangian function through stochastic gradient ascent. However, the conventional primal-dual algorithm shows unstable and slow convergence performance. It is mainly because the optimisations of the primal and dual domains have different requirements for the parameter tuning of the Lagrangian multiplier. To combat this challenge, I improve the conventional primal-dual algorithm by introducing another normalisation coefficient, β, during the optimisation of the dual variable. Furthermore, I set $\beta = \beta_0$ if the constraint is met. Otherwise, $\beta = \beta_1$. Therefore, the policy gradient is derived as follows:

$$\begin{aligned}\nabla_{\theta_i} L =& \frac{1}{M}\sum_{m=1}^{M}\nabla_{\theta_i}\Big(\min_{\phi} Q_R^\phi(o_m, \pi_N(o_m|\theta_{\pi_N}), \pi_D(o_m, \pi_N(o_m|\theta_{\pi_N})|\theta_{\pi_D})|\theta_{Q_R}^\phi) \\ &- \beta\lambda(\max_{\phi} Q_C^\phi(o_m, \pi_N(o_m|\theta_{\pi_N}), \pi_D(o_m, \pi_N(o_m|\theta_{\pi_N})|\theta_{\pi_D})|\theta_{Q_C}^\phi) - \frac{\Upsilon}{1-\Gamma})\Big),\end{aligned} \tag{3.3.7}$$

where $\theta_i \triangleq \{\theta_{\pi_D}, \theta_{\pi_N}\}$. θ_i are updated as follows:

$$\theta_i^{(j+1)} = \theta_i^{(j)} + \eta_i^{(j)}\nabla_{\theta_i} L, \tag{3.3.8}$$

where j is the training step, $\eta_i^{(j)} \triangleq \{\eta_{\pi_N}^{(j)}, \eta_{\pi_D}^{(j)}\}$ denotes the step size for π_N and π_D at

Algorithm 4 Primal-Dual CA-TD3

Require: initial parameter $\theta^{(0)}$, $\lambda^{(0)}$

1: **for** $e = 0, 1, 2, \cdots$ **do**
2: Initialise $\mathbf{h}_0$ based on Rayleigh distribution
3: **for** $i = 0, 1, 2, \cdots, k$ **do**
4: Generate one sample of $\mathbf{h}_i$ according to autoregressive model
5: Evaluate $\langle o_i, D_i, N_i, r_i, c_i, o_{i+1} \rangle$ and save the transition into ER buffer
6: Select batch from replay memory $\langle o_m, D_m, N_m, r_m, c_m \rangle, m = \{1, 2, \cdots, M\}$
7: Calculate TD errors following (3.3.3) to (3.3.6). Update Q_R^ϕ and Q_C^ϕ
8: Update π_N and π_D following (3.3.8) and dual variable following (3.3.10)
9: Update the target DNNs following (3.3.15) to (3.3.18).

the j-th step. The dual variable λ can be updated with the policy gradient:

$$\nabla_\lambda L = \frac{1}{M} \sum_{m=1}^{M} \Big(\max_{\phi} \beta Q_C^\phi(o_m, \pi_N(o_m|\theta_{\pi_N}), \pi_D(o_m, \pi_N(o_m|\theta_{\pi_N})|\theta_{\pi_D})|\theta_{Q_C}^\phi) - \frac{\Upsilon}{1-\Gamma} \Big), \tag{3.3.9}$$

$$\lambda^{(j+1)} = \Big[\lambda^{(j)} - \eta_\lambda^{(j)} \nabla_\lambda L \Big]^+, \tag{3.3.10}$$

where $\eta_\lambda^{(j)}$ is the step size of the Lagrangian multiplier at the j-th step and $[x]^+ = \max\{0, x\}$. Then, I update the target networks by (3.3.15) to (3.3.18). The primal-dual CA-TD3 algorithm is summarised in Algorithm 4.

3.3.3 Primal CA-TD3

Different from the primal-dual methods [74, 80], which optimise dual variables to satisfy the constraint, the CRPO algorithm provides a primal-type method that can

promptly switch the policy optimisation between objective improvement and constraint satisfaction. Specifically, if the predicted long-term cost can meet the constraint of (3.2.7a), the policy is updated by maximizing the long-term reward. Otherwise, the policy is updated to minimise the constraint violation. I apply the CRPO to update the actor networks. I first measure the predicted PEP outage probability of the training batch and compare it with Υ. If $\mathbb{E}_m[\max_\phi Q_C^\phi(o_m, \pi_N(o_m|\theta_{\pi_N}), \pi_D(o_m, \pi_N(o_m|\theta_{\pi_N})|\theta_{\pi_D})|\theta_{Q_C}^\phi)] \leq \Upsilon/(1-\Gamma) + \upsilon$, where υ is a small positive parameter that indicates the tolerance of constraint violation, I apply the stochastic gradient ascent to maximise the long-term reward following

$$\nabla_{\theta_g} L_R = \frac{1}{M} \sum_{m=1}^{M} \nabla_{\theta_g} \min_{\phi} Q_R^\phi(o_m, \pi_N(o_m|\theta_{\pi_N}), \pi_D(o_m, \pi_N(o_m|\theta_{\pi_N})|\theta_{\pi_D})|\theta_{Q_R}^\phi). \tag{3.3.11}$$

$$\theta_g^{(j+1)} = \theta_g^{(j)} + \eta^{(j)} \nabla_{\theta_g} L_R, \tag{3.3.12}$$

where $\theta_g \triangleq \{\theta_{\pi_N}, \theta_{\pi_D}\}$, j refers to the j-th training step, and $\eta^{(j)}$ is the learning rate at the j-th training step. Once $\mathbb{E}_m[\max_\phi Q_C^\phi(o_m, \pi_N(o_m|\theta_{\pi_N}), \pi_D(o_m, \pi_N(o_m|\theta_{\pi_N})|\theta_{\pi_D})|\theta_{Q_C}^\phi)] > \Upsilon/(1-\Gamma) + \upsilon$, I apply stochastic gradient descent to minimise the long-term cost,

$$\nabla_{\theta_g} L_C = \frac{1}{M} \sum_{m=1}^{M} \nabla_{\theta_g} \max_{\phi} Q_C^\phi(o_m, \pi_N(o_m|\theta_{\pi_N}) \pi_D(o_m, \pi_N(o_m|\theta_{\pi_N})|\theta_{\pi_D})|\theta_{Q_C}^\phi). \tag{3.3.13}$$

$$\theta_g^{(j+1)} = \theta_g^{(j)} - \eta^{(j)} \nabla_{\theta_g} L_C. \tag{3.3.14}$$

Algorithm 5 Primal CA-TD3

Require: initial parameter $\theta^{(0)}$

for $e = 0, 1, 2, \cdots$ **do**
  Initialise $\mathbf{h}_0$ based on Rayleigh distribution
  for $i = 0, 1, 2, \cdots, k$ **do**
    Generate one sample of $\mathbf{h}_i$ according to autoregressive model
    Evaluate $\langle o_i, D_i, N_i, r_i, c_i, o_{i+1} \rangle$ and save the transition into ER buffer
    Select batch from replay memory $\langle o_m, D_m, N_m, r_m, c_m \rangle$, where $m = \{1, 2, \cdots, M\}$
    Calculate TD errors following (3.3.3) to (3.3.6). Update Q_R^{ϕ} and Q_C^{ϕ}
    if predicted long-term cost is less or equal to $\Upsilon/(1-\Gamma) + \upsilon$ **then**
      Update the π_N and π_D following (3.3.12)
    else
      Update the π_N and π_D following (3.3.14)
    Update the target DNNs following (3.3.15) to (3.3.18)

Finally, I can update the target networks following

$$\theta'_{\pi_N} \triangleq \tau\theta_{\pi_N} + (1+\tau)\theta'_{\pi_N}, \tag{3.3.15}$$

$$\theta'_{\pi_D} \triangleq \tau\theta_{\pi_D} + (1+\tau)\theta'_{\pi_D}, \tag{3.3.16}$$

$$\theta'^{\phi}_{Q_R} \triangleq \tau\theta^{\phi}_{Q_R} + (1+\tau)\theta'^{\phi}_{Q_R} \tag{3.3.17}$$

$$\theta'^{\phi}_{Q_C} \triangleq \tau\theta^{\phi}_{Q_C} + (1+\tau)\theta'^{\phi}_{Q_C}, \tag{3.3.18}$$

where $\tau \ll 1$ is the soft target update factor. The primal CA-TD3 algorithm is summarised in Algorithm 5.

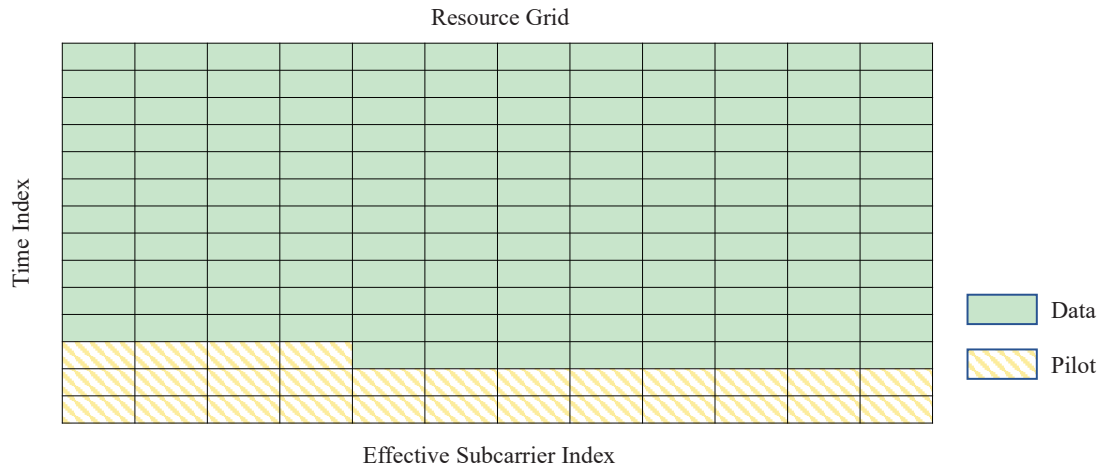


Figure 3.3: Mapping of pilot and data symbols in each resource grid.

3.4 Simulation Results

3.4.1 Simulation Setup

I follow 3GPP TS 38.211 [82] to design the resource grid, where the TTI is 1 ms, the subcarrier spacing is 15 kHz, the number of effective subcarriers is 12, and the number of OFDM symbols in one TTI is 14. Therefore, during each TTI, the effective bandwidth is 180 kHz, and the total number of OFDM symbols is 168. I set the resource budget of each URLLC frame as 4 TTIs. I first map all the pilot symbols in the frequency domain of each resource grid. After all the subcarriers of one symbol duration are occupied, I map the pilot symbols to the time domain. The remaining symbols are allocated for data transmission. For example, if there are 28 pilot symbols in one resource grid, the pilot pattern is shown in Fig. 3.3. Note that I use this basic mapping solution to validate the performance of my proposed model. The impact of how to map the pilot symbols to the resource grid will be left as future work. The large-scale channel fading follows the path-loss model, i.e., $\alpha = -35.3 - 37.6 \log_{10}(d)$,

where d (m) is the distance between the BS and the UE. The parameters of the downlink MISO system are summarised in Table. 3.1 unless otherwise specified.

3.4.2 Channel Models

Two correlated channel models are considered in this chapter: the first-order autoregressive model [68] and the CDL channel model [69].

First-order Autoregressive Model

The correlated channel coefficient is generated according to an autoregressive model following

$$\mathbf{h}_{i+1} = \varphi \mathbf{h}_i + \varepsilon_i, \tag{3.4.1}$$

where φ is the correlation coefficient and $\varepsilon_i \sim \mathcal{CN}(0, 1-\varphi^2)$ is the random white noise during the transition. The initial channel realisation $\mathbf{h}_0 \sim \mathcal{CN}(0, 1)$. The correlation coefficient is set as $\varphi = 0.9$.

CDL Channel Model

The CDL channel model is widely used for generating link-level radio channel impulse responses [83, 84]. The TR38.901 standard defines five types of CDL channel models, including CDL-A, CDL-B, and CDL-C for non-line-of-sight channels, while CDL-D and CDL-E are for line-of-sight channels. The CDL channel contains information about different delay spreads, powers, angles of departures, mobile speeds, and many other significant channel features. I use the CDL-A scenario to simulate the fading channels, where the carrier frequency is 3.5 GHz, the user speed is 3 m/s, and the delay spread is 10 ns.

Table 3.1: Simulation parameters

Parameters	Values
Number of subcarriers	12
Number of OFDM symbols	14
Subcarrier spacing	15 kHz
Transmission time interval	1 ms
BS antenna	4
Distance	300 m
Transmission power	23 dBm
Noise spectral density	173 dBm/Hz
Number of correlated channels	50
PEP requirement	10^{-5}
PEP outage probability requirement	0.1

3.4.3 DRL Setup

For the CA-TD3 framework, the weights of all DNNs are initialised following normal distribution, and the bias is fixed at 0.1. The cascaded DNN structure is composed of a three-layer DNN (with neurons 32/16/1) and a four-layer DNN (8/4/2/1). Regarding the critic networks, I use a three-layer DNN (32/4/1) for the long-term reward and a three-layer (8/4/1) for the long-term cost. For all the DNNs, the hidden layer uses a leaky Rectified Linear Unit (ReLu) with a ratio of 0.1 as the activation function. The output layers of the cascaded DNN use hyperbolic tangent, while the output layers of critic DNNs are ReLu. The packet size is adjusted in the range of $[0, 8000]$ bits. The learning rate values for Q_R, Q_C, π_N, and π_D are 0.01, 0.01, 0.0001, 0.005, respectively.

The learning rate of the Lagrangian multiplier is 0.005. The batch size is set as 2000, and the replay buffer size is set as 5000. The discount factor is 0.99. The update factor of target networks is 0.05. The tolerance of CRPO is set at 0.1. The reward and cost are evaluated based on the mean value of the latest 1000 records. A learning rate decay of 0.9 is applied for training the actor networks when the long-term cost exceeds the boundary.

3.4.4 Performance Evaluation

In this section, I first present the challenge of parameter tuning in the primal-dual method. Then, I compare the proposed primal CA-TD3 with the enhanced primal-dual algorithm in different channel models. Finally, I test my trained models by comparing them with an existing benchmark.

Parameter Tuning for Primal-Dual CA-TD3

Compared to the primal CA-TD3, one critical drawback of the primal-dual method is the challenge in parameter tuning, as it introduces an extra Lagrangian multiplier. To address the tuning of the Lagrangian multiplier, I enhance the conventional primal-dual algorithm by using a normalisation coefficient. I choose the autoregressive channel model for training. I first display the training performance of selecting different normalisation coefficients in Fig. 3.4. The curve of $\beta_0 = 1$ and $\beta_1 = 1$ represents the conventional primal-dual algorithm. However, the PEP outage probability cannot be trained to adhere to the expected constraint. If I set β_0 and β_1 at 10^2, the PEP outage probability fails to meet the requirement. Hence, I need to set different values for a more stable training procedure. Concerning the curve of $\beta_0 = 10^2, \beta_1 = 1$,

it takes over 50000 episodes for the PEP outage probability to decrease, significantly reducing the convergence efficiency. In Fig. 3.5, I also exhibit the reward training performance for different normalisation coefficients. We can observe that if β_0 and β_1 are set with the same value, the resource utilisation efficiency cannot be trained to achieve the optimal result. This issue arises because either the primal domain or the dual domain dominates during training, leading to an ineffective Lagrangian multiplier. Only $\beta_0 = 1, \beta_1 = 10^2$ can achieve optimal performance in terms of the PEP outage probability and resource utilisation efficiency with a relatively fast convergence speed. It's worth mentioning that such a parameter tuning issue doesn't exist in the primal CA-TD3. I use the optimal normalised coefficients to train the primal-dual algorithms in the following.

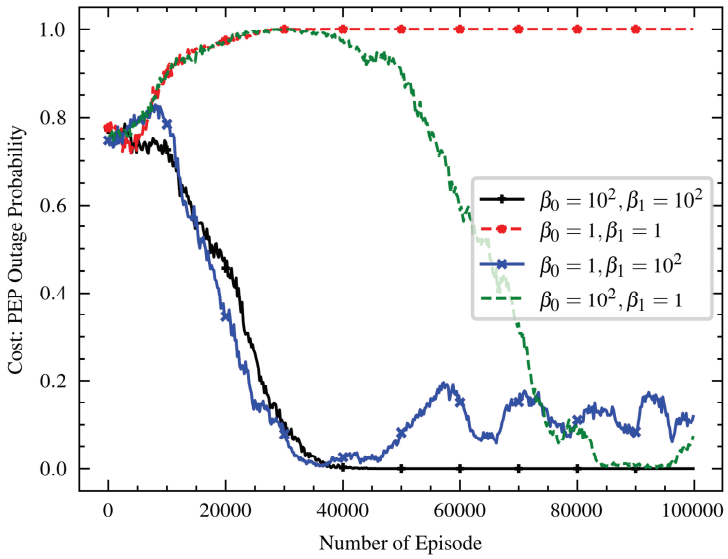


Figure 3.4: PEP outage probability (defined as the cost) against training episodes for the primal-dual CA-TD3.

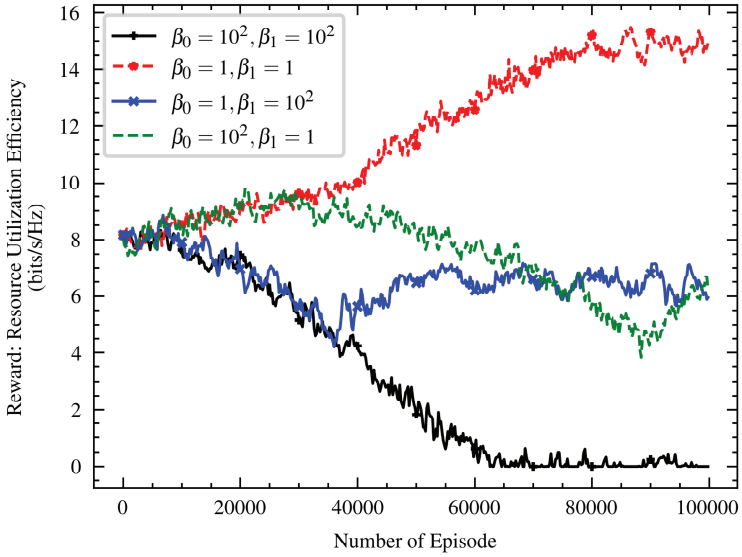


Figure 3.5: Resource utilisation efficiency (defined as the reward) against training episodes for the primal-dual CA-TD3.

Comparison between Primal and Primal-Dual CA-TD3

I compare the primal CA-TD3 with the primal-dual method [74, 80]. In Fig. 3.6, I present the PEP outage probability against training episodes. With the primal CA-TD3 algorithm, the PEP outage probability starts to decrease after 10000 episodes and converges around 30000 episodes for both channel models. However, with the primal-dual CA-TD3, the PEP outage probability continues to increase until over 20000 episodes for the autoregressive channel model and 40000 episodes for the CDL channel model. This occurs because the primal-dual algorithm first optimises the primal variable when the Lagrangian multiplier is small, causing a delay in convergence.

Fig. 3.7 illustrates the resource utilisation efficiency performance. Using the primal CA-TD3 algorithm, the resource utilisation efficiency initially increases for 10000 episodes and then decreases to meet reliability requirements. Convergence is

achieved at 30000 episodes for both channel models. In contrast, with the primal-dual CA-TD3 algorithm, the PEP outage probability converges after 80000 episodes over the CDL channels and 50000 episodes over the autoregressive channels. Although primal and primal-dual CA-TD3 algorithms can obtain similar policies, the primal CA-TD3 algorithm converges much faster than the primal-dual algorithm.

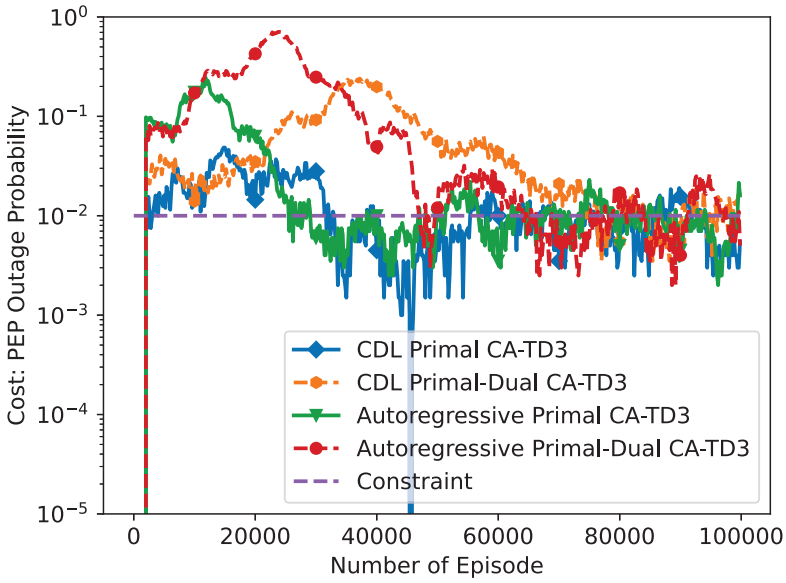


Figure 3.6: PEP outage probability (cost) versus training episodes.

Test Performance Comparison with the Benchmark

In Table 3.2, I test a well-trained policy by using 50 randomly generated continuous autoregressive and CDL channels. Meanwhile, I compared my algorithm with the benchmark in [19], where the number of pilot symbols equals the number of transmission antennas. The results show that my proposed DRL algorithm can achieve more

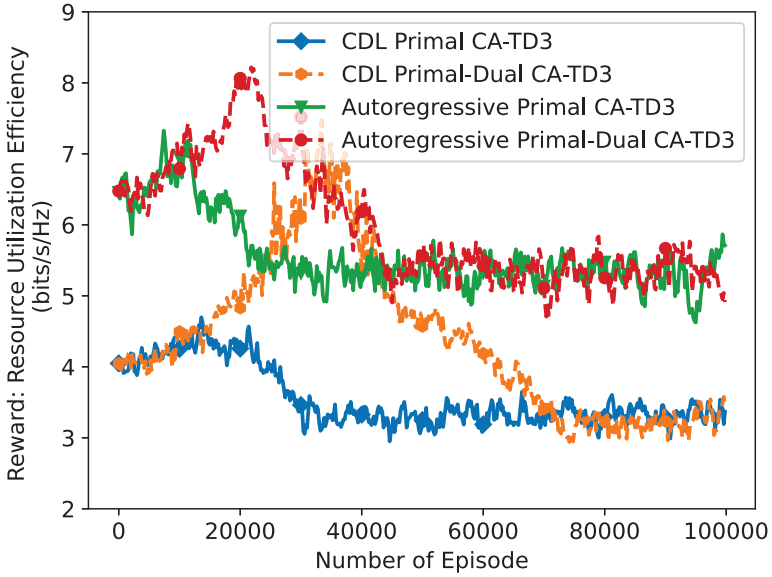


Figure 3.7: Resource utilisation efficiency (reward) versus training episodes.

than a 30% performance gain in resource utilisation efficiency compared to the benchmark. This improvement is because CA-TD3 adjusts the number of pilot symbols according to CSI to maximise resource utilisation efficiency. Additionally, CA-TD3 does not rely on any analytical results or assumptions like those in [19], making it a more practical solution.

There are minor differences between the primal and primal-dual CA-TD3 approaches regarding resource utilisation efficiency and PEP (Packet Error Probability) outage probability. This discrepancy occurs because when the DRL algorithms converge, their rewards and costs fluctuate within a small range, as depicted in Figs. 3.6 and 3.7.

Table 3.2: Test results of resource utilisation efficiency and PEP outage probability with respect to different algorithms.

Performance Metric	Primal CA-TD3		Primal-Dual CA-TD3		Benchmark Primal CA-TD3	
	Autoregressive	CDL	Autoregressive	CDL	Autoregressive	CDL
Resource Utilisation Efficiency (bits/Hz/s)	5.57	3.50	5.22	3.37	4.21	2.63
PEP Outage Probability	0.011	0.006	0.009	0.013	0.006	0.009

3.5 Conclusion

In this chapter, I propose a constrained DRL framework, namely CA-TD3, for MISO-URLLC systems in temporally correlated channels. Specifically, I design a resource allocation policy for channel estimation and data transmission to maximise the resource utilisation efficiency, subject to a PEP outage probability constraint. Considering the imperfect CSI and partial observations of the wireless channel, I formulate this optimisation problem as a POMDP and develop the primal CA-TD3 algorithm to solve the problem. I validate my algorithm on the first-order autoregressive model and the practical CDL model. In my results, the primal CA-TD3 can not only obtain a similar near-optimal solution but also achieve faster convergence. In the future, the proposed constrained DRL algorithm can be applied for data rate, modulation, and coding scheme selection in MISO-URLLC systems.

Chapter 4

Enabling Real-Time Quality-of-Service and Fine-Grained Aggregation for Wireless TSN

Wireless Time-Sensitive Networking (WTSN) is a promising technology for Industrial Internet of Things (IIoT) applications. To meet the latency requirements of WTSN, the wireless local area network (WLAN), such as the IEEE 802.11 protocol with the time division multiple access (TDMA) mechanisms, is shown to be a practical solution. In this chapter, I propose the RT-WiFiQA protocol with two novel schemes to improve the latency and reliability performance: real-time quality of service (RT-QoS) and fine-grained aggregation (FGA) for TDMA-based 802.11 systems. The RT-QoS is designed to guarantee the quality of service requirements of different

traffic and support the FGA mechanism. The FGA mechanism aggregates frames for different stations to reduce the physical layer transmission overhead. The trade-off between reliability and FGA packet size is analysed with numerical results. Specifically, I derive a critical threshold such that the FGA can achieve higher reliability when the aggregated packet size is smaller than the required threshold. Otherwise, the non-aggregation scheme outperforms the FGA scheme. Extensive experiments are conducted on the commercial off-the-shelf 802.11 interfaces. The experiment results show that, compared to the existing TDMA-based 802.11 system, the developed RT-WiFiQA protocol can achieve deterministic bounded real-time latency and significantly improve reliability performance.

4.1 Introduction

With the increasing demand for unmanned devices and automatic control systems, the Industrial Internet of Things (IIoT) has attracted significant attention. It has become one of the most critical aspects of Industry 4.0 [85]. Different from the conventional Internet of Things (IoT) applications, IIoT applications have very stringent requirements in terms of precise synchronisation, transmission reliability, and bounded latency. TSN is a promising solution to meet the stringent requirements of IIoT by utilising the collision-free and low packet error rate (PER) features of wired connections. However, wireless technologies are flexible, scalable, and can be deployed easily and rapidly compared to wired communication solutions. The development of enabling Wireless Time-Sensitive Networking (WTSN) for IIoT has recently attracted much attention [53]. It is challenging for wireless technologies to meet the stringent latency and reliability requirements of critical IIoT applications due to the shared

medium and collision environment of wireless channels.

As one of the most widely applied wireless protocols [86], IEEE 802.11 WiFi systems can achieve high-rate transmissions and potentially meet the stringent latency requirements of IIoT applications by optimising their algorithms and protocols. The existing WiFi technologies adopt carrier sense multiple access with collision avoidance (CSMA/CA) mechanism with distributed random access, which cannot guarantee deterministic latency and reliability. Therefore, some research works focused on the modification of the legacy 802.11 medium access control (MAC) layer towards the WTSN. For example, RT-WiFi [57], Soft-TDMAC [58], and Det-WiFi [59] were proposed based on the time division multiple access (TDMA) protocols and implemented on the IEEE 802.11 commercial off-the-shelf (COTS) network interfaces. However, these designs mainly focused on reducing latency without optimising transmission efficiency and reliability. Moreover, such existing works only consider a single traffic type. How to design an effective system for multiple traffic types to meet their respective quality-of-service (QoS) requirements in IIoT applications remains an open problem.

In this chapter, I propose the real-time WiFi protocol with QoS and aggregation (RT-WiFiQA) by introducing two novel schemes to enhance the performance of the TDMA-based 802.11 systems: real-time quality of service (RT-QoS) and fine-grained aggregation (FGA). I realise that it can be hard to develop a rigid design for a wide range of IIoT applications because different applications may have significantly different requirements of reliability, latency, packet generation rates, etc. Therefore, I develop a flexible and transparent design by creating user application programming interfaces (APIs) for the settings of the proposed RT-WiFiQA protocol. The users

can have great flexibility in choosing their setups in terms of RT-QoS and FGA based on specific application requirements. Furthermore, the design of the RT-WiFiQA protocol is based on the COTS 802.11 interface and is compatible with the existing 802.11 systems with no or minimal modifications.

The proposed RT-QoS scheme can accommodate different traffic types, guarantee their QoS requirements, and support the FGA mechanism. For the system with mixed real-time and non-real-time traffic, the conventional periodic time slot allocation cannot meet the QoS requirements of real-time traffic. RT-QoS can optimise the allocation of time slots based on the distributions of traffic types, their QoS requirements, and available time slot resources. Furthermore, by jointly designing the RT-QoS and application (APP)-layer retransmissions, the reliability of the APP layer can be significantly improved.

The proposed FGA can improve the downlink transmission efficiency by significantly reducing the overhead. I also realise that there is a fundamental trade-off in packet aggregation when considering the WTSN. On the one hand, it reduces the overhead, thus improving the transmission efficiency and allowing more retransmissions. Furthermore, it results in a higher PER for each transmission due to a longer packet length. A natural question arises: *will the packet aggregation scheme benefit the WTSN or not in terms of reliability and latency*? In order to answer this critical question, the trade-off between reliability and latency is analysed, and comprehensive simulations are conducted to validate the impact of the proposed FGA on reliability. According to my analysis, I also give insights on how to choose the aggregation parameters for the design of RT-WiFiQA networks. To the best knowledge of the authors', this is the first paper that studies packet aggregation in WTSN with detailed

implementation and trade-off analysis.

4.2 System Design and Implementation

The architecture of my proposed RT-WiFiQA protocol is shown in Fig. 4.1. The user applications represent a group of concurrent applications with various timeliness, sampling rates, and reliability requirements. I provide APIs for users to allocate an RT-QoS value and determine a specific traffic setting for each packet regarding its QoS requirements. The RT-QoS is designed to guarantee the QoS requirements of real-time traffic and enable the FGA scheme, which is explained in detail in Sec. 4.2.1. The FGA scheme aims to reduce the overhead and improve the downlink transmission reliability and efficiency, which is introduced in detail in Sec. 4.2.2. Besides the two enhancements, I also provide APIs for traffic settings, including the APP-layer retransmission (APP-Re) and the network profile. At last, for the basic TDMA system, I follow the design proposed in [57] for COTS 802.11 interfaces that can achieve a synchronisation accuracy of 20 μs and a time slot duration as low as around $100\mu s$.

4.2.1 APP-Layer Configuration and RT-QoS

Basic MAC-layer retransmissions of the TDMA-based 802.11 system are conducted within one time slot, which is executed if the sender does not receive the acknowledgment (ACK) packet from the receiver. However, in the TDMA-based IIoT system without carrier sense, the MAC-layer retransmissions may fail when burst interference exists. Therefore, I develop APIs for the APP-Re scheme, which can efficiently avoid

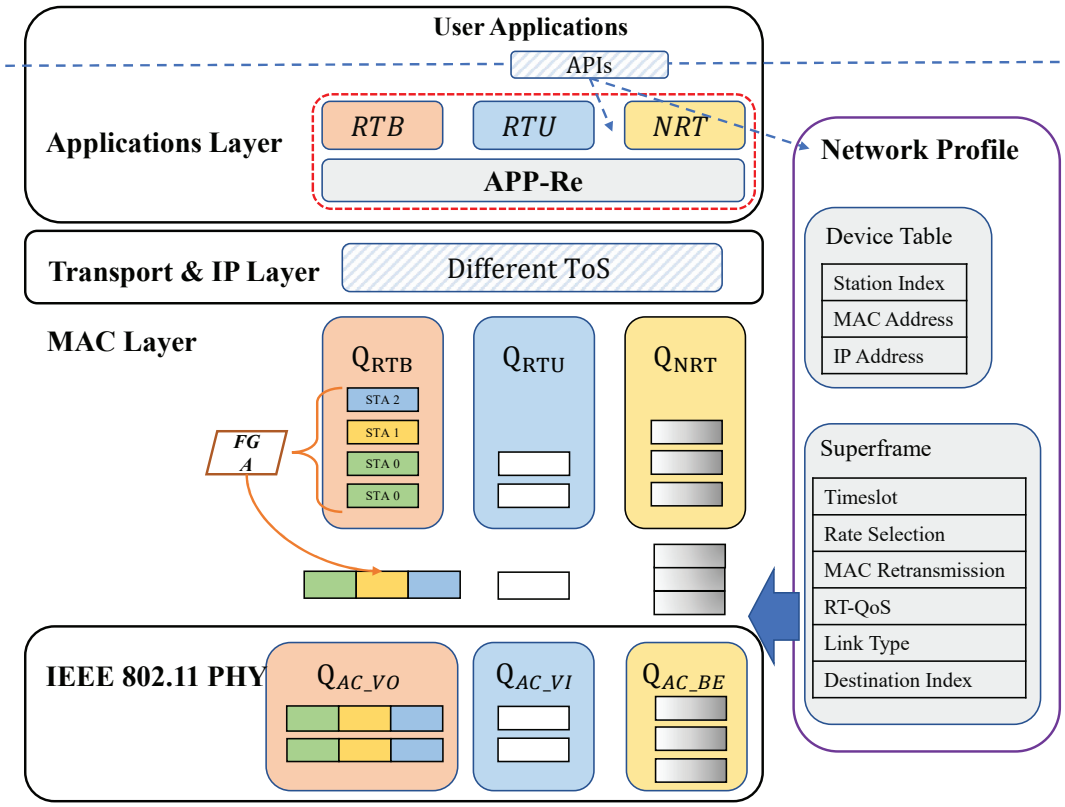


Figure 4.1: Overview of RT-WiFiQA architecture.

burst interference by retransmitting the packets at different time instants. The procedure of APP-Re is addressed as follows. Once senders start a transmission event, each packet is allocated with a unique sequence number. Receivers will reply with an APP-layer ACK to the sender if they receive a new packet and record the sequence number at the same time. Senders stop retransmissions once they receive the ACK. Otherwise, the APP-Re keeps executing until reaching the maximum retransmission times limit. Receivers drop packets with the same recorded sequence. Besides the APP-Re solution, other error control coding techniques can also be added according to the application-specific requirements through my APIs.

The network file is a static file to control the transmission pattern and maintain the network information, including a device table and a superframe structure maintained by the access point (AP) and all stations. The device table contains the MAC address, IP address, and a unique index of each station. The TDMA-based communication pattern is established by a superframe structure, which defines the transmission behaviors in a sequence of consecutive time slots. In RT-WiFiQA, each transmission in the superframe is configured with a link type (i.e., downlink or uplink), an index of the targeted destination, a transmission rate, MAC-layer retransmission times, and an RT-QoS indicator. The maximum MAC-layer retransmission times can be evaluated through the transmission rate, allocated time slots, and packet lengths. The network profile plays the role of the bridge between the APP layer and the MAC layer and can be generated through my APIs. The scheduler on the MAC layer will follow the superframe structure to transmit packets with transmission rates and retransmission times.

The RT-QoS setting is attached to each packet to distinguish different traffic

classes, which have different transmission patterns on the MAC layer. I design APIs for RT-QoS based on the existing type of service (ToS) field in the IP header. Users can determine a ToS value for each frame classified to a specific Access Categories (AC) value on the MAC layer. According to the different AC values, I design a first-in-first-out (FIFO) queue system for each RT-QoS traffic. I create three RT-QoS traffic types as follows:

- *RTB*: Real-time data transmissions through broadcast. The packets stored in the RTB queue can use the FGA mechanism, which aggregates the RTB packets and broadcasts the aggregated packet to multiple desired stations. The RTB queue is particularly suitable for industrial downlink data transmissions using the UDP protocol where ACKs are not required from the MAC layer. In my protocol design, the users can determine whether to use the aggregation or not by enabling the FGA and choosing the RT-QoS through my APIs.

- *RTU*: Real-time transmissions through unicast. Packets stored in the RTU queue will not use the FGA mechanism and are transmitted according to the scheduler by unicasting. The RTU queue is compatible with all the existing upper-layer protocols, e.g., UDP and TCP, for both uplink and downlink. If there is no RTU packet buffered, RTB packets are allowed to be transmitted in unicast within the time slot allocated to RTU.

- *NRT*: Non-real-time transmissions. NRT packets are transmitted only within a temporal window, namely the NRT window (NRTW). The transmission of NRT packets will strictly not exceed NRTW and influence the real-time traffic. The NRT queue is also compatible with all the existing upper-layer protocols

and suitable for packets without latency requirements.

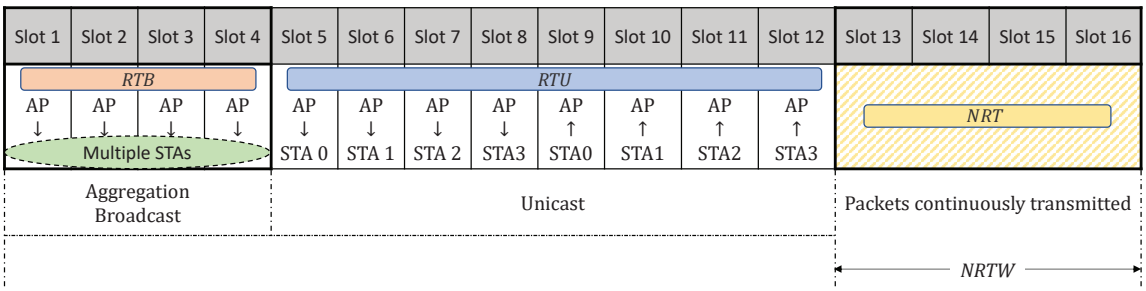


Figure 4.2: An example of RT-WiFiQA superframe design.

Fig. 4.2 is an example of the superframe structure. An RT-QoS indicator is defined for each time slot to determine the packet selection from different queues. The first four time slots are used for RTB packets, which can be transmitted through multiple time slots because of the FGA mechanism. Slots 5 to 12 are reserved for RTU packets, including both uplink and downlink. NRTW for NRT packets is from the 13th slot to the last. In the NRTW, I adopt a best-effort transmission scheme to transmit as many packets as possible under the constraint that the packet transmissions do not exceed the NRTW. In the best-effort scheme, I first fetch a packet from the NRT queue and estimate its transmission duration according to its packet length, transmission rate, and retransmission times. The transmitter can reserve the duration for each packet and trigger the next transmission after this duration. This process keeps executing until the end of the NRTW.

4.2.2 FGA

The FGA process starts with the selection of packets for aggregation. Once the timer triggers a transmission, the scheduler first enables the FGA if the RT-QoS setting of

the current time slot is RTB. Then, the scheduler continues to search for packets in the RTB queue and considers both the packet length and critical threshold of each packet to determine whether a new packet can be transmitted through the FGA mechanism or not. The critical threshold can be evaluated according to Sec. 4.3.1. Specifically, I need to make sure the aggregated packet length is smaller than the critical threshold of the new packet and each selected packet for aggregation. Otherwise, the new packet cannot be transmitted through FGA. The scheduler will first fetch packets for different stations according to the scheduling information of the superframe. When the scheduler cannot find such packets from the RTB queue and the aggregated packet length is smaller than the critical threshold, it will seek other existing UDP packets from all RT-QoS queues for aggregation, following the sequence of RTB, RTU, and NRT, until reaching the critical length constraint. This is because my FGA scheme is compatible with all UDP packets. Additionally, if no packet can be aggregated to the head-of-line packet, the scheduler will transmit the packet without FGA by using one time slot. In the next time slot, the scheduler continues to fetch the new head-of-line packet in the RTB queue and search for other packets that can be aggregated. This procedure keeps executing until the end of the allocated time slots for the RTB transmission.

After selecting FGA packets, the scheduler will initially push these packets into a temporal singly linked list. The first frame in the linked list will keep slices of the MAC header and frame check sequence (FCS). A one-byte hexadecimal aggregation flag is appended after the MAC header to recognise the aggregated packet. Then, a unique station flag will be generated for each frame, including the station identity recorded in the device table and a frame length. The scheduler further appends

the combinations of one station flag and its corresponding full original frame after the aggregation flag. Finally, the destination address at the MAC header of the aggregated frame is changed to the broadcast MAC address, e.g., 0xFF. An example of the final aggregated frame format is shown in Fig. 4.3.

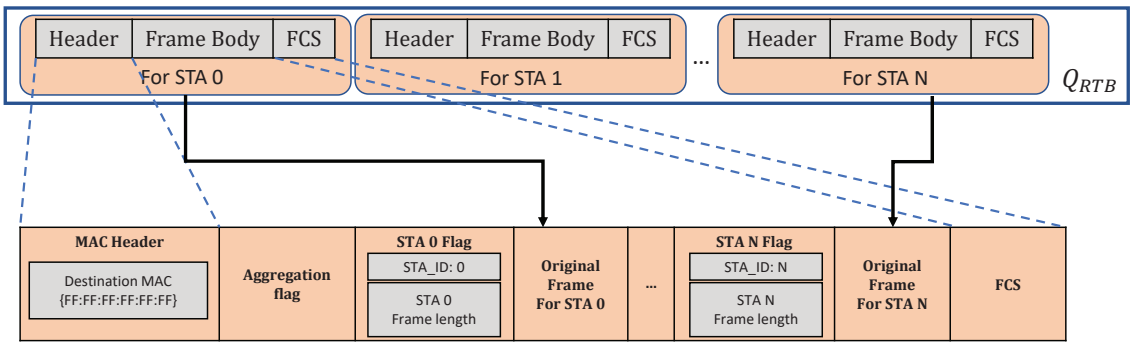


Figure 4.3: Typical FGA frame format.

The disaggregation process happens when a station receives the aggregated frame. The station determines whether the packet is sent through the aggregation process based on two conditions: 1) The destination MAC address is a broadcast address; 2) The octet after the MAC header is the special aggregation flag. Then, the receiver reads the aggregated frame from the aggregation flag to the end of the frame. Once the station identity in the station flag is matched with its own identity, the station will keep the following frame with the frame length recorded in the station flag, and the rest parts of the aggregated frame will be discarded. If the station identity is not matched, the station will skip the frame length recorded in the station flag and read the next station flag until the end pointer of the aggregated frame.

4.3 FGA Analysis and Numerical Results

With a given time budget for transmissions, i.e., the number of allocated time slots, aggregation of multiple packets can potentially retransmit more times than the conventional non-aggregated transmission. It is because aggregation reduces the overhead of the PHY layer significantly, especially when the payload size of the packets is relatively small. However, aggregation leads to a larger packet size, which may increase the PER of the transmitted packet compared with the non-aggregated transmission. Therefore, it is important to determine whether the aggregation mechanism will benefit the system in terms of reliability and latency or not. Besides, how shall I choose the system parameters for the packet length of the aggregated packet in order to achieve higher reliability with the bounded latency requirement? To answer these critical questions, in the following, I compare the reliability performance of aggregation and non-aggregation schemes under a predefined latency constraint, i.e., a given number of allocated time slots.

4.3.1 Trade-off Analysis of FGA

To analyse the trade-off between the reliability and latency of applying the FGA scheme, I assume that a total number of n packets can be aggregated for transmission. For a fair comparison, each packet without aggregation will occupy one time slot according to the existing non-aggregation mechanisms. Differently, the aggregated n packets can utilise n time slots for transmission such that the total time consumption for the non-aggregation and aggregation schemes are the same. Note that the latency constraint considered in this case is n time slots for all the packets. It is reasonable because the slot duration of the TDMA-based system is very small, e.g., in the level

of μs, and users can choose the setup of n depending on the specific application requirements.

Let $l_i, 1 \leq i \leq n$ denote the length of i^{th} packet, $l_{a,i}$ denote the total length of packets that can be aggregated to the i^{th} packet. T_s denotes the duration of one time slot, and r denotes the transmission rate. I define T_i as the transmission time of the i^{th} packet without using FGA, and $T_{a,i}$ as the transmission time of the i^{th} packet adopting FGA with a total packet length of $l_i + l_{a,i}$.

Based on l_i and $l_{a,i}$, I can derive that

$$T_i = \frac{l_i}{r} + T_{PLCP} + T_{DIFS}, \tag{4.3.1a}$$

$$T_{a,i} = \frac{l_{a,i} + l_i}{r} + T_{PLCP} + T_{DIFS}, \tag{4.3.1b}$$

where $T_{PLCP} = 20\mu s$ is the physical layer convergence procedure (PLCP) preamble with header delay of each packet transmitted by the IEEE 802.11 PHY layer [87]; $T_{DIFS} = 28\mu s$ is the inter-frame spacing. I now define the maximum transmission times for the i^{th} packet without using FGA as $\mathcal{M}_i$, and that for the i^{th} packet adopting FGA as $\mathcal{M}_{a,i}$. They can be calculated as

$$\mathcal{M}_i = \frac{T_s - T_g}{T_i}, \tag{4.3.2a}$$

$$\mathcal{M}_{a,i} = \frac{nT_s - T_g}{T_{a,i}}, \tag{4.3.2b}$$

where $T_g = 20\mu s$ is the guard time to tolerate the synchronisation error.

To verify whether my FGA scheme can improve reliability or not, I compare the PER performance of the FGA with the conventional non-aggregation scheme. The PER is defined as the probability that a packet cannot be successfully transmitted

after all transmissions and retransmissions within the same amount of allocated time slots. To capture the PHY-layer overhead, I model the PLCP preamble data length as $l_o = rT_{PLCP}$ because the PLCP preamble is transmitted at $1Mbps$ [88]. Based on the above model, I define P_i as the PER of the i^{th} packet without using FGA, and $P_{a,i}$ as the PER of the i^{th} packet adopting FGA. Let p denote the bit error rate [1] (BER), P_i and $P_{a,i}$ can be evaluated by

$$P_i = (1 - (1-p)^{l_i + l_o})^{\mathcal{M}_i}, \tag{4.3.3a}$$

$$P_{a,i} = (1 - (1-p)^{l_i + l_{a,i} + l_o})^{\mathcal{M}_{a,i}}. \tag{4.3.3b}$$

To compare the PER of non-aggregation and aggregation, in the following, I will mathematically derive the solution for $P_{a,i} \leq P_i$, such that the FGA scheme can outperform the non-aggregation transmission. Because $P_i, P_{a,i} > 0$, I first take the logarithm on both sides of the inequality. I then define $f(l_{a,i}) = \ln(P_{a,i}) - \ln(P_i)$. The inequality $P_{a,i} \leq P_i$ is thus equivalent to

$$f(l_{a,i}) = \ln(P_{a,i}) - \ln(P_i) \leq 0. \tag{4.3.4}$$

The solution to the inequality can be summarised and given in Proposition.4.3.1.

Proposition 4.3.1. *There exists a unique solution $l^*_{a,i}$ to Eq. (4.3.4), such that when $l_{a,i} \leq l^*_{a,i}$, $P_{a,i} \leq P_i$, and the FGA scheme outperforms non-aggregation in terms of reliability. Otherwise, $P_{a,i} > P_i$, and the FGA scheme has a higher PER.*

Proof. I first prove that function $f(l_{a,i})$ is a monotonically increasing function of $l_{a,i}$, then prove there exists a unique solution for Eq. (4.3.4). Let $T_o = T_{PLCP} + T_{DIFS}$

[1] The value of p can be evaluated approximately by the long-term average PER at each station using Eq. (4.3.3), which can be acquired in an offline manner.

and Eq. (4.3.4) can be simplified to

$$f(l_{a,i}) = \frac{nT_s - T_g}{\frac{l_i+l_{a,i}}{r} + T_o} \ln(1-(1-p)^{l_i+l_o+l_{a,i}}) - \frac{T_s - T_g}{\frac{l_i}{r} + T_o} \ln(1-(1-p)^{l_i+l_o}). \tag{4.3.5}$$

To prove the monotonicity, the first order derivative of Eq. (4.3.5) with respect to $l_{a,i}$ can be evaluated by

$$\frac{\mathrm{d}f}{\mathrm{d}l_{a,i}} = -(nT_s - T_g)\left(\frac{r\ln\left(1-(1-p)^{l_i+l_o+l_{a,i}}\right)}{(l_i+l_{a,i}+rT_o)^2} + \frac{r(1-p)^{l_i+l_o+l_{a,i}}\ln(1-p)}{\left(1-(1-p)^{l_i+l_o+l_{a,i}}\right)(l_i+l_{a,i}+rT_o)}\right) > 0. \tag{4.3.6}$$

Because $0 < p < 1$, it can be readily verified that the above inequality holds. Therefore, Eq. (4.3.5) is monotonically increasing with $l_{a,i}$. I now prove there exists a unique solution of Eq. (4.3.5). On one hand, because $l_{a,i} \geq 0$, I have

$$f(0) = \frac{(n-1)T_s}{\frac{l_i}{r} + T_o} \ln(1-(1-p)^{l_i+l_o}). \tag{4.3.7}$$

Due to $n \geq 1$, it can be verified that $f(0) \leq 0$. On the other hand, if $l_{a,i}$ approach infinity, I have

$$f(+\infty) = -\frac{T_s - T_g}{\frac{l_i}{r} + T_o} \ln(1-(1-p)^{l_i+l_o}) > 0 \tag{4.3.8}$$

With the monotonicity of $f(l_{a,i})$, I can deduce that there must exist a unique solution $l^*_{a,i}$ such that $f(l^*_{a,i}) = 0$. This completes the proof. □

Based on the Proposition 4.3.1, I can obtain the critical aggregated packet length $l^*_{a,i}$ for each packet length l_i by solving $f(l_{a,i}) = 0$. Due to the complicated structure

of $f(l_{a,i})$, it is intractable to obtain a closed-form expression of $l^*_{a,i}$. Fortunately, $l^*_{a,i}$ can be solved through numerical methods such as the bisection method. To determine whether the i^{th} packet can be aggregated or not, packets from the first to the i^{th} need to meet their individual requirement of the critical length threshold. Specifically, the i^{th} packet can be aggregated only if for $\forall j \in [1, i]$, $l_{a,j} \leq l^*_{a,j}$. Based on the above analysis, I can then determine whether a packet should be transmitted through the FGA scheme or unicast to achieve optimal reliability in the RT-WiFiQA system.

4.3.2 Numerical Results

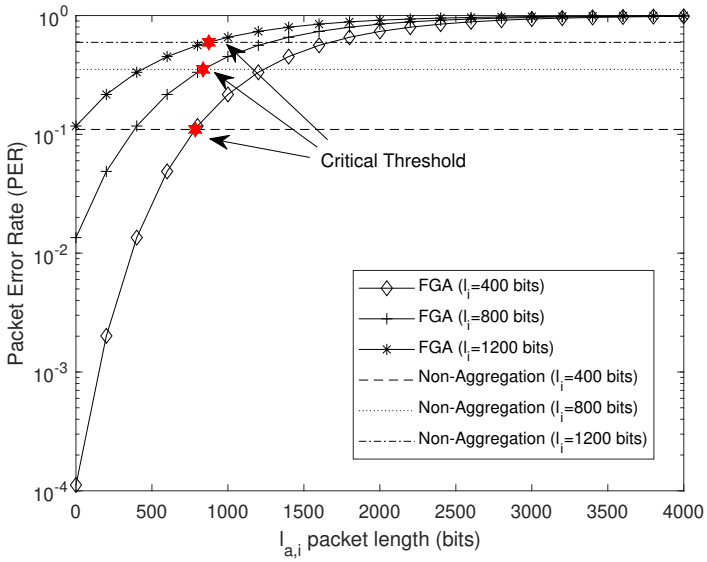


Figure 4.4: PER against aggregated packet size $l_{a,i}$ for different l_i, where the critical thresholds can be evaluated based on Proposition 4.3.1.

I consider a setup with four time slots allocated for RTB transmissions with a time slot of 512 μs, and a BER of 1.3×10^{-3} for the link, which is comparable to my practical setup in Sec. 4.4.1. In Fig. 4.4, I depict the PER against $l_{a,i}$ for different l_i

and compare the performance of FGA with the non-aggregation scheme. The PER of the non-aggregation scheme does not change with $l_{a,i}$ and is shown as a horizontal line, while the curves of FGA are increasing as $l_{a,i}$ grows because the PER of FGA is monotonically increasing with $l_{a,i}$ as discussed in Sec. 4.3.1. I also depict the critical threshold of the FGA scheme according to Proposition 4.3.1 by using a bisection method. From Fig. 4.4, I can observe that each FGA curve and the non-aggregation line have a unique intersection point, which coincides with my theoretical threshold. If $l_{a,i}$ is smaller than the critical threshold, the FGA scheme can have a lower PER than the non-aggregation scheme and vice versa. It validates my analysis provided in Sec. 4.3.1.

I then show the PER against different BER p in Fig. 4.5 and set l_i as 800 bits. In Fig. 4.5, I can first observe that the PER curves grow as BER increases. If the BER is relatively low, the FGA scheme can achieve a lower PER than the non-aggregation scheme for a large variety of $l_{a,i}$. Otherwise, the FGA scheme can achieve a lower PER only for a small $l_{a,i}$. This observation indicates that packets with a large size can be aggregated when the channel condition has a mild BER. It is because a good channel condition can achieve a low PER even when the packet size is large.

4.4 Experiments and Results

My experiment design and performance evaluation are presented in this section. In my experimental platform, I use miniPCs from Qotom [89] for AP and stations. The miniPCs are running Ubuntu 14.04 operating system, and the Linux kernel version is 3.13.0-32. The CPU used for AP is Intel Core i5-4200U, while the one for stations is Intel Core i3-5005U. The RAM of all devices is 8G. For the IEEE 802.11 interface, I

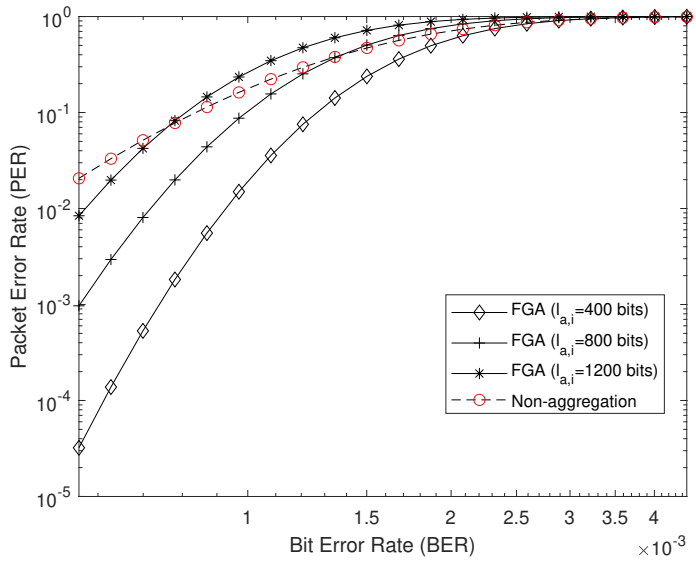


Figure 4.5: PER against increasing BER for different total length $l_{a,i}$ of packets that can be aggregated to the i^{th} packet with l_i of 800 bits.

choose the Atheros NIC AR9285, which supports IEEE 802.11 b/g/n protocols and uses open-source driver ATH9k [90]. Besides one AP and four stations, an additional PC is used to monitor and evaluate the performance of all devices. Fig. 4.6 presents my experimental platform, which is comparable to the practical IIoT network. In order to obtain practical results in a real channel environment, I have implemented the proposed schemes on my platform using 802.11 b/g/n PHY layer due to the available open-source driver. Nevertheless, my protocol design can also be extended to more advanced IEEE 802.11 ac/ax with orthogonal frequency-division multiple access (OFDMA) PHY layer by further improving the time-domain transmission efficiency and reliability within a given number of resource units, which will be left as future work.

Figure 4.6: Experimental environment, which is designed to simulate a real industrial wireless setting. One AP and four stations are placed on the ground for testing.

4.4.1 Experiment Design

I measure the performance on both the MAC layer and the APP layer, and the performance metrics include latency and reliability. The MAC-layer results aim to validate the bounded latency and optimised reliability performance of the proposed RT-WiFiQA protocol compared to WiFi and RT-WiFi. The APP-layer results can present the overall transmission latency and reliability of IIoT applications because packets are forwarded to the APP layer as a destination. Specifically, the MAC-layer latency is the time difference between one packet leaving the transmitter's MAC layer and entering the receiver's MAC layer. The MAC-layer reliability is measured by the ratio of successfully received packets to the total number of transmitted packets on the MAC layer. The APP-layer latency is the time difference between one packet generated by the transmitter application and successfully received by the receiver

application. The APP-layer reliability is defined as the ratio of successfully received packets by the receiver application to the total number of packets transmitted from the transmitter application. For the measurement of latency in the MAC and APP layers, the synchronisation between the MAC layer is realised by the timing synchronisation function (TSF) of Linux [57] with a drift lower than 20 μs. The synchronisation between the applications of transmitter and receiver is achieved by IEEE 1588 Precision Time Protocol (PTP) [91] with its software tool PTP daemon (ptpd) [92]. The APP-layer synchronisation error is smaller than 40 μs. The synchronisation error is acceptable for my delay measurement where the MAC-layer delay is around 300 μs, and the APP-layer delay is on the level of ms.

My experiments focus on the downlink performance of my proposed mechanisms. In the following experiments, I compare the MAC-layer and the APP-layer performance of RT-WiFiQA with RT-WiFi, and conventional WiFi. RT-WiFi is a basic TDMA system based on 802.11 interfaces without considering the proposed RT-QoS and FGA mechanisms. I develop applications in Python 3 to simulate downlink traffic with different RT-QoS types as well as uplink traffic. The different programs can reveal the concurrent running state of multiple practical applications with various workloads and QoS requirements.

All the applications generate packets of different types with a length of 50 bytes every 20 ms, which are represented as the traffic payload and packet interval in Table 4.1. I execute a combination of the three applications for each station concurrently, and each experiment lasts 40 minutes. Note that I add uplink traffic to my experiments, but the uplink performance of the proposed RT-WiFiQA is similar to the result of RT-WiFi, which was extensively investigated in [57]. I also measure the

approximate average BER of the four stations by measuring the long-term average PER using Eq. (4.3.3). According to the offline measurement, I set $p = 1.3 \times 10^{-3}$ for all the considered stations, which is used to determine the critical thresholds for each station in my proposed FGA algorithm. A time slot duration of 512 μs and a PHY-layer transmission rate of 36 Mbps are set to both RT-WiFi and RT-WiFiQA. The MAC-layer retransmission times are pre-calculated according to Eq. (4.3.2), which are 4 for transmissions without the FGA and 10 for the FGA packets. The APP-Re is set to 4 times in terms of the COTS configuration.

Table 4.1: Experiment parameters preset for RT-WiFiQA.

Parameters	Values
Number of stations	4
MAC retransmissions	10
Traffic payload	50 Bytes
Packet interval	20 ms
Transmit rate	36 $Mbps$
Time slot	512 μs
Test duration per group	40 min

4.4.2 MAC-Layer Performance

I define the deadline as the required latency performance of a given application and the effective packet loss ratio (EPLR) as the percentage of packets unsuccessfully received at or exceeding the given deadline. I use complementary cumulative distribution function (CCDF) curves in Fig. 4.7 and 4.8 to present the EPLR performance of all

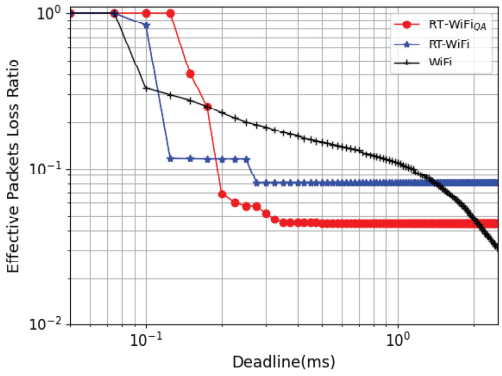
the stations, which can indicate the trade-off between latency and reliability. I first plot the MAC-layer EPLR performance in Fig. 4.7a to 4.7d. The curves of RT-WiFi and WiFi drop earlier than RT-WiFiQA, illustrating that the minimum achievable delay of RT-WiFi and WiFi is lower than RT-WiFiQA. It is because the proposed FGA mechanism leads to a larger packet size such that the FGA packets cannot be transmitted within an extremely small amount of time.

The downward tendency of RT-WiFiQA and RT-WiFi is concentrated upon the mean delay and follows a step case. Differently, the curve of WiFi shows a gradual downward trend. It is because the WiFi system uses a dynamic rate control algorithm, i.e., Minstrel [93], but RT-WiFi and RT-WiFiQA choose a fixed rate and retransmission times setting. This observation also presents the proposed RT-WiFiQA and RT-WiFi can guarantee bounded latency. Moreover, RT-WiFiQA achieves lower EPLR than RT-WiFi. It is because the proposed FGA scheme can potentially increase the reliability of the system by reducing the transmission overhead and allowing more retransmission times. At last, WiFi can outperform RT-WiFi and RT-WiFiQA when the delay requirement is very high, e.g., more than 2 ms for STA0. It is because WiFi uses the CSMA/CA scheme and has a larger number of retransmissions. Differently, RT-WiFi and RT-WiFiQA apply limited retransmission times in order to achieve bounded latency.

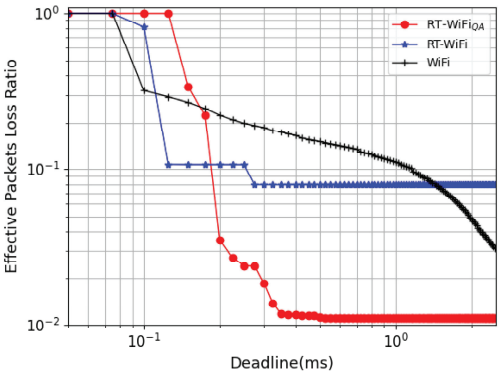
For a quantified analysis, I define deadline as the required latency performance of a given application and effective reliability as the percentage of successfully received packets within the deadline. Table 4.2 provides the effective reliability result against different MAC-layer deadlines. Because all stations show a similar process of data exchange, I take STA0 as an example. In terms of 100 μs deadline, the effective

reliability results for RT-WiFiQA, RT-WiFi, and WiFi are 0, 16.24%, and 67.26%, respectively. Similarly, it is because RT-WiFiQA adopts the FGA scheme, and the transmission time of FGA packets is longer than 100 μs. The effective reliability result remains 95.57% after 1000 μs for RT-WiFiQA and 91.95% after 500 μs for RT-WiFi. The improvement in achievable reliability is due to the deployment of the FGA scheme.

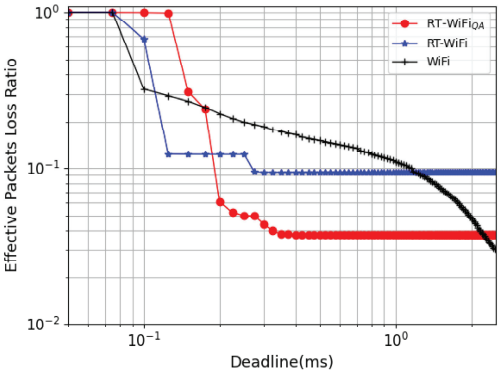
Figure 4.7: MAC-layer EPLR CCDF curves of RT-WiFiQA, RT-WiFi and WiFi.



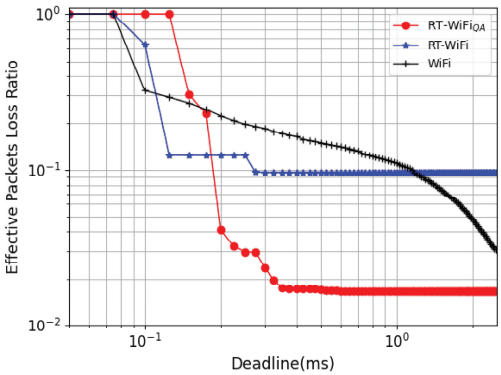
(a) STA0 MAC-layer EPLR



(b) STA1 MAC-layer EPLR



(c) STA2 MAC-layer EPLR



(d) STA3 MAC-layer EPLR

Table 4.2: Effective reliability with a MAC delay lower than a specific deadline value for four stations

Timeout	STA0(%)			STA1(%)		
(ms)	RT-WiFiQA	RT-WiFi	WiFi	RT-WiFiQA	RT-WiFi	WiFi
0.1	0	16.24	67.26	0	18.73	67.88
0.3	94.82	91.94	81.45	98.15	92.06	81.36
0.5	95.55	91.95	85.14	98.87	92.07	85.02
1	95.57	91.95	89.07	98.89	92.07	88.94
1.5	95.57	91.95	92.5	98.89	92.07	92.41
2	95.57	91.95	95.19	98.89	92.07	95.18
2.5	95.57	91.95	96.96	98.89	92.07	96.94
Timeout	STA2(%)			STA3(%)		
(ms)	RT-WiFiQA	RT-WiFi	WiFi	RT-WiFiQA	RT-WiFi	WiFi
0.1	0	32.78	67.65	0	35.99	67.89
0.3	95.61	90.55	81.41	97.64	90.31	81.59
0.5	96.22	90.56	85.02	98.3	90.33	85.08
1	96.23	90.56	88.99	98.32	90.33	88.98
1.5	96.23	90.56	92.49	98.32	90.33	92.49
2	96.23	90.57	95.23	98.32	90.33	95.19
2.5	96.23	90.57	97.03	98.32	90.33	96.99

4.4.3 APP-Layer Performance

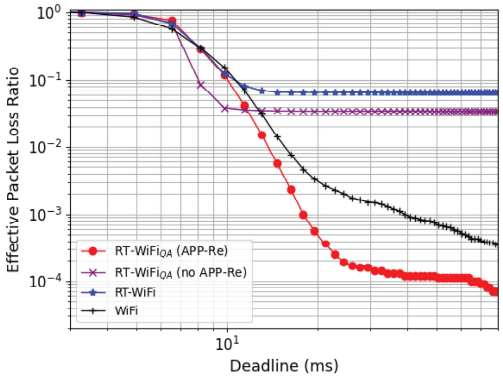
I now turn to the APP-layer EPLR performance and show the CCDF curves in Fig. 4.8a to 4.8d. For a more comprehensive comparison, I also add a benchmark curve of RT-WiFiQA without the APP-Re.

Taking STA1 as an instance, the curve of WiFi starts to drop firstly and keeps a gradual downward tendency as the delay grows and outperforms other systems when the delay is relatively small, e.g., from 5 ms to 10 ms. It is also because WiFi uses the Minstrel rate control algorithm, which may select higher rates than the fixed rate used in the TDMA systems. RT-WiFiQA without APP-Re performs best as the delay requirement is from 10ms to 15ms. Compared to RT-WiFi, RT-WiFiQA with the RT-QoS scheme can transmit real-time packets without internal interference from non-real-time packets, leading to smaller APP-layer latency. Compared to WiFi, due to the coexistence of both uplink and NRT traffic, the CSMA/CA mechanism of WiFi leads to a longer back-off delay. When the delay requirement is over 12 ms, RT-WiFiQA without APP-Re and RT-WiFi almost reach the bound of their reliability. RT-WiFiQA with APP-Re performs the best because the APP-Re can effectively combat burst interference. At the delay of 40 ms, RT-WiFiQA can ultimately achieve an EPLR of 10^{-4}. The achievable latency and reliability on the APP layer can benefit many existing IIoT applications, such as the wireless control of Automated Guided Vehicles (AGVs) for logistic sorting [66, 94], and the interlocking control systems in process automation domain [95].

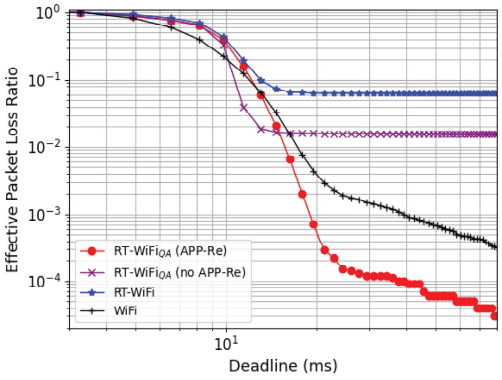
Table 4.3 provides the result of effective reliability in terms of deadline values on the APP layer. Taking STA1 as an example, at the deadline of 15 ms, RT-WiFiQA without APP-Re achieves the highest effective reliability at 98.38% because of the

RT-QoS scheme, which can guarantee the performance of real-time packets. At the deadline of 20 ms, RT-WiFiQA with APP-Re outperforms other systems with effective reliability of 99.94%, better than 99.61% of WiFi because the APP-Re scheme can combat burst interference and maintain the real-time performance at the same time. In terms of the four stations, RT-WiFiQA can achieve average reliability of 99.99%.

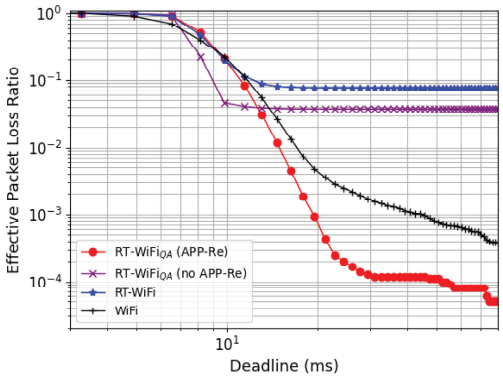
Figure 4.8: APP-layer EPLR CCDF curves of RT-WiFiQA, RT-WiFi and WiFi.



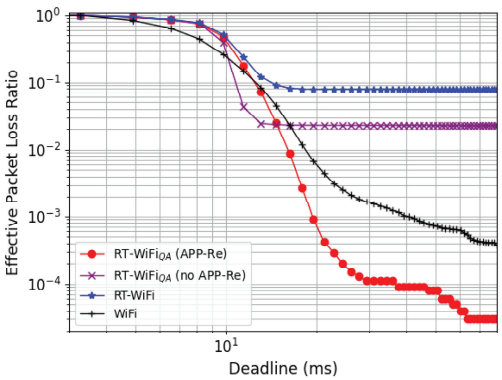
(a) STA0 APP-layer EPLR



(b) STA1 APP-layer EPLR



(c) STA2 APP-layer EPLR



(d) STA3 APP-layer EPLR

Table 4.3: Effective reliability with an APP delay lower than a specific deadline value for four stations

Timeout (ms)	STA0(%)				STA1(%)			
	RT-WiFiQA (APP-Re)	RT-WiFi-QA (no APP-Re)	RT-WiFi	WiFi	RT-WiFiQA (APP-Re)	RT-WiFi-QA (no APP-Re)	RT-WiFi	WiFi
5	7.34	9.04	4.12	15.79	11.3	12.85	7.88	18.97
8	68.78	89.15	66.9	68.19	34.97	32.92	28.55	58.42
10	89.39	96.18	88.56	86.04	63.98	72.25	60.21	79.2
15	99.53	96.51	93.39	98.75	98.34	98.38	92.83	97.13
20	99.95	96.53	93.49	99.69	99.94	98.43	93.58	99.61
40	99.99	96.53	93.51	99.91	99.99	98.43	93.61	99.91
80	99.99	96.53	93.51	99.97	99.99	98.43	93.61	99.97
Timeout (ms)	STA2(%)				STA3(%)			
	RT-WiFiQA (APP-Re)	RT-WiFi-QA (no APP-Re)	RT-WiFi	WiFi	RT-WiFiQA (APP-Re)	RT-WiFi-QA (no APP-Re)	RT-WiFi	WiFi
5	2.24	1.27	2.14	10.29	6.62	6.82	6.07	16.97
8	41.88	70.87	48.56	57.68	23.68	22.8	20.91	53.76
10	81.34	95.52	81.96	79.41	56.46	67.43	52.55	75.36
15	99.02	96.23	92.06	97.69	97.96	97.69	91.14	96.05
20	99.92	96.27	92.34	99.57	99.93	97.74	92.14	99.4
40	99.99	96.27	92.37	99.89	99.99	97.74	92.17	99.9
80	99.99	96.27	92.37	99.97	99.99	97.74	92.17	99.96

4.5 Conclusions and Future Work

In this chapter, I develop the RT-WiFiQA protocol with two novel schemes, i.e., RT-QoS and FGA, for IIoT applications based on 802.11 TDMA systems. The RT-QoS protocol is used to guarantee the latency and reliability performance of real-time traffic when multiple types of traffic coexist and support the proposed FGA mechanism. The FGA mechanism can aggregate multiple packets for different stations and reduce the transmission overhead to improve the efficiency and reliability of the system. I aim at a flexible design by developing APIs for the configuration of RT-WiFiQA and providing insights on network parameter selection. Based on the observation of the trade-off between the FGA packet size and reliability, I analytically derive a critical threshold such that the FGA scheme can outperform non-aggregation in terms of reliability when the aggregated packet size is smaller than the critical threshold and provide numerical results. I also implement the proposed RT-WiFiQA protocol on the COTS hardware running the Linux system and conduct extensive experiments to compare the performance of RT-WiFiQA with RT-WiFi and conventional WiFi. The experiment results demonstrate that RT-WiFiQA can promise higher reliability than RT-WiFi and guarantee a real-time performance compared to WiFi on both the MAC and APP layers.

Despite the fact that my proposed RT-WiFiQA protocol can improve the latency and reliability performance compared with the RT-WiFi and legacy WiFi, it still has some limitations which need to be addressed in the future. Firstly, the achievable reliability of the designed RT-WiFiQA protocol is confined by my designed rate control mechanism, where a fixed rate is adopted for each packet transmission. To further improve the performance, in my future work, I will develop advanced rate control

mechanisms by using machine learning algorithms for the proposed RT-WiFiQA protocol that can select the rate adaptively according to the dynamics of the channel environment. Secondly, the current RT-WiFiQA protocol is designed and implemented on 802.11 b/g/n COTS chips where an OFDM physical layer is adopted. Due to the hardware limitations, I are not able to extend them to the recent OFDMA 802.11 systems, e.g., 802.11ax. It is important to redesign the proposed protocol for OFDMA systems and evaluate its performance on OFDMA systems, which will be left as my future work.

There are multiple interesting topics to be explored in my future work. Firstly, I will optimise the FGA scheme on the more recent 802.11ax and 802.11be interfaces. My current FGA scheme focuses on resource allocation in the time domain for 802.11 b/g/n. Note that in the OFDMA systems, the resource allocation needs to consider both the time-domain and frequency-domain resources by allocating the resource units to different devices. In this way, the FGA algorithm needs to be further optimised for more efficient transmissions, and the trade-off analysis provided in this chapter needs to be revisited. Moreover, to combat random burst interference and improve reliability performance, adaptive rate control mechanisms need to be developed to deal with the dynamics of the wireless environment. Other critical features such as throughput, energy efficiency, and security should be considered in future work.

Chapter 5

Conclusions and Future Work

In this thesis, I proposed multiple novel resource allocation strategies for URLLC systems and developed an experimental platform for low-latency services. In this chapter, I primarily summarise the contributions and results. Moreover, I discuss some potential future work.

5.1 Summary of Results

In Chapter 2, I proposed unsupervised learning algorithms for resource allocation in URLLC systems by optimizing resource utilization efficiency. I first investigated several challenges in practical URLLC systems, including limited channel resources, random channel realizations, channel estimation errors, and beam training errors. I considered a practical scenario where the BS or UE only has imperfect CSI. Due to the existence of many unknown variables, no closed-form solutions could be derived for the optimal resource allocation policy. To overcome these practical challenges, I presented data-driven unsupervised learning algorithms to estimate the optimal resource allocation policy. Based on the availability of PEP and SINR, I proposed model-based and model-free algorithms. I also abided by the 5G NR standard to

design the communication model, where I considered CSI-RS for beam training, DM-RS for channel estimation, and codebook-based precoding techniques. The numerical results show that although the model-free algorithm requires less information than the model-based method, it can still obtain a near-optimal policy. Moreover, the unsupervised learning algorithm could fully explore the features of various channel realizations and can outperform the existing benchmark. The reason is that the SINR of the benchmark has a more extended tail distribution than our methods.

In Chapter 3, I exploited the resource allocation strategies for the correlated channel in URLLC systems. Due to the short time scale of URLLC transmissions, the channel correlation could not be neglected. Considering the practical imperfect CSI scenario, I aimed to seek a resource allocation policy that could lead to optimal resource utilization efficiency within the correlated time while controlling the PEP under the constraint. Therefore, I proposed the data-driven DRL-based algorithms: primal CA-TD3 and primal-dual CA-TD3. Since the conventional primal-dual algorithm was shown to be unstable and slow during training, I set an extra normalization coefficient. I validated my algorithms on the first-order autoregressive channel model and the CDL channel. In the numerical results, I first presented the challenge of parameter tuning in the primal-dual method and showed the effectiveness of this enhanced scheme in the simulation. I also showed that both algorithms could achieve near-optimal solutions. However, the primal CA-TD3 could easily tune the parameters and achieve faster convergence than the enhanced primal-dual solution. Moreover, my trained models can be used in correlated channel realizations and outperform the existing benchmark.

In Chapter 4, I focused on the hardware-based experimental platform of WTSN

for factory automation and the manufacturing industry, which is one of the most significant scopes of URLLC systems. The testbed is developed based on 802.11 COTS open-source hardware. I initially enabled the TDMA scheduling to guarantee a deterministic transmission pattern, then enhanced the platform with FGA and RT-QoS schemes. I theoretically derived the critical point where the FGA can lead to optimal performance in terms of reliability and latency. The experimental results also confirm that my platform can outperform the existing TDMA-based 802.11 platform and legacy 802.11 system.

5.2 Future Work

URLLC remains an important application and an open challenge in future 6G communication systems, which have more stringent latency and reliability requirements [2, 96, 97]. The improvement of URLLC, namely extreme URLLC (xURLLC), will promote the development of emerging critical applications, such as massive IIoT communications and autonomous robotics. I present the following potential future works.

In Chapter 2, I proposed algorithms for the MISO-URLLC system. For future xURLLC systems, the communication architecture will be more complex than the considered scenario. First, massive MIMO, which is the communication system where the BS has a large number of antennas and serves a massive number of UEs, is an extension direction of my current research, as the massive MIMO can further improve spatial diversity and efficiency. How to intelligently execute resource allocation for massive MIMO systems with limited resource blocks remains a significant challenge. Second, the proposed solutions only considered the resource allocation for URLLC. It is also worthwhile to investigate how to optimise resource utilisation efficiency when

emBB and URLLC coexist.

In Chapter 3, I utilise DRL frameworks to obtain the resource allocation policy for URLLC systems under correlated channels. I consider the autoregressive and the CDL channel model to validate the feasibility of my primal CA-TD3 and compare the performance with the primal-dual algorithm, which serves as a benchmark. However, in terms of the classic autoregressive channel model, I can validate the algorithm on multiple different scenarios, such as the impact of different correlated coefficients. Regarding the CDL channel, I also utilise one typical setting according to the 3GPP standard. However, the algorithm can be applied to more diverse scenarios with different values of time-delay spread, mobile speed, and subcarrier spacing. In this way, the results will provide more insights for xURLLC in terms of modulation and coding scheme selection. Moreover, since the practical channel changes frequently, it is crucial to enable the learning model to be adaptive for different varying channels, despite the fact that the learning model is trained on a basic channel model. Transfer learning is one solution that I can further validate. Nevertheless, transfer learning can only fine-tune the parameters of a learning model in one distinct environment, which is inefficient for practical xURLLC systems and cannot be applied in a real-time pattern. Therefore, I can also apply the meta-learning method [98] to train the learning model on multiple different correlated channel models, which will further improve the efficiency, compatibility, and adaptability of my proposed algorithm.

In Chapter 4, I developed an experimental platform for WTSN. Due to the restrictions in available open-source COTS chips, I was able to deploy the framework on 802.11 b/g/n hardware. I will upgrade the platform to 802.11 ax/be hardware in the future, which will bring more bandwidth and exploitable features, such as OFDMA

and Basic Service Set Coloring. Moreover, current TDMA scheduling still follows a static pattern, which cannot combat the varying channels. An adaptive data rate and MCS selection algorithm based on the channel status need to be proposed, where I can take advantage of periodic CSI feedback in 802.11 networks and machine learning techniques. In addition, the developed testbed can be used to validate the resource allocation policy proposed in Chapter 2 and 3, which is also considered as future work.

Bibliography

[1] G. J. Sutton, J. Zeng, R. P. Liu, W. Ni, D. N. Nguyen, B. A. Jayawickrama, X. Huang, M. Abolhasan, Z. Zhang, E. Dutkiewicz, and T. Lv, "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, Feb. 2019.

[2] C. She, C. Sun, Z. Gu, Y. Li, C. Yang, H. V. Poor, and B. Vucetic, "A tutorial on ultrareliable and low-latency communications in 6G: integrating domain knowledge into deep learning," *Proceedings of the IEEE*, vol. 109, no. 3, pp. 204–246, Mar. 2021.

[3] E. Sisinni, A. Saifullah, S. Han, U. Jennehag, and M. Gidlund, "Industrial internet of things: challenges, opportunities, and directions," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 11, pp. 4724–4734, 2018.

[4] S. S. Husain, A. Kunz, A. Prasad, E. Pateromichelakis, and K. Samdanis, "Ultra-high reliable 5G V2X communications," *IEEE Communications Standards Magazine*, vol. 3, no. 2, pp. 46–52, Jun. 2019.

[5] M. Simsek, A. Aijaz, M. Dohler, J. Sachs, and G. Fettweis, "5G-enabled tactile internet," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 3, pp. 460–473, Feb. 2016.

[6] B. Holfeld, D. Wieruch, T. Wirth, L. Thiele, S. A. Ashraf, J. Huschke, I. Aktas, and J. Ansari, "Wireless communication for factory automation: an opportunity for LTE and 5G systems," *IEEE Communications Magazine*, vol. 54, no. 6, pp. 36–43, Jun. 2016.

[7] R. Gupta, S. Tanwar, S. Tyagi, and N. Kumar, "Tactile-internet-based telesurgery system for healthcare 4.0: an architecture, research challenges, and future directions," *IEEE Network*, vol. 33, no. 6, pp. 22–29, Dec. 2019.

[8] 3GPP, TS 38.824, *Study on physical layer enhancements for NR ultra-reliable and low latency case (URLLC).* Release 16, Mar. 2019.

[9] M. Shafi, A. F. Molisch, P. J. Smith, T. Haustein, P. Zhu, P. De Silva, F. Tufvesson, A. Benjebbour, and G. Wunder, "5G: A tutorial overview of standards, trials, challenges, deployment, and

practice," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 6, pp. 1201–1221, 2017.

[10] Y. Polyanskiy, H. V. Poor, and S. Verdu, "Channel coding rate in the finite blocklength regime," *IEEE Transactions on Information Theory*, vol. 56, no. 5, pp. 2307–2359, 2010.

[11] Y. Zhu, Y. Hu, A. Schmeink, and J. Gross, "Energy minimization of mobile edge computing networks with harq in the finite blocklength regime," *IEEE Transactions on Wireless Communications*, pp. 1–1, 2022.

[12] S. E. Elayoubi, P. Brown, M. Deghel, and A. Galindo-Serrano, "Radio resource allocation and retransmission schemes for URLLC over 5G networks," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 4, pp. 896–904, 2019.

[13] C. Sun, C. She, C. Yang, T. Q. S. Quek, Y. Li, and B. Vucetic, "Optimizing resource allocation in the short blocklength regime for ultra-reliable and low-latency communications," *IEEE Transactions on Wireless Communications*, vol. 18, no. 1, pp. 402–415, 2019.

[14] C. Li, C. She, N. Yang, and T. Q. S. Quek, "Secure transmission rate of short packets with queueing delay requirement," *IEEE Transactions on Wireless Communications*, vol. 21, no. 1, pp. 203–218, 2022.

[15] 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.

[16] M. Biguesh and A. Gershman, "Training-based MIMO channel estimation: a study of estimator tradeoffs and optimal training signals," *IEEE Transactions on Signal Processing*, vol. 54, no. 3, pp. 884–893, Mar. 2006.

[17] H. Ren, C. Pan, Y. Deng, M. Elkashlan, and A. Nallanathan, "Joint power and blocklength optimization for URLLC in a factory automation scenario," *IEEE Transactions on Wireless Communications*, vol. 19, no. 3, pp. 1786–1801, 2020.

[18] W. R. Ghanem, V. Jamali, Y. Sun, and R. Schober, "Resource allocation for multi-user downlink MISO OFDMA-URLLC systems," *IEEE Transactions on Communications*, vol. 68, no. 11, pp. 7184–7200, 2020.

[19] Y. Lin, C. Shen, Y. Hu, B. Ai, and Z. Zhong, "Joint design of channel training and data transmission for MISO-URLLC systems," *IEEE Transactions on Wireless Communications*, pp. 1–1, Apr. 2022.

[20] J. Zeng, T. Lv, R. P. Liu, X. Su, Y. J. Guo, and N. C. Beaulieu, "Enabling ultrareliable and low-latency communications under shadow fading by massive MU-MIMO," *IEEE Internet of Things Journal*, vol. 7, no. 1, pp. 234–246, 2020.

[21] 3GPP, TS 38.214, *NR; Physical layer procedures for data*. Release 16, Jun. 2021.

[22] H. Yang, K. Zheng, K. Zhang, J. Mei, and Y. Qian, "Ultra-reliable and low-latency communications for connected vehicles: challenges and solutions," *IEEE Network*, vol. 34, no. 3, pp. 92–100, May. 2020.

[23] R. Dong, C. She, W. Hardjawana, Y. Li, and B. Vucetic, "Deep learning for radio resource allocation with diverse quality-of-service requirements in 5G," *IEEE Transactions on Wireless Communications*, vol. 20, no. 4, pp. 2309–2324, 2021.

[24] H. Sun, X. Chen, Q. Shi, M. Hong, X. Fu, and N. D. Sidiropoulos, "Learning to optimize: training deep neural networks for interference management," *IEEE Transactions on Signal Processing*, vol. 66, no. 20, pp. 5438–5453, 2018.

[25] L. Liu, B. Yin, S. Zhang, X. Cao, and Y. Cheng, "Deep learning meets wireless network optimization: identify critical links," *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 1, pp. 167–180, Jan. 2020.

[26] C. Sun and C. Yang, "Learning to optimize with unsupervised learning: training deep neural networks for URLLC," in *Proc. 2019 IEEE 30th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, 2019, pp. 1–7.

[27] L. Liang, H. Ye, G. Yu, and G. Y. Li, "Deep-learning-based wireless resource allocation with application to vehicular networks," *Proceedings of the IEEE*, vol. 108, no. 2, pp. 341–356, 2020.

[28] M. Eisen, C. Zhang, L. F. O. Chamon, D. D. Lee, and A. Ribeiro, "Learning optimal resource allocations in wireless systems," *IEEE Transactions on Signal Processing*, vol. 67, no. 10, pp. 2775–2790, May. 2019.

[29] N. C. Luong, D. T. Hoang, S. Gong, D. Niyato, P. Wang, Y.-C. Liang, and D. I. Kim, "Applications of deep reinforcement learning in communications and networking: a survey," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 4, pp. 3133–3174, May. 2019.

[30] F. Hamidi-Sepehr, M. Sajadieh, S. Panteleev, T. Islam, I. Karls, D. Chatterjee, and J. Ansari, "5G URLLC: evolution of high-performance wireless networking for industrial automation," *IEEE Communications Standards Magazine*, vol. 5, no. 2, pp. 132–140, 2021.

[31] "IEEE 802.1 Task Group," https://1.ieee802.org/tsn/.

[32] O. Seijo, I. Val, and J. A. Lopez-Fernandez, "w-SHARP: Implementation of a high-performance wireless time-sensitive network for low latency and ultra-low cycle time industrial applications," *IEEE Transactions on Industrial Informatics*, pp. 1–1, Jul. 2020.

[33] Z. Pang, M. Luvisotto, and D. Dzung, "Wireless high-performance communications: The challenges and opportunities of a new target," *IEEE Industrial Electronics Magazine*, vol. 11, no. 3, pp. 20–25, Sept. 2017.

[34] S. Schiessl, J. Gross, M. Skoglund, and G. Caire, "Delay performance of the multiuser MISO downlink under imperfect CSI and finite-length coding," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 4, pp. 765–779, 2019.

[35] F. Librino and P. Santi, "The complexity–performance tradeoff in resource allocation for URLLC exploiting dynamic CSI," *IEEE Internet of Things Journal*, vol. 8, no. 17, pp. 13 266–13 277, Sept. 2021.

[36] H. Ren, K. Wang, and C. Pan, "Intelligent reflecting surface-aided URLLC in a factory automation scenario," *IEEE Transactions on Communications*, vol. 70, no. 1, pp. 707–723, Jan. 2022.

[37] J. Cao, X. Zhu, Y. Jiang, Y. Liu, Z. Wei, S. Sun, and F.-C. Zheng, "Independent pilots versus shared pilots: short frame structure optimization for heterogeneous-traffic URLLC networks," *IEEE Transactions on Wireless Communications*, vol. 21, no. 8, pp. 5755–5769, Aug. 2022.

[38] W. Xia, G. Zheng, Y. Zhu, J. Zhang, J. Wang, and A. P. Petropulu, "A deep learning framework for optimization of MISO downlink beamforming," *IEEE Transactions on Communications*, vol. 68, no. 3, pp. 1866–1880, 2020.

[39] F. B. Mismar, B. L. Evans, and A. Alkhateeb, "Deep reinforcement learning for 5G networks: joint beamforming, power control, and interference coordination," *IEEE Transactions on Communications*, vol. 68, no. 3, pp. 1581–1592, 2020.

[40] R. Dong, C. She, W. Hardjawana, Y. Li, and B. Vucetic, "Deep learning for hybrid 5G services in mobile edge computing systems: learn from a digital twin," *IEEE Transactions on Wireless Communications*, vol. 18, no. 10, pp. 4692–4707, Jul. 2019.

[41] J. Fan, Z. Wang, Y. Xie, and Z. Yang, "A theoretical analysis of deep Q-learning," in *Proceedings of the 2nd Conference on Learning for Dynamics and Control*, ser. Proceedings of Machine Learning Research, A. M. Bayen, A. Jadbabaie, G. Pappas, P. A. Parrilo, B. Recht, C. Tomlin, and M. Zeilinger, Eds., vol. 120. PMLR, 10–11 Jun 2020, pp. 486–489. [Online]. Available: https://proceedings.mlr.press/v120/yang20a.html

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

[43] N. S. Saatchi, H.-C. Yang, and Y.-C. Liang, "Novel adaptive transmission scheme for effective URLLC support in 5G NR: a model-based reinforcement learning solution," *IEEE Wireless Communications Letters*, vol. 12, no. 1, pp. 109–113, Jan. 2023.

[44] M. Alsenwi, N. H. Tran, M. Bennis, S. R. Pandey, A. K. Bairagi, and C. S. Hong, "Intelligent resource slicing for eMBB and URLLC coexistence in 5G and beyond: a deep reinforcement learning based approach," *IEEE Transactions on Wireless Communications*, vol. 20, no. 7, pp. 4585–4600, Jul. 2021.

[45] Z. Meng, C. She, G. Zhao, and D. De Martini, "Sampling, communication, and prediction co-design for synchronizing the real-world device and digital model in metaverse," *IEEE Journal on Selected Areas in Communications*, vol. 41, no. 1, pp. 288–300, Jan. 2023.

[46] S. Li, C. She, Y. Li, and B. Vucetic, "Constrained deep reinforcement learning for low-latency wireless VR video streaming," in *2021 IEEE Global Communications Conference (GLOBECOM)*, 2021, pp. 01–06.

[47] J. Song, S. Han, A. Mok, D. Chen, M. Lucas, M. Nixon, and W. Pratt, "WirelessHART: Applying wireless technology in real-time industrial process control," in *Proc. 2008 IEEE Real-Time and Embedded Technology and Applications Symposium*, 2008, pp. 377–386.

[48] "ISA100," http://www.isa.org/isa100.

[49] K. An, M. Lin, J. Ouyang, and W.-P. Zhu, "Secure transmission in cognitive satellite terrestrial networks," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 11, pp. 3025–3037, Oct. 2016.

[50] Z. Lin, H. Niu, K. An, Y. Wang, G. Zheng, S. Chatzinotas, and Y. Hu, "Refracting RIS aided hybrid satellite-terrestrial relay networks: Joint beamforming design and optimization," *IEEE Transactions on Aerospace and Electronic Systems*, pp. 1–1, Mar. 2022.

[51] B. Holfeld, D. Wieruch, T. Wirth, L. Thiele, S. A. Ashraf, J. Huschke, I. Aktas, and J. Ansari, "Wireless communication for factory automation: an opportunity for LTE and 5G systems," *IEEE Communications Magazine*, vol. 54, no. 6, pp. 36–43, Jun. 2016.

[52] P. Schulz, M. Matthe, H. Klessig, M. Simsek, G. Fettweis, J. Ansari, S. A. Ashraf, B. Almeroth, J. Voigt, I. Riedel, A. Puschmann, A. Mitschele-Thiel, M. Muller, T. Elste, and M. Windisch, "Latency critical IoT applications in 5G: Perspective on the design of radio interface and network architecture," *IEEE Communications Magazine*, vol. 55, no. 2, pp. 70–78, Feb. 2017.

[53] D. Cavalcanti, J. Perez-Ramirez, M. M. Rashid, J. Fang, M. Galeev, and K. B. Stanton, "Extending accurate time distribution and timeliness capabilities over the air to enable future wireless industrial automation systems," *Proceedings of the IEEE*, vol. 107, no. 6, pp. 1132–1152, Mar. 2019.

[54] G. Patti, G. Alderisi, and L. Lo Bello, "SchedWiFi: An innovative approach to support scheduled traffic in ad-hoc industrial IEEE 802.11 networks," in *Proc. 2015 IEEE 20th Conference on Emerging Technologies Factory Automation (ETFA)*, 2015, pp. 1–9.

[55] P. G. Peón, E. Uhlemann, W. Steiner, and M. Björkman, "Medium access control for wireless networks with diverse time and safety real-time requirements," in *Proc. IECON 2016-42nd Annual Conference of the IEEE Industrial Electronics Society*, 2016, pp. 4665–4670.

[56] G. Cena, S. Scanzio, and A. Valenzano, "Seamless link-level redundancy to improve reliability of industrial Wi-Fi networks," *IEEE Transactions on Industrial Informatics*, vol. 12, no. 2, pp. 608–620, Jan 2016.

[57] Y. Wei, Q. Leng, S. Han, A. K. Mok, W. Zhang, and M. Tomizuka, "RT-WiFi: Real-time high-speed communication protocol for wireless cyber-physical control applications," in *Proc. 2013 IEEE 34th Real-Time Systems Symposium*, 2013, pp. 140–149.

[58] P. Djukic and P. Mohapatra, "Soft-TDMAC: A software TDMA-based MAC over commodity 802.11 hardware," in *Proc. IEEE INFOCOM 2009*, 2009, pp. 1836–1844.

[59] Y. Cheng, D. Yang, and H. Zhou, "Det-WiFi: A multihop TDMA MAC implementation for industrial deterministic applications based on commodity 802.11 hardware," *Wireless Communications and Mobile Computing*, vol. 2017, Apr. 2017.

[60] A. Aijaz, "High-performance industrial wireless: Achieving reliable and deterministic connectivity over IEEE 802.11 WLANs," *IEEE Open Journal of the Industrial Electronics Society*, vol. 1, pp. 28–37, Mar 2020.

[61] "Link aggregaion," https://1.ieee802.org/tsn/802-1ax-rev/.

[62] F. Tramarin, S. Vitturi, M. Luvisotto, and A. Zanella, "On the use of IEEE 802.11n for industrial communications," *IEEE Transactions on Industrial Informatics*, vol. 12, no. 5, pp. 1877–1886, Dec. 2015.

[63] J. Neander, T. Lennvall, and M. Gidlund, "Prolonging wireless HART network lifetime using packet aggregation," in *Proc. 2011 IEEE International Symposium on Industrial Electronics*, 2011, pp. 1230–1236.

[64] F. Li, Z. Zhang, Z. Jia, and L. Ju, "Superframe scheduling for data aggregation in wirelessHART networks," in *Proc. 2015 IEEE 17th International Conference on High Performance Computing and Communications, 2015 IEEE 7th International Symposium on Cyberspace Safety and Security, and 2015 IEEE 12th International Conference on Embedded Software and Systems*, 2015, pp. 1540–1545.

[65] S. Girs, A. Willig, E. Uhlemann, and M. Björkman, "Scheduling for source relaying with packet aggregation in industrial wireless networks," *IEEE Transactions on Industrial Informatics*, vol. 12, no. 5, pp. 1855–1864, Oct. 2016.

[66] W. Liang, M. Zheng, J. Zhang, H. Shi, H. Yu, Y. Yang, S. Liu, W. Yang, and X. Zhao, "WIA-FA and its applications to digital factory: A wireless network solution for factory automation," *Proceedings of the IEEE*, vol. 107, no. 6, pp. 1053–1073, Feb. 2019.

[67] T. Xu, Y. Liang, and G. Lan, "CRPO: a new approach for safe reinforcement learning with convergence guarantee," in *Proceedings of the 38th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, M. Meila and T. Zhang, Eds., vol. 139. PMLR, 18–24 Jul 2021, pp. 11 480–11 491. [Online]. Available: https://proceedings.mlr.press/v139/xu21a.html

[68] K. Baddour and N. Beaulieu, "Autoregressive modeling for fading channel simulation," *IEEE Transactions on Wireless Communications*, vol. 4, no. 4, pp. 1650–1662, Jul. 2005.

[69] 3GPP, TR 38.901, *Study on channel model for frequencies from 0.5 to 100 GHz*. Release 17, Apr. 2022.

[70] S. Schiessl, J. Gross, and H. Al-Zubaidy, "Delay analysis for wireless fading channels with finite blocklength channel coding," in *Proc. ACM MSWiM*, 2015.

[71] J. Gregory and C. Lin, *Constrained optimization in the calculus of variations and optimal control theory*. Chapman and Hall/CRC, 2018.

[72] M. Giordani, M. Polese, A. Roy, D. Castor, and M. Zorzi, "A tutorial on beam management for 3GPP NR at mmWave frequencies," *IEEE Communications Surveys Tutorials*, vol. 21, no. 1, pp. 173–196, Sep. 2019.

[73] M. Hussain and N. Michelusi, "Learning and adaptation for millimeter-wave beam tracking and training: a dual timescale variational framework," *IEEE Journal on Selected Areas in Communications*, vol. 40, no. 1, pp. 37–53, Nov. 2022.

[74] S. Boyd, S. P. Boyd, and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[75] Y. Liu, C. She, Y. Zhong, W. Hardjawana, F.-C. Zheng, and B. Vucetic, "Interference-limited ultra-reliable and low-latency communications: Graph neural networks or stochastic geometry?" 2022. [Online]. Available: https://arxiv.org/abs/2207.06918

[76] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[77] A. Goldsmith, *Wireless communications*. Cambridge university press, 2005.

[78] M. Setayesh, S. Bahrami, and V. W. Wong, "Resource slicing for eMBB and URLLC Services in radio access network using hierarchical deep learning," *IEEE Transactions on Wireless Communications*, vol. 21, no. 11, pp. 8950–8966, Nov. 2022.

[79] S. Fujimoto, H. Hoof, and D. Meger, "Addressing function approximation error in actor-critic methods," in *International conference on machine learning*. PMLR, 2018, pp. 1587–1596.

[80] Y. Chow, M. Ghavamzadeh, L. Janson, and M. Pavone, "Risk-constrained reinforcement learning with percentile risk criteria," *The Journal of Machine Learning Research*, vol. 18, no. 1, pp. 6070–6120, Jan. 2017.

[81] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," 2015. [Online]. Available: https://arxiv.org/abs/1509.02971

[82] 3GPP, TS 38.211, *Physical channels and modulation*. Release 17, Apr. 2022.

[83] H. Xiao, W. Tian, W. Liu, and J. Shen, "ChannelGAN: deep learning-based channel modeling and generating," *IEEE Wireless Communications Letters*, vol. 11, no. 3, pp. 650–654, Jan. 2022.

[84] L. Wang, G. Liu, J. Xue, and K.-K. Wong, "Channel prediction using ordinary differential equations for MIMO systems," *IEEE Transactions on Vehicular Technology*, vol. 72, no. 2, pp. 2111–2119, Oct. 2023.

[85] R. S. Mogensen, I. Rodriguez, G. Berardinelli, A. Fink, R. Marcker, S. Markussen, T. Raunholt, T. Kolding, G. Pocovi, and S. Barbera, "Implementation and trial evaluation of a wireless manufacturing execution system for industry 4.0," in *Proc. 2019 IEEE 90th Vehicular Technology Conference (VTC2019-Fall)*, 2019, pp. 1–7.

[86] T. Carlsson, "Industrial network market shares 2020 according to hms networks," https://www.hms-networks.com/news-and-insights/news-from-hms/2020/05/29/industrial-network-market-shares-2020-according-to-hms-networks, May. 29, 2020.

[87] D. Vassis, G. Kormentzas, A. Rouskas, and I. Maglogiannis, "The IEEE 802.11g standard for high data rate WLANs," *IEEE network*, vol. 19, no. 3, pp. 21–26, Jun. 2005.

[88] J. Okech, Y. Hamam, A. Kurien, T. Olwal, and M. Odhiambo, "A dynamic packet aggregation scheme for VoIP in wireless mesh networks," *Proc. of International Journal Of Computer Science*, 2008.

[89] "Qotom," https://www.qotom.net/.

[90] D. C. Mur, "Linux Wi-Fi open source drivers-mac80211, ath9k/ath5k."

[91] J. Kannisto, T. Vanhatupa, M. Hannikainen, and T. D. Hamalainen, "Software and hardware prototypes of the IEEE 1588 precision time protocol on wireless LAN," in *Proc. 14th IEEE Workshop on Local Metropolitan Area Networks*, 2005, pp. 6 pp.–6.

[92] "Precision time protocol daemon," http://ptpd.sourceforge.net/.

[93] D. Xia, J. Hart, and Q. Fu, "On the performance of rate control algorithm minstrel," in *Proc. 2012 IEEE 23rd International Symposium on Personal, Indoor and Mobile Radio Communications-(PIMRC)*, 2012, pp. 406–412.

[94] M. Luvisotto, Z. Pang, and D. Dzung, "Ultra high performance wireless control for critical applications: Challenges and directions," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 3, pp. 1448–1459, Jun. 2017.

[95] J. Åkerberg, M. Gidlund, and M. Björkman, "Future research challenges in wireless sensor and actuator networks targeting industrial automation," in *Proc. 2011 9th IEEE International Conference on Industrial Informatics*, 2011, pp. 410–415.

[96] W. Saad, M. Bennis, and M. Chen, "A vision of 6G wireless systems: applications, trends, technologies, and open research problems," *IEEE Network*, vol. 34, no. 3, pp. 134–142, Oct. 2020.

[97] J. Park, S. Samarakoon, H. Shiri, M. K. Abdel-Aziz, T. Nishio, A. Elgabli, and M. Bennis, "Extreme ultra-reliable and low-latency communication," Mar 2022. [Online]. Available: https://www.nature.com/articles/s41928-022-00728-8

[98] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *Proceedings of the 34th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, D. Precup and Y. W. Teh, Eds., vol. 70. PMLR, 06–11 Aug 2017, pp. 1126–1135. [Online]. Available: https://proceedings.mlr.press/v70/finn17a.html