Safe DRL for Resource Allocation with PAoI Violation Guarantees

by

Berire Güneş Reyhan

A Dissertation Submitted to the
Graduate School of Sciences and Engineering
in Partial Fulfillment of the Requirements for
the Degree of

Master of Science

in

Electrical and Electronics Engineering

KOÇ ÜNİVERSİTESİ

July, 2025

Safe DRL for Resource Allocation with PAoI Violation Guarantees

Koç University

Graduate School of Sciences and Engineering

This is to certify that I have examined this copy of a master's thesis by

Berire Güneş Reyhan

and have found that it is complete and satisfactory in all respects,
and that any and all revisions required by the final
examining committee have been made.

Committee Members:

_____

Prof. Dr. Sinem Çöleri (Advisor)

_____

Prof. Dr. Alper Tunga Erdoğan

_____

Prof. Dr. Mehmet Kemal Özdemir

Date: _____

I dedicate this thesis to my dearest family and my beloved husband for their unwavering love, endless patience, and constant encouragement.

ABSTRACT

Safe DRL for Resource Allocation with PAoI Violation Guarantees
Berire Güneş Reyhan
Master of Science in Electrical and Electronics Engineering
July, 2025

In Wireless Networked Control Systems (WNCSs), control and communication systems must be co-designed due to their strong interdependence. This thesis presents a novel optimization theory-based safe deep reinforcement learning (DRL) framework for ultra-reliable WNCSs, ensuring constraint satisfaction while optimizing performance, for the first time in the literature. The approach minimizes power consumption under key constraints, including Peak Age of Information (PAoI) violation probability, transmit power, and schedulability in the finite blocklength (FBL) regime. PAoI violation probability is uniquely derived by combining stochastic maximum allowable transfer interval (MATI) and maximum allowable packet delay (MAD) constraints in a multi-sensor network. The framework consists of two stages: optimization theory and safe DRL. The first stage derives optimality conditions to establish mathematical relationships among variables, simplifying and decomposing the problem. The second stage employs a safe DRL model where a teacher-student framework guides the DRL agent (student). The control mechanism (teacher) evaluates compliance with system constraints and suggests the nearest feasible action when needed. Extensive simulations show that the proposed framework outperforms rule-based and other optimization theory based DRL benchmarks, achieving faster convergence, higher rewards, and greater stability.

ÖZETÇE

Yüksek Lisans Tez Başlığı
Berire Güneş Reyhan
Elektrik ve Elektronik Mühendisliği, Yüksek Lisans
Temmuz 2025

Kablosuz Ağ Tabanlı Kontrol Sistemlerinde (WNCS), kontrol ve iletişim sistemleri arasındaki güçlü karşılıklı bağımlılık, bu iki yapının eşzamanlı olarak tasarlanmasını zorunlu kılmaktadır. Bu tez, literatürde ilk kez, performansı optimize ederken sistem kısıtlamalarını sağlamayı garantileyen ultra güvenilir WNCS'ler için yeni bir optimizasyon teorisi tabanlı güvenli Derin Pekiştirmeli Öğrenme (DRL) metodu sunmaktadır. Önerilen algoritma, sonlu blok uzunluğu (FBL) rejiminde, Bilginin Tepe Yaşı (PAoI) ihlal olasılığı, iletim gücü (transmit power) ve planlanabilirlik (schedulability) dahil olmak üzere temel kısıtlamalar altında güç tüketimini en aza indirir. PAoI ihlal olasılığı, bir çoklu sensör ağında, stokastik maksimum izin verilen aktarım aralığı (MATI) ve maksimum izin verilen paket gecikmesi (MAD) kısıtlamalarının birleştirilmesiyle benzersiz bir şekilde türetirlir. Önerilen metot iki aşamadan oluşur: optimizasyon teorisi ve güvenli DRL. İlk aşama, değişkenler arasında matematiksel ilişkiler kurmak, sorunu basitleştirmek ve ayrıştırmak için optimallik koşullarını (optimality conditions) türetirlir. İkinci aşama, bir öğretmen-ögrenci çerçevesinin (teacher-student framework) DRL etmenini (öğrenci) yönlendirdiği güvenli bir DRL modeli kullanır. Kontrol mekanizması (öğretmen) seçilen eylemin sistem kısıtlamalarına uyumunu değerlendirir ve gerektiğinde sistem gereksinimlerini sağlayan en yakın eylemi önerir. Kapsamlı simülasyonlar, önerilen metodun kural tabanlı ve diğer optimizasyon teorisi tabanlı DRL kıyaslama ölçütlerinden daha iyi performans gösterdiğini, daha hızlı yakınsama, daha yüksek ödüller ve daha büyük istikrar elde ettiğini göstermektedir.

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to my advisor, Prof. Dr. Sinem Çöleri, whose continuous encouragement, insightful guidance, and deep knowledge have played a vital role in shaping this thesis. Her unwavering support throughout this journey has been truly invaluable.

I am deeply grateful to my parents, Ömer and Fatıma, and my sister, Serra, for their unconditional love, support, and belief in me at every step.

I would also like to thank my dear friends, Melis and Recep, for their friendship, motivation, and encouragement, which have made this process more enjoyable and fulfilling.

Last but certainly not least, my heartfelt thanks go to my beloved husband, Fatih, for his endless patience, understanding, and steadfast support. His presence has been a constant source of strength and comfort during this journey.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

ADP	Adaptive Dynamic Programming
AoI	Age of Information
CMDP	Constrained Markov Decision Process
CSCG	Circularly Symmetric Complex Gaussian
CSI	Channel State Information
DDPG	Deep Deterministic Policy Gradient
DDQN	Double Deep Q-Network
DL	Deep Learning
DNN	Deep Neural Network
DQN	Deep Q-Network
D3QN	Dueling Double Deep Q-Network
DRL	Deep Reinforcement Learning
FBL	Finite Blocklength
FNN	Feedforward Neural Network
KPI	Key Performance Indicator
LQR	Linear Quadratic Regulator
MAD	Maximum Allowable Delay
MATI	Maximum Allowable Transfer Interval
PAoI	Peak Age of Information
PAoL	Peak Age of Loop
PGF	Probability Generating Function
QoS	Quality of Service
RA	Resource Allocation
RL	Reinforcement Learning
RSMA	Rate-Splitting Multiple Access

SNR	Signal-to-Noise Ratio
SPSA	Simultaneous Perturbation Stochastic Approximation
STP	Secrecy Timeout Probability
TDMA	Time Division Multiple Access
URLLC	Ultra-Reliable Low-Latency Communication
UAV	Unmanned Aerial Vehicle
WNCS	Wireless Networked Control System

Chapter 1

INTRODUCTION

The content of this thesis has been published in IEEE Transactions on Communications (TCOM).[1]

Wireless Networked Control Systems (WNCSs) comprise spatially distributed sensors, actuators, and controllers that communicate via wireless networks [Park et al., 2018]. Utilizing wireless communication within control systems leads to cost-effective and adaptable network architectures, reducing expenses associated with system component installation, modification, and upgrades when compared to their wired counterparts. As a result, WNCSs have been successfully deployed in a wide range of applications, including automotive electrical systems [Sadi and Coleri Ergen, 2013], avionics control systems [International Telecommunication Union – Radiocommunication Sector (ITU-R), 2013], building automation systems [Witrant et al., 2010], and industrial automation systems [Pister et al., 2009]. The main challenge in designing the communication system for a WNCS lies in maintaining control system performance and stability despite the inherent unreliability of wireless transmissions, transmission delays, the shared wireless medium, and the limited battery resources of sensor nodes.

The performance and stability of WNCSs are commonly evaluated using key metrics, including stochastic maximum allowable transfer interval (MATI), maximum allowable packet delay (MAD) [Sadi et al., 2014, Sadi and Coleri Ergen, 2017, Ali et al., 2024], Age of Information (AoI) [Costa et al., 2016, Kosta et al., 2021], and

[1] B. Gunes Reyhan and S. Coleri, "Safe Deep Reinforcement Learning for Resource Allocation with Peak Age of Information Violation Guarantees," in *IEEE Transactions on Communications*, doi: 10.1109/TCOMM.2025.3591167.

Peak Age of Information (PAoI) [Kosta et al., 2021, Zhang et al., 2021b, Devassy et al., 2019, Östman et al., 2019, Kartal et al., 2023, Zhang et al., 2021a]. Stochastic MATI entails maintaining the time interval between consecutive state vector reports from sensor nodes to the controller below a specified threshold with a given probability, while MAD denotes the maximum permissible delay for packets transmitted from sensor nodes to the controller [Sadi et al., 2014, Sadi and Coleri Ergen, 2017, Ali et al., 2024]. AoI quantifies the freshness of information at the destination node, with PAoI capturing the maximum AoI immediately before the reception of an update [Costa et al., 2016]. A crucial performance metric in this context is the probability that the PAoI exceeds a predetermined threshold, referred to as the PAoI violation probability. This metric is essential for designing systems that guarantee the freshness of the information at the destination at any given time.

The PAoI violation probability has been formulated in the literature for both single-node and multi-node systems. [Kosta et al., 2021] examines the distributions of AoI and PAoI in a source-destination communication link for a wide class of discrete time status update systems. In addition, [Zhang et al., 2021b] computes PAoI violation probability indirectly by deriving an upper-bound using the Mellin transform for point-to-point communication in the finite blocklength (FBL) regime. Moreover, [Devassy et al., 2019] characterizes PAoI violation probability through the probability-generating function (PGF) of the steady-state PAoI for a single-node system using FBL theory. Extending this analysis to multi-node systems, [Östman et al., 2019] and [Kartal et al., 2023] also employ PGF to obtain PAoI violation probability. Furthermore, [Zhang et al., 2021a] derives an upper bound on PAoI violation probability in the FBL regime for a system comprising multiple unmanned aerial vehicles (UAVs) by applying the Mellin transform. However, none of these studies directly derive the PAoI violation probability for a multi-node system satisfying ultra-reliable low-latency communication (URLLC) requirements within the FBL regime.

WNCS optimization problems aim to determine the optimal values for packet error probability, sampling time, and network scheduling parameters by utilizing various objective and constraint functions. These problems are often solved through

either model-based techniques [Sadi et al., 2014, Sadi and Coleri Ergen, 2017, Zhang et al., 2021b, Kartal et al., 2023, Wang et al., 2024, Cao et al., 2023, Zhang et al., 2021a] or machine learning-based approaches [Zhao et al., 2024, Ashraf et al., 2022, Ali et al., 2024]. Model-based approaches typically involve deriving mathematical models that capture the system behavior and then solving optimization problems using established analytical techniques. In [Sadi et al., 2014], the optimization of energy-efficient data transmission in a WNCS is addressed, aiming to minimize power consumption while satisfying constraints related to system stability, schedulability, and maximum transmit power. The authors use relaxation techniques to simplify and solve the complex Integer Programming problem. In [Sadi and Coleri Ergen, 2017], the focus shifts to jointly optimizing energy consumption and control system performance, with the goal of minimizing energy usage while ensuring high reliability and strict delay requirements. The solution employs a combination of optimality conditions, smart enumeration, and heuristic algorithms to achieve near-optimal performance. [Zhang et al., 2021b] jointly optimizes PAoI and delay-bound violation probabilities under FBL constraints, employing stochastic network calculus and convex optimization techniques to balance trade-offs between these metrics. In [Kartal et al., 2023], resource allocation is optimized to minimize delay and Peak AoI violation probabilities under packet error constraints in URLLC systems using a combination of analytical derivations and numerical optimization. [Zhang et al., 2021a] optimizes PAoI violation probability under the constraints of transmit power, UAV trajectories, and decoding error probability, using an iterative approach involving convex optimization. In [Wang et al., 2024], resource allocation and user pairing are optimized in rate splitting multiple access (RSMA)-based WNCSs to maximize sum rate while ensuring control stability under imperfect channel state information (CSI), using semi-definite programming relaxation, successive convex approximation, and hypergraph game theory. [Cao et al., 2023] analyzes Peak Age of Loop (PAoL) in WNCSs under the FBL regime, with closed-form expressions for its average, variance, and outage probability. A PAoL-oriented blocklength adaptation scheme with retransmission is proposed, optimizing uplink (UL) and downlink (DL) blocklengths using convex optimization and Newton's method while ensuring

stability with a Linear Quadratic Regulator (LQR) controller. However, the high complexity of these model-based techniques limits their applicability in low latency WNCS applications.

To address the complexity associated with model-based approaches, researchers have increasingly turned to machine learning techniques. [Zhao et al., 2024] presents a deep reinforcement learning (DRL) based algorithm that optimizes controllers and schedulers by leveraging both model-free and model-based data, incorporating awareness of sensor AoI states and dynamic channel conditions. Similarly, [Ashraf et al., 2022] proposes a DRL-based co-design strategy for jointly optimizing WNCSs to achieve the best control performance despite network uncertainties like delay and variable throughput. Recently, [Ali et al., 2024] introduces a DRL model grounded in optimization theory for co-designing control and communication systems, where the sampling period, blocklength, and packet error probability are optimized. However, ensuring feasibility and safety in DRL-based optimal resource allocation remains a challenge in WNCSs. While these studies leverage DRL for co-design, they do not incorporate mechanisms to explicitly enforce constraint satisfaction during training and decision-making. In contrast, our approach integrates optimization theory with safe DRL to guarantee constraint compliance throughout the learning process and final policy execution.

Numerous safe reinforcement learning (RL) methods have been explored across various domains. [Zhang et al., 2021c] investigates secure video streaming with low energy consumption in rotary-wing UAV-enabled wireless networks. The constraints satisfied with this safe method include secrecy timeout probability (STP) to meet long-term time delay requirements, while optimizing energy efficiency through video quality selection, power allocation, and trajectory optimization. A safe Deep Q-Network (DQN) is employed to develop a policy set using a Lyapunov function, dynamically satisfying the constraints of a Constrained Markov Decision Process (CMDP). However, this method ensures constraint satisfaction only on average over time using Lyapunov optimization, rather than guaranteeing that each decision individually meets all constraints, which may be insufficient for the strict real-time requirements of URLLC. On the other hand, [Schmoeller Roza et al., 2022] presents

a safety layer for robot navigation, where a convex optimization problem is formulated to identify the safest action that adheres to constraints while minimizing deviation from the current action. This method is effective for convex constraint systems but has been applied in domains such as robotic navigation, rather than communication networks. In [Nguyen et al., 2022], fuzzy logic is used to shape the reward function in Q-learning for vehicular crowdsensing. However, designing effective rules is non-trivial in systems with conflicting constraints, which limits their applicability to real-time URLLC scenarios, such as WNCSs. Moreover, [Zheng et al., 2022] introduces a teacher-student approach for resource allocation in networking, where the teacher employs simple white-box logic rules to guide the DRL student. While this method allows the teacher to offer precise action advice, the use of simple logic rules falls short in ensuring safety within RL-based URLLC systems. These rules lack the flexibility, adaptability, and scalability required to operate effectively in the complex, dynamic, and uncertain environments that characterize real-time communication networks. Although these safe RL methods have advanced the field, none has been specifically applied to wireless networked control systems (WNCSs) to ensure URLLC under real-time constraints.

This thesis presents a safe DRL framework grounded in optimization theory for resource allocation in WNCSs. With the goal of minimizing overall power consumption, resource allocation seeks to identify the optimal values for key control and communication system parameters, including sampling period, blocklength, and packet error probability. The framework addresses constraints, including finite blocklength, PAoI violation probability, maximum transmit power, and schedulability, which are critical for the joint design of control and communication systems. To enhance efficiency, the proposed framework leverages the optimality conditions derived from mathematical system models, reducing the number of decision variables and, consequently, the amount of training data required for the DRL model. Additionally, it employs a teacher-student framework, wherein the teacher intervenes and recommends safe actions when the DRL agent (the student) fails to meet system requirements. The novel contributions of the thesis are listed as follows:

- We propose a safe DRL framework for the joint optimization of controller and communication systems involving multiple nodes, incorporating PAoI violation probability constraint while considering URLLC conditions within the finite blocklength regime, for the first time in the literature.

- We introduce a novel and direct formulation for the PAoI violation probability by integrating stochastic MATI and MAD constraints in a multi-node WNCS under URLLC conditions, utilizing finite blocklength theory, for the first time in the literature.

- We propose a novel safe DRL approach grounded in optimization theory for optimal resource allocation in WNCSs, for the first time in the literature. The proposed approach reduces the training time for DRL by deriving optimality conditions and ensures system requirements are consistently met during exploration by employing safe DRL techniques.

- We evaluate the performance of the proposed optimization theory based safe DRL approach through extensive simulations, comparing it against rule-based DRL and several optimization theory based DRL approaches across various network sizes and constraint parameters.

The rest of the thesis is organized as follows. In Chapter 2, the system model and associated assumptions are described. Chapter 3 presents the joint optimization problem of control and communication systems. The proposed optimization theory based safe DRL approach is provided in Chapter 4. Simulation results are presented in Chapter 5. Finally, concluding remarks are given in Chapter 6.

Chapter 2

SYSTEM MODEL AND ASSUMPTIONS

We consider a Wireless Networked Control System architecture comprising multiple plants, each continuously monitored by a central controller through a wireless network, as illustrated in Fig. 2.1. The system consists of N sensors, each of which periodically measures and transmits the corresponding plant state. The transceiver at each sensor node is assumed to have a single antenna. Sensor node i transmits the plant state information in small packets of length L_i by using m_i symbols, referred to as the blocklength, where $i \in \{1, 2, \ldots, N\}$. Due to the inherent unreliability of the wireless communication channel, there is a possibility that the transmitted data may not reach the controller. For simplicity, packet errors are modeled as a Bernoulli random process. The controller utilizes the most recent state update to calculate and send new control commands to the actuators. The actuators establish and maintain stable communication links with the controller, ensuring the reliable and uninterrupted reception of control commands [Zhang et al., 2020]. We denote the sampling period, blocklength, packet transmission delay, and packet error probability of sensor node i by h_i, m_i, d_i, and p_i, respectively, for $i \in \{1, 2, \ldots, N\}$.

Time Division Multiple Access (TDMA) is assumed as the channel access method due to its guaranteed delay, making it suitable for industrial control applications [ANS,]. The channel time is segmented into frames, with each frame containing time slots. The first slot is reserved for the beacon frame. The controller periodically broadcasts this beacon to synchronize and update scheduling across the nodes in the WNCS. Nodes are assigned specific time slots for data transmission, with optimal settings for transmission power, blocklength, and sampling period tailored to each node's requirements. To ensure fixed determinism over the channel coherence time, during which wireless conditions remain stable, each sensor node is assigned a dedicated, fixed-duration time slot. These slots repeat periodically in alignment with the

sampling interval, enabling predictable and consistent transmissions. Nodes do not transmit concurrently, and the network manager continuously monitors the packet error rate. Each node can operate in three modes: active (sensing, transmitting, and receiving), sleep, and transient. Energy consumption is considered only during the active mode, as sleep and transient modes consume minimal energy [Cui et al., 2005].

Next, we discuss the optimization problem formulations related to the performance of the wireless communication and control systems. These include PAoI violation probability, power consumption, and schedulability expressions of the underlying communication system in the finite blocklength regime.

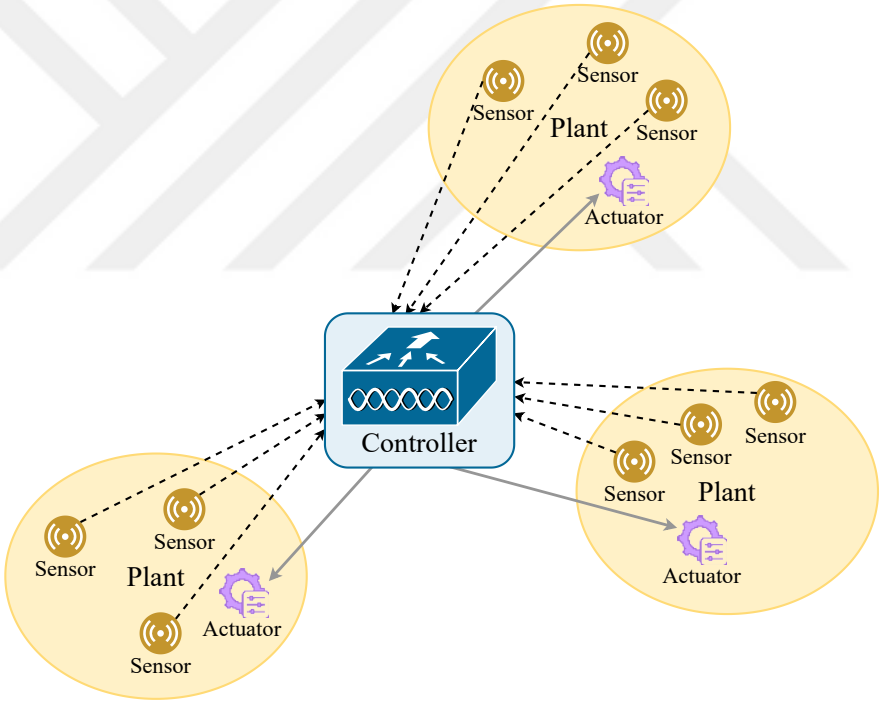

Figure 2.1: Overview of the WNCS setup.

2.1 Peak AoI Violation Probability

AoI is a critical metric that measures the freshness of data in a system by tracking the time elapsed since the last update was received. PAoI represents the maximum AoI just before a new update is received, indicating the worst-case scenario in terms

of data freshness. In the considered system, PAoI is a key performance indicator (KPI), since maintaining up-to-date information is vital for timely and accurate decision-making, directly influencing the quality of control decisions and overall system stability. The PAoI violation probability quantifies the likelihood that the PAoI exceeds a predefined critical threshold. Keeping PAoI violation probability under a threshold value ensures that the system maintains real-time data freshness. This constraint is formulated as

$$P[A_i(h_i, d_i(m_i), p_i) \geq \alpha] \leq (1 - \delta), \tag{2.1}$$

where A_i denotes the PAoI of sensor node i and is a function of the sampling period h_i, packet transmission delay d_i, and packet error probability p_i; the delay d_i is a function of the blocklength m_i; α denotes the endurable peak AoI and $(1 - \delta)$ indicates the maximum tolerated probability for peak AoI violation. The value of α is application-specific and can vary significantly. For some applications, particularly those requiring ultra-low latency, α might be on the order of a few milliseconds, while for others, such as periodic monitoring or less time-sensitive applications, it could extend to several minutes [Saurav and Vaze, 2023, Champati et al., 2021].

The PAoI violation probability can be rewritten as

$$P[\mu_i(h_i, d_i(m_i), p_i) + d_i(m_i) \geq \alpha] \leq (1 - \delta), \tag{2.2}$$

where μ_i denotes the time between two subsequent state vector updates of sensor node i. This expression combines MAD and stochastic MATI constraints in [Ali et al., 2024]. For given values of h_i and α, there are $\left\lfloor \frac{\alpha - d_i(m_i)}{h_i} \right\rfloor$ opportunities of the state vector update reception. We model the packet error p_i as a Bernoulli random process and rewrite the PAoI violation probability constraint as [Ali et al., 2024]

$$p_i^{\left\lfloor \frac{\alpha - \frac{m_i}{B}}{h_i} \right\rfloor} \leq (1 - \delta), \tag{2.3}$$

where B is the bandwidth of the wireless channel and $d_i(m_i) = \frac{m_i}{B}$ accounts for

the packet transmission delay based solely on the blocklength m_i. Given that fixed determinism is assumed, the queuing delay is excluded from the analysis.

2.2 Power Consumption

Power consumption is a vital consideration in energy-constrained systems with limited energy sources. Efficient power management is essential to maximize the operational lifespan of these systems, ensuring reliable performance over extended periods without the need for frequent recharging or battery replacement. Average power consumption W_i of node i is a function of the sampling period h_i, packet error probability p_i, and delay $d_i(m_i)$ such that:

$$W_i(h_i, d_i(m_i), p_i) = \frac{(W_{\text{tx},i} + W_i^c) d_i(m_i)}{h_i}, \tag{2.4}$$

where $W_{\text{tx},i}$ is the transmit power of node i and W_i^c is the circuit power consumption when the transmitter is in the active mode.

The coding rate, also referred to as the data rate of a communication system, is the ratio of the number of information bits to the total number of channel uses (also known as the blocklength). In URLLC scenarios, where the blocklength per frame is small, the decoding error probability becomes significant and cannot be neglected. The coding rate R_i (in bits/sec/Hz) for node i under the finite blocklength regime can be approximated as [Ali et al., 2024]

$$R_i \approx \log_2(1 + \gamma_i) - \sqrt{\frac{V_i}{m_i}} \frac{Q^{-1}(\epsilon_i)}{\ln(2)}, \tag{2.5}$$

where $\gamma_i = \frac{W_{\text{tx},i}|g_i|}{\sigma^2}$ is the signal-to-noise ratio (SNR) at the controller; g_i is the channel gain from node i to the controller; σ^2 is the noise power; $V_i = 1 - (1+\gamma_i)^{-2}$ is the channel dispersion; ϵ_i is the decoding error probability, and Q^{-1} denotes the inverse of the Gaussian Q function.

In a short frame structure, the frame duration is smaller than the channel coherence time, resulting in a quasi-static channel. Therefore, all symbols within a packet experience nearly identical fading conditions. Considering that decoding errors are

the primary cause of packet errors, the decoding error probability can be equated with the packet error probability [Durisi et al., 2016]. As a result, throughout the remainder of this thesis, we can safely replace the decoding error probability ϵ_i with the packet error probability p_i in our equations.

The transmit power $W_{\text{tx},i}$ of node i can be extracted from Eq. (2.5) by substituting R_i with $\frac{L_i}{m_i}$ as [Ali et al., 2024]

$$W_{\text{tx},i} = C_{i1}\left[\exp\left(\frac{Q^{-1}(p_i)}{\sqrt{m_i}} + \frac{\ln(2)L_i}{m_i}\right) - 1\right], \tag{2.6}$$

where $C_{i1} = \frac{\sigma^2}{|g_i|}$ and the channel dispersion is approximated as $V_i \approx 1, \forall i \in \mathcal{N}$. While the unit dispersion assumption holds exactly in the high SNR regime, it is also commonly employed in the medium SNR range under practical blocklengths to simplify analysis. This approximation enables tractable problem formulation without significantly compromising accuracy, as supported in prior studies [Ren et al., 2019, You et al., 2023, Yang et al., 2014]. By substituting $d_i(m_i)$ with $\frac{m_i}{B}$, Eq. (2.4) can be rewritten as

$$W_i(h_i, d_i(m_i), p_i) = \frac{C_{i1}\, m_i}{h_i B}\left[\exp\left(\frac{Q^{-1}(p_i)}{\sqrt{m_i}} + \frac{\ln(2)L_i}{m_i}\right) - 1\right] + \frac{W_i^c m_i}{h_i B}. \tag{2.7}$$

2.3 Schedulability

The schedulability constraint manages the assignment of transmission times to multiple sensor nodes in a network, taking into account wireless transmission limitations. We assume that no two nodes can transmit simultaneously. The transmissions are scheduled using the Earliest Deadline First (EDF) algorithm, a dynamic scheduling method that prioritizes transmissions based on their proximity to deadlines [Dertouzos, 1974]. We define the schedulability constraint as proposed by [Sadi et al., 2014]

$$\sum_{i=1}^{N} \frac{d_i(m_i)}{h_i} \leq \beta, \tag{2.8}$$

where β is the utilization bound, constrained by $0 < \beta \leq 1$. Each node i is allocated a transmission fraction of $\frac{d_i(m_i)}{h_i}$. Since nodes cannot transmit simultaneously, the sum of these fractions must fit within the total schedule length. Therefore, β cannot exceed one.

Chapter 3

JOINT OPTIMIZATION OF CONTROL AND COMMUNICATION SYSTEMS

In this chapter, we formulate the joint optimization problem for ultra-reliable WNCSs in the finite blocklength regime. The objective is to minimize the power consumption of the communication system while adhering to constraints on maximum blocklength, peak AoI violation probability, maximum transmit power, and schedulability. The problem is formulated as follows:

$$\min_{h_i, m_i, p_i, i \in [1,N]} \quad \sum_{i=1}^{N} W_i(h_i, d_i(m_i), p_i) \tag{3.1a}$$

s.t.

$$m_i \leq M_{\text{th}}, \quad \forall i \in [1, N] \tag{3.1b}$$

$$\left\lfloor \frac{\alpha - \frac{m_i}{B}}{h_i} \right\rfloor \ln p_i - \ln(1-\delta) \leq 0, \quad \forall i \in [1, N] \tag{3.1c}$$

$$0 < h_i \leq \left(\alpha - \frac{m_i}{B}\right), \quad \forall i \in [1, N] \tag{3.1d}$$

$$0 < p_i < 1, \quad \forall i \in [1, N] \tag{3.1e}$$

$$W_{\text{tx},i} \leq W_{\text{tx}}^{\max}, \quad \forall i \in [1, N] \tag{3.1f}$$

$$\sum_{i=1}^{N} \frac{d_i(m_i)}{h_i} \leq \beta. \tag{3.1g}$$

The decision variables of the optimization problem are the sampling period h_i, packet error probability p_i, and blocklength $m_i, i \in [1, N]$. Eq. (3.1a) gives the objective of the problem for the minimization of the total power consumption, as

derived in Eq. (2.7). Eq. (3.1b) represents the maximum blocklength limit, which specifies the maximum number of symbols or channel uses allowed for packet transmission. Eq. (3.1c) represents the PAoI violation probability constraint. Additionally, Eq. (3.1d) states the constraints on sampling periods required to ensure the freshness of data updates. Eq. (3.1e) provides the limits on error probability. Eq. (3.1f) is the maximum transmit power constraint to prevent excessive energy consumption, minimize interference with other devices, and comply with regulatory limits. Finally, Eq. (3.1g) represents the schedulability constraint.

This optimization problem is a non-convex Mixed-Integer Programming problem; thus, any solution for global optimum is difficult [Boyd and Vandenberghe, 2004].

Chapter 4

OPTIMIZATION THEORY BASED SAFE DRL

Optimization theory-based safe DRL combines domain-specific optimization theory formulation, data-driven deep learning techniques, and a teacher-student architecture that guarantees a secure exploration. This hybrid approach minimizes the amount of training data required in model-agnostic DRL while safely exploring the state space. The approach is divided into two stages: 1) Optimization Theory Stage: Focuses on simplifying the optimization problem using optimality conditions and domain knowledge from communication systems. 2) Safe DRL stage: Involves training the deep learning model with insights from student experiences and optimized teacher advice. The details of these stages are provided in Sections 4.1 and 4.2.

4.1 Optimization Theory Stage

In the optimization theory stage, we apply optimality conditions for the sampling period, packet error probability, and the blocklength, following the approach outlined in [Ali et al., 2024]. This analysis allows us to focus on the blocklength, m_i, and determine the other variables, i.e., sampling period and packet error probability, by using these conditions.

In Lemma 1, we relate the optimal sampling period and packet error probability to the PAoI violation probability constraint.

Lemma 1. *Let h_i^*, p_i^*, and m_i^* denote the optimal values of the sampling period, packet error probability, and blocklength, respectively. The relationship between the optimal sampling period and packet error probability is expressed as*

$$\frac{\alpha - \frac{m_i}{B}}{h_i^*} = \frac{\ln(1-\delta)}{\ln p_i^*} = k_i^*, \tag{4.1}$$

where k_i^* is a positive integer.

Proof. We prove the Lemma by contradiction, following the methods in [Ali et al., 2024] and [Sadi et al., 2014]. We start by assuming that $\frac{\alpha-\frac{m_i}{B}}{h_i^*}$ is not a positive integer such that $\left\lfloor \frac{\alpha-\frac{m_i}{B}}{h_i^*} \right\rfloor < \frac{\alpha-\frac{m_i}{B}}{h_i^*}$. If we increase h_i^* to make $\left\lfloor \frac{\alpha-\frac{m_i}{B}}{h_i^*} \right\rfloor = \frac{\alpha-\frac{m_i}{B}}{h_i^*}$, while ensuring the upper bound in Eq. (3.1d) is not violated, the constraint in Eq. (3.1c) remains satisfied, as $\left\lfloor \frac{\alpha-\frac{m_i}{B}}{h_i^*} \right\rfloor$ does not change, and the constraint (3.1g) is still met. However, the objective function in Eq. (3.1a) decreases since it is a monotonically decreasing function of h_i. Thus, keeping the inequality would contradict the optimality of h_i^*, proving that the equality must hold.

Similarly, we can prove $\frac{\alpha-\frac{m_i}{B}}{h_i^*} = \frac{\ln(1-\delta)}{\ln p_i^*}$ by contradiction. Suppose $\frac{\alpha-\frac{m_i}{B}}{h_i^*} > \frac{\ln(1-\delta)}{\ln p_i^*}$. By increasing p_i^* such that $\frac{\alpha-\frac{m_i}{B}}{h_i^*} = \frac{\ln(1-\delta)}{\ln p_i^*}$, the constraint in Eq. (3.1f) still holds since the power consumption of node i is a non-increasing function of p_i. Nonetheless, the power consumption function in Eq. (3.1a) decreases since it is a monotonically decreasing function of p_i. Since reducing power consumption is the optimization goal, allowing this inequality would imply a suboptimal choice of p_i^*, confirming that the equality must hold in the optimal solution. □

Lemma 2. *The optimal value of k_i in Lemma 1, denoted as k_i^*, is expressed as a function of m_i and given by*

$$k_i^* = \max \left[1, \left\lfloor \frac{\ln(1-\delta)}{\ln\left[Q\left[\sqrt{m_i}\ln\left(\frac{W_{tx}^{max}}{m_i C_{i1}}+1\right)-\frac{\ln(2)L_i}{\sqrt{m_i}}\right]\right]} \right\rfloor\right]. \tag{4.2}$$

Proof. The power consumption in Eq. (3.1a) increases monotonically with k_i^*. Therefore, k_i^* must be the smallest positive integer that satisfies the constraints (3.1b)-(3.1g). In Eq. (4.2), the second term of the maximum function represents the minimum value of k_i obtained in Lemma 1. □

By using (4.1), the transmit power can be rewritten as

$$W_{\text{tx},i}^* = C_{i1}\left[\exp\left(\frac{Q^{-1}((1-\delta)^{1/k_i^*})}{\sqrt{m_i}}+\frac{\ln(2)L_i}{m_i}\right)-1\right], \tag{4.3}$$

while the schedulability constraint can be reformulated as

$$\sum_{i=1}^{N} \frac{m_i k_i^*}{B\alpha - m_i} \leq \beta. \tag{4.4}$$

Also, Eq. (2.7) is updated as

$$\begin{aligned} W_i^*(m_i) = \frac{C_{i1}\, m_i k_i^*}{B\alpha - m_i} \left[\exp\left(\frac{Q^{-1}(1-\delta)^{\frac{1}{k_i^*}}}{\sqrt{m_i}} + \frac{\ln(2) L_i}{m_i} \right) - 1 \right] \\ + \frac{W_i^c m_i k_i^*}{B\alpha - m_i}, \end{aligned} \tag{4.5}$$

The joint optimization problem (3.1) can then be reformulated as

$$\min_{m_i, i \in [1,N]} \quad \sum_{i=1}^{N} W_i^*(m_i) \tag{4.6a}$$

s.t.

$$m_i \leq M_{\text{th}}, \quad \forall i \in [1, N] \tag{4.6b}$$

$$W_{\text{tx},i}^* \leq W_{\text{tx}}^{\max}, \quad \forall i \in [1, N] \tag{4.6c}$$

$$\sum_{i=1}^{N} \frac{m_i k_i^*}{B\alpha - m_i} \leq \beta. \tag{4.6d}$$

4.2 Safe DRL Stage

Now that the problem is simplified, we propose a safe deep reinforcement learning based solution to solve the non-convex optimization problem. After solving the simplified version, the optimal packet error probability and sampling period can be obtained by utilizing the optimality conditions obtained in Section 4.1.

We now provide an overview of safe DRL, the teacher-student control mechanism, and its integration with the DRL model.

4.2.1 Overview of Safe DRL

RL is a machine learning approach where an agent learns to make decisions by interacting with an environment, aiming to maximize cumulative rewards over time. The agent selects actions based on the current state of the environment, and these actions affect both immediate rewards and future states. RL methods, such as Q-learning and DQN, allow the agent to learn optimal strategies through trial and error, which often involves exploring suboptimal or unsafe actions during the learning phase. In many practical applications, such as autonomous driving or wireless communications, it is crucial to prevent unsafe behaviors, especially during the learning process.

Safe RL addresses this challenge by ensuring that the agent operates within predefined safety constraints. Various approaches are used to enforce these constraints, such as modifying the reward function to penalize constraint violations, extending Markov Decision Processes (MDPs) to include safety constraints, and using teacher-student frameworks where the teacher oversees the agent's actions, ensuring that unsafe actions are corrected in real-time. In other cases, Lyapunov-based methods are employed to guarantee stability and safety throughout the learning process. Safe RL is critical in scenarios where violating constraints can lead to costly or dangerous outcomes, ensuring that the agent can still explore and learn effectively without compromising safety.

In the context of URLLC systems, safe RL plays a vital role due to the strict requirements on latency, reliability, and power consumption. Traditional RL methods, which lack built-in safety mechanisms, may result in the exploration of unsafe actions, potentially violating these stringent constraints. Safe RL, however, ensures that the agent can optimize key performance metrics, such as minimizing latency or power consumption, while consistently adhering to reliability and latency thresholds. This ability to balance optimization with guaranteed adherence to system constraints makes safe RL indispensable for meeting the real-time, mission-critical demands of URLLC applications, where even minor deviations can have significant impacts on service quality and system performance.

4.2.2 Teacher-Student Control Mechanism

a) Teacher Role in Ensuring Safety: The teacher in the teacher-student framework plays a critical role in ensuring that the agent's actions remain within predefined safety constraints. The teacher continuously monitors the student's decisions and provides real-time guidance to prevent constraint violations. The following are key stages where constraint violations may occur, along with the mechanisms used by the teacher to mitigate risks and why these interventions are essential:

- *Random Experience Accumulation Phase:* In the early phase of random experience accumulation, the agent takes exploratory actions without any prior knowledge, significantly increasing the risk of constraint violations. Since these actions are chosen randomly, they may breach critical limits on latency or packet error rates. To prevent this, the teacher continuously monitors the agent's actions and intervenes when necessary by adjusting them to the closest feasible option that adheres to all predefined safety constraints. This ensures that the experience stored in the replay buffer is not only valuable for effective learning and future decision-making but also safe.

- *Exploration in Training Phase:* During training, the agent employs an epsilon-greedy policy to balance exploration and exploitation. While this strategy facilitates effective learning, the exploration phase (triggered with probability epsilon) can still result in the selection of unsafe actions that violate key constraints. To mitigate this, the teacher carefully evaluates each action proposed under the epsilon-greedy policy and intervenes when needed, adjusting any unsafe choices to ensure they remain within the predefined safety boundaries. This guidance helps the agent maintain compliance with constraints, promoting safer exploration throughout the learning process.

- *Exploitation in Training Phase:* The agent's decision-making, guided by learned Q-values, can occasionally recommend actions that fail to meet safety requirements, often due to incomplete learning or suboptimal policy updates. To

prevent this, the teacher continuously evaluates the agent's suggested actions, intervening whenever a potential violation occurs. This ensures that the learning process does not reinforce unsafe behavior, maintaining a stable and reliable learning environment.

- *Testing Phase:* During the testing phase, the agent executes actions without further training or exploration, relying on the learned policy to perform optimally. However, if the policy has learned suboptimal strategies that occasionally violate constraints, the teacher's role is to monitor and correct these actions in real-time. By doing so, the teacher guarantees that the agent's performance remains within safety boundaries, even in the absence of explicit penalties for violations in the reward function.

By intervening at these critical points, the teacher ensures that the agent's actions consistently adhere to the required safety constraints, preventing any violations that could compromise system performance or safety. This constant guidance not only helps maintain a safe learning environment but also accelerates the convergence of the agent's policy towards optimal, safe solutions.

b) State, Action, and Reward Design: The key components of the teacher controlled DRL are outlined below.

- *States:* The state space for each agent in the system represents factors that directly influence the environment, with each agent's next state being affected by the actions of all agents collectively. The state for agent i at time t is defined as

$$s_i^{(t)} = \left(m^{(t-1)}, R_i^{(t-1)}, W^{(t-1)}, P^{(t-1)}, \gamma_i^{(t)}, g^{(t)} \right). \tag{4.7}$$

In this state space, $m^{(t-1)}$ is a vector containing the blocklengths selected by

each node at the previous time step, with the ith element specifically representing the blocklength chosen by node i. The variable $R_i^{(t-1)}$ denotes the data rate of node i from the prior time step. The vector $W^{(t-1)}$ includes the power levels used by each node at the previous time step, where the ith element indicates the transmit power for node i. Additionally, $P^{(t-1)}$ captures the total power consumption of all nodes during the previous time step. The SNR for node i at the current time step, $\gamma_i^{(t)} = \frac{g_i^{(t)} W_{\text{tx},i}^{(t-1)}}{\sigma^2}$, is calculated using the transmit power from the previous step and the current channel gain $g_i^{(t)}$. Finally, the state space includes $g^{(t)}$, which is a vector of the current channel gains for all nodes. This comprehensive state representation combines historical performance data with current channel conditions, enabling agents to make well-informed decisions. An MDP is created using the parameters from the previous time step, further constraining the model for the blocklength. By incorporating both SNR and channel gains, the framework dynamically adapts to real-time variations in the wireless environment.

- *Actions:* The agent i selects the blocklength $m_i^{(t)}$ for transmission. The action spaces of all agents are identical, and that is given by

$$A = \{1, 2, ..., M_{\text{th}}\}, \tag{4.8}$$

which is decided based on the maximum blocklength constraint.

- *Reward:* The reward function is a crucial element in the RL process. Each agent aims to take optimal action to maximize the total reward. The reward corresponds to the objective function of the optimization problem (4.6) and is given as follows:

$$r^{(t)} = -\sum_{i=1}^{N} W_i(m_i^{(t)}). \tag{4.9}$$

In our DRL model, the reward function does not include penalties for constraint violations because we employ a safe method that inherently prevents

any violations. As a result, the agent only explores safe actions, allowing the reward function to focus solely on optimizing the primary objective.

c) Action Advising Strategy: In its role as the control mechanism, the teacher identifies the closest action to the student's proposed action while satisfying all of the constraints similar to [Schmoeller Roza et al., 2022]. To achieve this, the teacher solves the following convex optimization problem:

$$\min_{o^{(t)}} \quad \|o^{(t)} - m^{(t)}\| \tag{4.10a}$$

s.t.

$$o_i^{(t)} \leq M_{\text{th}}, \quad \forall i \in [1, N] \tag{4.10b}$$

$$C_{i1}\left[\exp\left(\frac{Q^{-1}((1-\delta)^{1/k_i^{(t)}})}{\sqrt{o_i^{(t)}}} + \frac{\ln(2)L_i}{o_i^{(t)}}\right) - 1\right] \leq W_{\text{tx}}^{\max}, \tag{4.10c}$$

$$\sum_{i=1}^{N} \frac{o_i^{(t)} k_i^{(t)}}{B\alpha - o_i^{(t)}} \leq \beta, \tag{4.10d}$$

where $o^{(t)}$ is the current action advice vector from the teacher, with the ith element $o_i^{(t)}$ corresponding to the advised action for sensor node i; $m^{(t)}$ is the current action vector derived from the student's DRL algorithm, where the ith element $m_i^{(t)}$ reflects the student's action for sensor node i. The objective of the optimization problem (4.10a) is to minimize the Euclidean distance between the teacher's action advice vector and the student's action vector. The constraints (4.10b), (4.10c) and (4.10d) correspond to the constraints (4.6b), (4.6c) and (4.6d) in the original problem, respectively.

This optimization problem is solved using convex optimization solvers. The resulting action advice, $o^{(t)}$, is implemented by all nodes instead of the student's proposed action, thereby ensuring that the system operates safely without violating

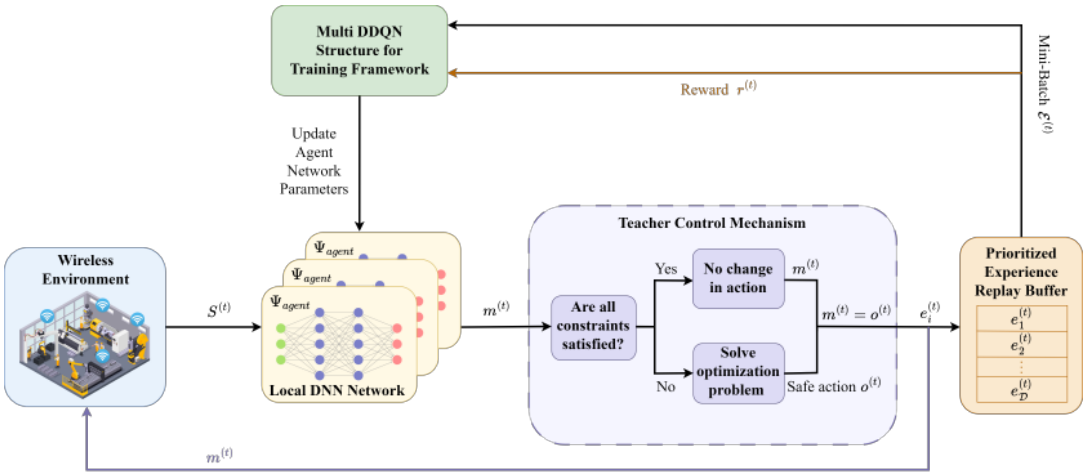

Figure 4.1: Proposed safe DRL algorithm.

any requirements. While the teacher restricts unsafe exploration, it does not limit the discovery of better policies within the feasible space, allowing the DRL agent to optimize performance while maintaining constraint compliance.

4.3 Integration with DRL Model

In the proposed safe DRL algorithm shown in Fig. 4.1, each sensor node functions as an independent agent within a multi-agent DRL architecture, where the teacher advice method is employed to ensure that system constraints are consistently satisfied. The evolution of the environment's states in this framework is determined by the collective actions of all agents. While multi-agent reinforcement learning models have demonstrated strong empirical performance, they often lack theoretical guarantees of convergence, as noted in [Nguyen et al., 2020]. To enhance both implementation efficiency and stability, a centralized training and execution approach is adopted, wherein all agents operate synchronously and select their actions simultaneously [Gupta et al., 2017].

In this setup, each agent i observes the state $s_i^{(t)}$ and chooses an action $m_i^{(t)}$ via its respective local neural network. If the action of the agent, identified as the student, satisfies all the constraints, the teacher does not interfere. However, if any constraints are violated, the teacher intervenes by suggesting the closest permissible

action $o_i^{(t)}$ that satisfies all the constraints. Then, the student action $m_i^{(t)}$ is set to $o_i^{(t)}$ to ensure compliance with the requirements. Upon executing the action, each agent receives a reward $r_i^{(t)}$, and the cumulative reward function $r^{(t)}$ across all agents is computed. At each time step t, an experience tuple is recorded as $e^{(t)} = (s^{(t)}, m^{(t)}, r^{(t)}, s^{(t+1)})$, where $s^{(t)}$ is the state vector with the ith element $s_i^{(t)}$ corresponding to the state of agent i. Subsequently, the experiences of all agents are consolidated into a Prioritized Experience Replay (PER) buffer D, as not all transitions contribute equally to learning. The buffer assigns priority to each transition based on its impact on the learning process, which is measured by the change in cumulative reward. Transitions with larger changes in reward are deemed more significant and are given higher priorities. The probability of sampling a transition is proportional to its priority, meaning transitions with higher priority are more likely to be selected for training. The degree of prioritization is controlled by a parameter that adjusts how strongly the learning process favors high-priority transitions. This approach ensures that transitions that lead to significant improvements in reward are replayed more often, thereby accelerating the learning process and enhancing overall performance [Zheng et al., 2022]. Training is then conducted using the mini-batch $\mathcal{E}^{(t)}$ sampled from this buffer to update the network parameters.

The Q-learning algorithm seeks to discover a policy that maximizes the cumulative future reward. The Q-function represents the expected future reward for taking a specific action under policy π, expressed as

$$Q^{\pi}(s, m) = \mathbb{E}_{\pi}[R^{(t)} | s^{(t)} = s, m^{(t)} = m], \tag{4.11}$$

where $R^{(t)}$ denotes the discounted future reward, calculated as

$$R^{(t)} = \sum_{\tau=0}^{\infty} \alpha_D^{\tau} r^{(t+\tau)}, \tag{4.12}$$

with $\alpha_D \in (0, 1]$ being the discount factor. In conventional Q-learning, a lookup table of Q-values is created, and the agent selects actions according to the ϵ-greedy

policy. Initially, Q-values are randomly assigned, and the agent chooses actions based on this policy at each time step t. The ϵ-greedy strategy allows the agent to either exploit the best-known action with probability $1 - \epsilon$, or explore by selecting a random action with probability ϵ. To encourage more exploitation over time, the value of ϵ is progressively decreased using a decaying ϵ-greedy algorithm where $\epsilon^{(t+1)} = (1 - \alpha_\epsilon)^{(t)}\epsilon^{(0)}$, where α_ϵ is the decay factor.

A DQN improves upon traditional Q-learning by using deep neural networks to approximate the Q-value function, enabling it to handle high-dimensional state spaces. It incorporates a replay buffer, which stores past experiences for random sampling during training to reduce data correlation, and a fixed target network, which keeps target Q-values stable by periodically updating its parameters, ensuring more reliable learning. However, DQN often overestimates Q-values, leading to instability during training. To address this, Double Deep Q-learning (DDQN) was introduced, which decouples action selection and evaluation by using an online network for selecting actions and a target network for evaluation. Furthermore, the Dueling Q-network architecture enhances learning by separately estimating the state value and the advantage function, enabling the network to identify valuable states irrespective of the chosen action. Combining these approaches, the Dueling Double Deep Q-Network (D3QN) improves both stability and performance by addressing overestimation issues and enabling more efficient learning [Hessel et al., 2018].

In this work, a D3QN is employed, comprising two DQNs: the target network and the training network, with parameters $\theta^{(t)}_{\text{target}}$ and $\theta^{(t)}_{\text{train}}$, respectively. The target network parameters $\theta^{(t)}_{\text{target}}$ are updated using a soft update method every time frame, such that

$$\theta^{(t)}_{\text{target}} = \alpha_S \theta^{(t)}_{\text{train}} + (1 - \alpha_S)\theta^{(t)}_{\text{target}}, \tag{4.13}$$

where $\alpha_S \ll 1$ is a smoothing factor ensuring greater stability in the learning process. The least squares loss is calculated from a mini-batch of sampled transitions $E^{(t)}$

using

$$\mathcal{L}(\theta_{\text{train}}^{(t)}) = \sum_{(s,m,r,s')\in\mathcal{E}^{(t)}} (y_{\text{DQN}}^{(t)}(r,s') - Q(s,m;\theta_{\text{train}}^{(t)}))^2, \tag{4.14}$$

where

$$y_{\text{DQN}}^{(t)}(r,s') = r + \alpha_T \max_{m'} Q(s',m';\theta_{\text{target}}^{(t)}), \tag{4.15}$$

with $\alpha_T \in (0,1)$ as the discount factor.

The agent minimizes the loss function by adjusting the training network parameters $\theta_{\text{train}}^{(t)}$ using stochastic gradient descent, selected for its rapid convergence time [LeCun et al., 2015]. The gradient update is performed as

$$\theta_{\text{train}}^{(t)} \leftarrow \theta_{\text{train}}^{(t)} - \lambda^{(t)} \nabla_{\theta_{\text{train}}^{(t)}} \mathcal{L}(\theta_{\text{train}}^{(t)}), \tag{4.16}$$

where $\lambda^{(t)} \in (0,1)$ represents the learning rate. To ensure a more stable learning process, the learning rate is progressively decreased using $\lambda^{(t+1)} = (1-\alpha_L)\lambda^{(t)}$, where α_L is the learning rate decay factor.

At the end of each time frame, the updated D3QN parameters $\theta_{i,\text{train}}^{(t)}$ are synchronized with their corresponding local D3QN models, updating the local weights $\Psi_{i,\text{agent}}^{(t)}$.

The overall algorithm for the proposed safe multi-agent D3QN-based resource allocation framework is summarized in Algorithm 1, detailing the initialization, experience collection, and training-execution phases. In the initialization phase, the experience replay buffer $\mathcal{D}$ is created (Line 1). Each node $i \in \mathcal{N}$ initializes its local, train, and target D3QN networks with randomly assigned weights (Lines 2–4). During the experience collection phase, each node observes its local state and sends it to the central server (Line 7). The server selects a random blocklength, which is verified or modified by the teacher (Line 8). The resulting next state is observed, and the experience is stored in $\mathcal{D}$ (Line 9). Once sufficient data is available, a batch is sampled to compute the loss and update the train and target networks (Lines

11–16). In the training-execution phase, each node again observes its state and communicates it to the server (Line 20). An action is selected using the ϵ-greedy policy and then validated or adjusted by the teacher (Lines 21–22). The resulting experience is stored (Line 23). The networks are updated as in the training phase. Finally, the verified actions are executed (Lines 25–29).

Algorithm 1 Safe D3QN-Based Algorithm

Initialization:

Initialize replay buffer $\mathcal{D}$.
for each node $i \in \mathcal{N}$ **do**
    Initialize local, train, and target networks:
    $\mathcal{Q}(s_i^{(t)}, m_i^{(t)}; \Psi_{i,\text{agent}}^{(t)})$,
    $\mathcal{Q}(s_i^{(t)}, m_i^{(t)}; \theta_{i,\text{train}}^{(t)})$,
    $\mathcal{Q}(s_i^{(t)}, m_i^{(t)}; \theta_{i,\text{target}}^{(t)})$.
end for

Experience Collection and Replay:

for each time frame t **do**
    for each node $i \in \mathcal{N}$ **do**
        Observe state $s_i^{(t)}$, transmit to server.
        Server selects $m_i^{(t)}$; teacher modifies if needed.
        Observe next state $s_i^{(t+1)}$, and store $e_i^{(t)}$ in $\mathcal{D}$.
    end for
    if $|\mathcal{D}| \geq \mathcal{E}$ **then**
        Sample batch $\mathcal{E}^{(t)}$ using prioritized replay.
        Compute D3QN loss (4.14), update $\theta_{i,\text{train}}^{(t)}$ (4.16).
        Soft update $\theta_{i,\text{target}}^{(t)}$ via (4.13).
        Broadcast $\theta_{i,\text{train}}^{(t)}$ to update $\Psi_{i,\text{agent}}^{(t)}$.
    end if
end for

Training and Execution:

for each t **do**
    for each $i \in \mathcal{N}$ **do**
        Observe state $s_i^{(t)}$, transmit to server.
        Select action $m_i^{(t)}$ via ϵ-greedy.
        Teacher verifies $m_i^{(t)}$, modifies if needed.
        Observe next state $s_i^{(t+1)}$, and store $e_i^{(t)}$ in $\mathcal{D}$.
    end for
    Sample $\mathcal{E}^{(t)}$, compute loss (4.14).
    Update $\theta_{i,\text{train}}^{(t)}$ (4.16).
    Soft update $\theta_{i,\text{target}}^{(t)}$ (4.13).
    Broadcast $\theta_{i,\text{train}}^{(t)}$ to update $\Psi_{i,\text{agent}}^{(t)}$.
    Execute the verified actions.
end for

Chapter 5

PERFORMANCE EVALUATION

In this chapter, we analyze the performance of the proposed optimization theory-based safe DRL algorithm in comparison to rule-based DRL and optimization theory-based DRL benchmarks. The rule-based DRL benchmark serves as a safe RL baseline, where system constraints are enforced through predefined rules. While the reference approach in [Nguyen et al., 2022] directly encodes expert-defined rules for action selection, we adapt this concept by generating actions randomly and repeating until the selected action satisfies all constraints. The optimization theory-based DRL benchmarks employ D3QN and DDQN models without any integrated safe mechanisms.

5.1 Simulation Setup

Simulations are run for a network comprising uniformly distributed nodes that are in communication with a central controller inside a 50 m radius circle. Both large-scale and small-scale fading affect the connections between the sensor nodes and the gateway. The large-scale fading ζ_i happens due to path loss and shadowing effects caused by the obstacles between the transmitter and the receiver. It is modeled as follows: $PL(d) = PL(d_0) + 10\zeta \log(\frac{d_i}{d_0}) + Z$ dB, where d_i is the node's distance from the central controller, $PL(d_i)$ is its path loss at d_i from the controller, measured in decibels, $PL(d_0) = 35.3$ dB is its path loss at reference distance $d_0 = 1$ m, and $\zeta = 3.76$ is the path loss exponent [Access, 2010]. Z is the log-normal shadowing corresponding to a Gaussian random variable with zero mean and standard deviation equal to 4 dB. Jake's model, which is represented as a first-order complex Gauss-

Markov process, is used for the small-scale fading:

$$f_i^{(t)} = \rho f_i^{(t-1)} + \sqrt{1-\rho^2}\, e_i^{(t)}, \tag{5.1}$$

where ρ is the correlation coefficient, which is set to 0.6. Also, $f_i^{(t)}$ and $e_i^{(t)}$ are the channel coefficient and the channel innovation process of node i at time t, respectively. $f_i^{(0)}$ and $e_i^{(1)}, e_i^{(2)}, ...$ are independent and identically distributed circularly symmetric complex Gaussian (CSCG) random variables with unit variance. Accordingly, the channel gains are computed by

$$g_i^{(t)} = |f_i^{(t)}|^2 \zeta_i, \quad t = 1, 2, ... \tag{5.2}$$

The noise power spectral σ^2 is set to -174 dBm/Hz. The other parameters used in the simulations are given in Table 5.1. The simulation parameters are selected in accordance with typical URLLC settings and prior works on short-packet communication and finite blocklength analysis [Yates et al., 2021, Popovski et al., 2022, Ali et al., 2024, Sadi et al., 2014]. Although the packet length L_i is fixed for all nodes, the resulting blocklengths m_i vary by node. This variation arises from differences in wireless channel gains, sampling intervals, and packet error probabilities, all jointly considered in the optimization framework.

Table 5.1: Simulation Parameters

Parameter	**Value**	**Parameter**	**Value**
B	100 kHz	M_{th}	200 Symbols
L_i	100 bits	δ	0.99
α	101 ms	N	50
$W_{\max}$	250 mW	W_c	5 mW
β	0.9	σ^2	-174 dBm/Hz

The network architecture consists of one input layer, three hidden layers (N_1, N_2, N_3), and one output layer. The vanishing gradient issue is avoided by using the leaky ReLU activation function in the hidden layers. For Teacher-Student, Rule-Based, and D3QN algorithms, the final layer outputs the advantage and state

Table 5.2: Hyperparameters

Layers	**Input**	N_1	N_2	N_3	**Output**
Teacher-Student Rule-Based D3QN	$3N+3$	32	64	300	$M_{\text{th}}, 1$
DDQN	$3N+3$	32	64	300	M_{th}
Activation Function	Linear	ReLU	ReLU	ReLU	Linear
Mini-Batch Size	64				
Testing	10 simulations × 2,500 episodes				
Discount Factor α_T	0.666				
Soft Update Rate α_S	10^{-3}				
Initial Learning Rate $\lambda^{(0)}$	0.03				
Learning Decay Rate α_L	10^{-3}				
Initial Exploration Rate $\epsilon^{(0)}$	1				
ϵ-Decay Rate α_ϵ	10^{-4}				

value functions, featuring N_A neurons for the advantage function, corresponding to M_{th}, and N_V neurons for the state value function, which is set to 1. For the DDQN benchmark, the output is equal to the number of possible actions, which is M_{th}. The mini-batch size is set to 64 for all algorithms. The optimal hyperparameters are determined by employing the grid search algorithm, which systematically examines a grid of all potential combinations of hyperparameter values. The selected hyperparameters are tabulated in Table 5.2. The proposed algorithm is implemented using PyTorch [Paszke et al., 2019]. Each test result is averaged over 10 simulations, with each simulation initialized using a unique random seed and spanning 2,500 episodes.

5.2 Performance Comparison of Algorithms

Fig. 5.1(a) shows the training convergence of the reward for different algorithms in a network of 50 nodes with $\alpha = 101$ ms. The proposed algorithm demonstrates significantly superior performance compared to benchmark algorithms, showing faster convergence, higher rewards, and greater stability. This is due to its ability to intelligently select actions that satisfy system constraints, leading to higher rewards. D3QN performs slightly better than DDQN in terms of average reward, benefiting from its robust architecture and capability to identify patterns in a stochastic envi-

ronment. On the other hand, the rule-based D3QN exhibits instability as it resorts to random actions whenever a rule is violated. Fig. 5.1(b) presents the empirical cumulative distribution function (CDF) of the total power consumption (in Watts) obtained during the testing phase using the parameters saved at the end of training. The x-axis indicates the total power consumed per episode, while the y-axis represents the corresponding cumulative probability. This visualization enables a direct comparison of the energy efficiency of the methods by illustrating how frequently each achieves lower power usage. The results align closely with the outcomes observed during training. The teacher-student method demonstrates lower power consumption compared to the benchmark algorithms, with its primary advantage being strict compliance with system constraints, an attribute not observed in the D3QN, DDQN, and random approaches. On the other hand, rule-based, D3QN, and DDQN methods exhibit similar power consumption distributions. Finally, the random approach shows the highest power consumption, highlighting its inefficiency relative to the other methods.

The power violation plot for a single node over 50 episodes is shown in Fig. 5.1(c). Both the teacher-student and rule-based algorithms consistently satisfy the power constraint for all nodes. Although the rule-based D3QN guarantees constraint satisfaction, it exhibits high variance and instability due to its random action regeneration strategy, which becomes ineffective in complex real-time scenarios. In contrast, the D3QN, DDQN, and random benchmarks repeatedly breach the power threshold for this node. To further evaluate the power violation performance across different algorithms, two metrics are used: the episode violation rate, which indicates the percentage of episodes where any node breaches the constraint, and the average node violation rate, which provides a granular view of constraint adherence across all nodes. For the D3QN algorithm, the episode violation rate is 80.7%, with an average node violation rate of 1.72%. Similarly, the DDQN algorithm records an episode violation rate of 83.9% and an average node violation rate of 1.68%. In contrast, the random approach shows significantly lower values, with an episode violation rate of 4.08% and an average node violation rate of only 0.093%. The high episode violation rates observed in the D3QN and DDQN algorithms primarily

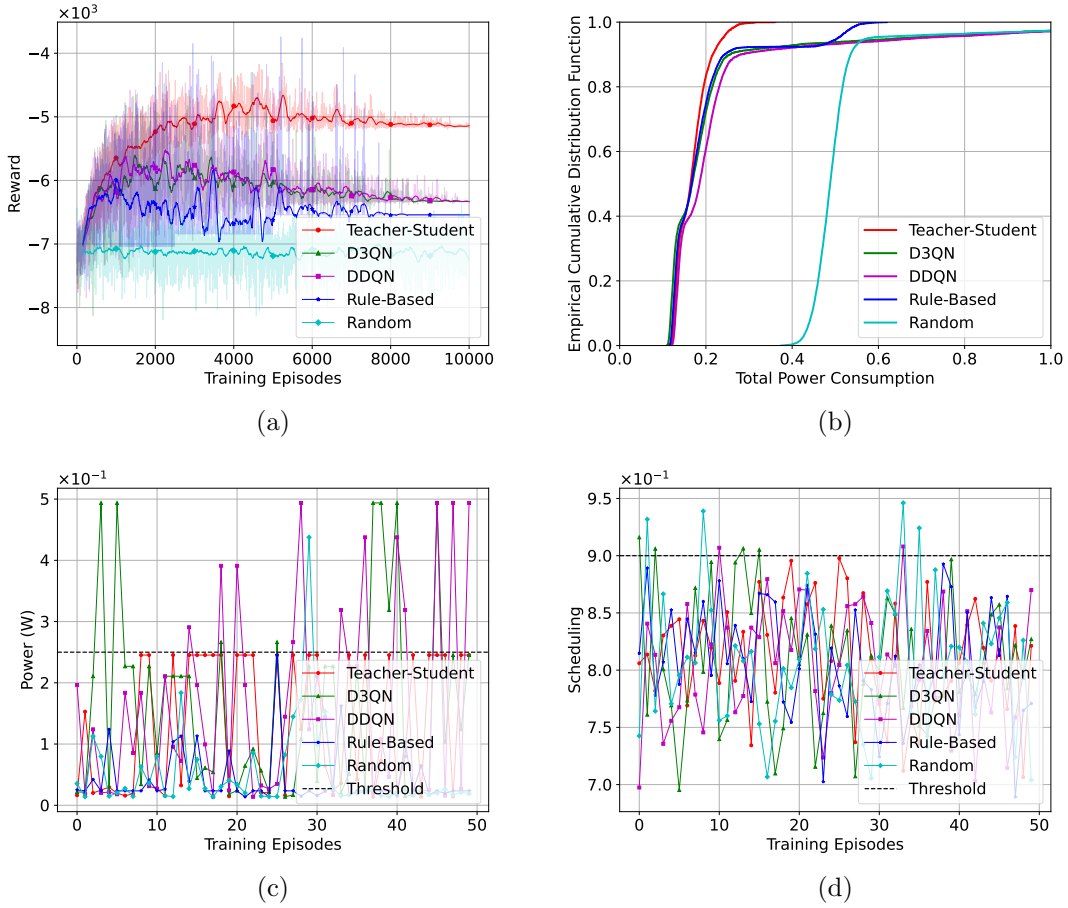

Figure 5.1: (a) Training, (b) testing, (c) power violation, and (d) scheduling violation results for different algorithms in a network of 50 nodes with $\alpha = 101$ ms.

result from their reward design, which prioritizes minimizing the objective function over satisfying power constraints for certain nodes. In DRL frameworks, as the reward is defined as the sum of the objective and penalties for constraint violations, the agent may favor actions that optimize the objective while accepting penalties for breaches. This trade-off often causes the agent to prioritize the objective at the expense of strict adherence to the constraints. The scheduling violation plot is shown in Fig. 5.1(d). Similarly, the teacher-student and rule-based algorithms satisfy the constraint in all of the episodes. However, the D3QN, DDQN, and random benchmarks exceed the threshold, with violation rates of 0.09%, 0.1%, and 4.62%, respectively. The relatively low violation probabilities of D3QN and DDQN reflect the less stringent nature of the scheduling constraint in this setup. It is important

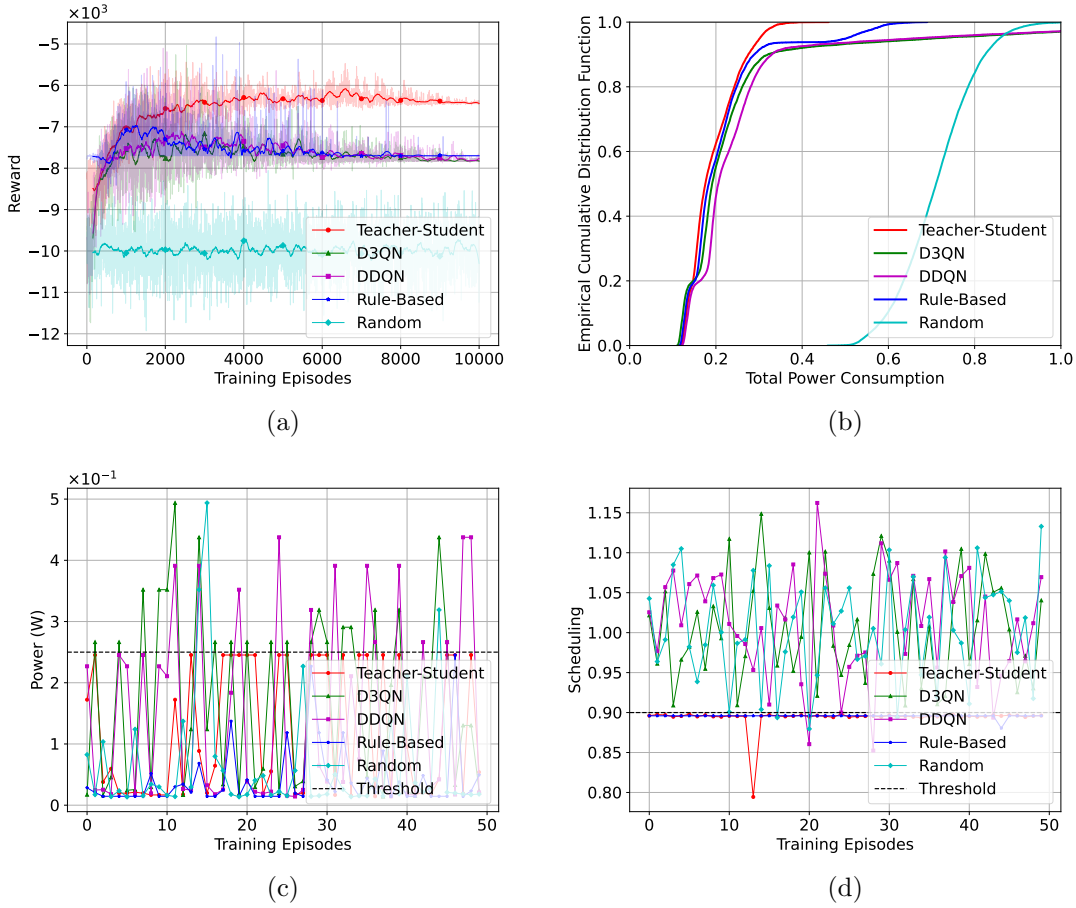

Figure 5.2: (a) Training, (b) testing, (c) power violation, and (d) scheduling violation results for different algorithms in a network of 50 nodes with $\alpha = 81$ ms.

to note that the PAoI violation probability constraint is inherently satisfied across all methods, as it is enforced in the optimization theory stage. Since the optimality conditions provided in Lemmas 1 and 2 are applied to every approach, none of the tested methods violates PAoI constraints during execution. In contrast, a pure DRL approach without optimization theory would struggle to maintain this guarantee, as it lacks an explicit mechanism for enforcing constraint satisfaction. This issue was observed in [Ali et al., 2024], where pure DRL exhibited worse performance due to frequent violations. As a result, we did not include a pure DRL baseline for comparison, as it would not provide a fair evaluation under strict PAoI constraints.

5.2.1 Impact of PAoI

This section examines the effect of varying the PAoI threshold value, α, on the performance of the algorithms during the training and testing phases. Fig. 5.2(a) shows the training convergence of the reward for different algorithms in a network of 50 nodes with $\alpha = 81$ ms. Compared to the earlier scenario, the stricter PAoI violation probability and scheduling constraints result in higher penalties. Consequently, the rule-based algorithm achieves better performance than the DDQN, DQN, and random benchmarks. However, the proposed teacher-student approach outperforms all algorithms in terms of both convergence and average reward, thanks to its logical and safe decision-making mechanism. Fig. 5.2(b) shows the testing phase results for the algorithms. Under the strict PAoI condition, the teacher-student and rule-based approaches significantly outperform the DDQN, DQN, and random methods, as meeting the requirements becomes both more challenging and more rewarding. This result aligns with the observations from the training convergence graph. As anticipated, the stringent PAoI constraint also leads to an increase in total power consumption to ensure successful transmissions. Overall, the proposed teacher-student method once again demonstrates superior performance during the testing phase, surpassing all other algorithms.

Fig. 5.2(c) and Fig. 5.2(d) indicate that D3QN, DDQN, and random approaches frequently violate power and scheduling constraints, exceeding the specified thresholds. Specifically, the episode violation percentages are 87.4%, 87.1%, and 4.2% for the D3QN, DDQN, and random approaches, respectively, with average node violation rates of 1.75%, 1.74%, and 0.089%. Both D3QN and DDQN often exceed the maximum transmit power limit of certain nodes to optimize overall system performance. In terms of scheduling violations, the violation percentages are 2.6%, 2.3%, and 96% for D3QN, DDQN, and random benchmarks, respectively. The random method exhibits a high violation rate due to the majority of its action combinations failing to meet the constraints. Conversely, D3QN and DDQN gradually learn the constraints over time, resulting in fewer violations as the high penalties discourage constraint breaches. Nonetheless, the rule-based and teacher-student methods con-

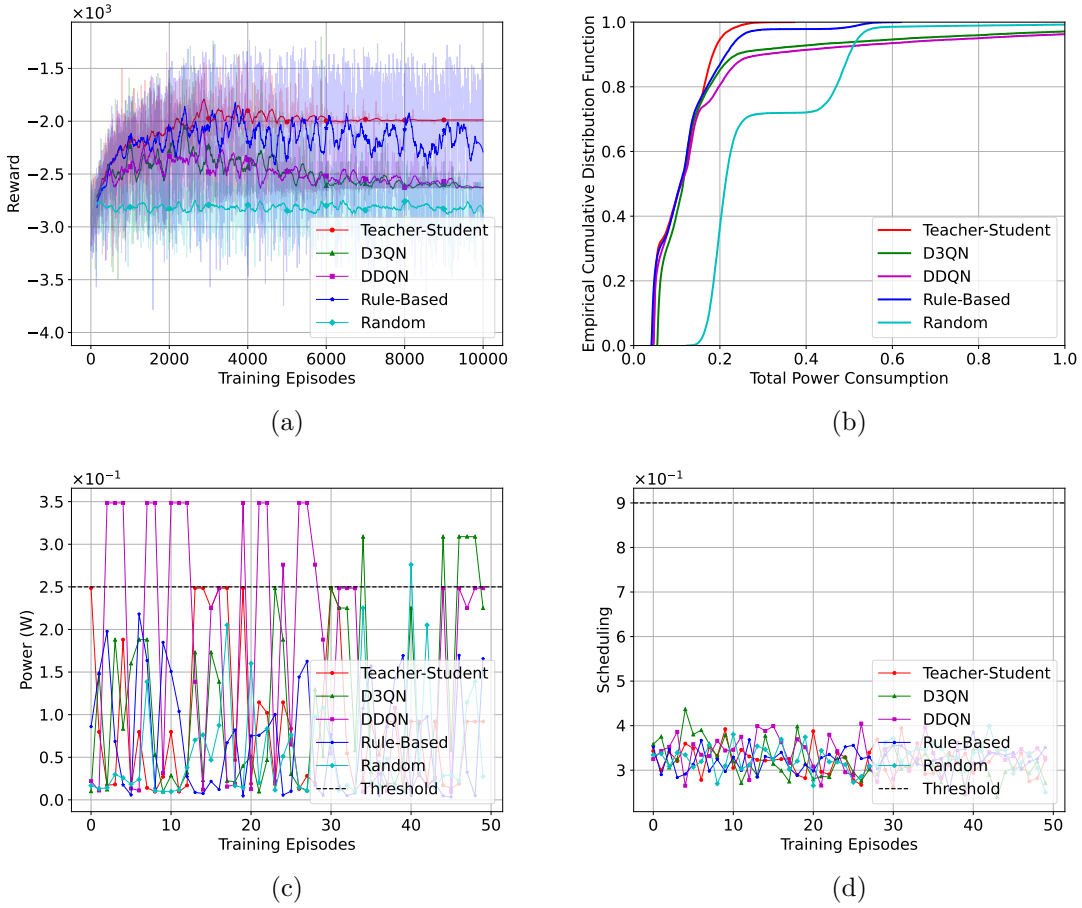

Figure 5.3: (a) Training, (b) testing, (c) power violation, and (d) scheduling violation results for different algorithms in a network of 20 nodes with $\alpha = 101$ ms.

sistently satisfy both power and scheduling requirements, even under stricter conditions, demonstrating their robustness and reliability in maintaining compliance with the increasingly challenging constraints.

5.2.2 Impact of the Number of Nodes

We analyze the effect of the number of nodes on the performance of the proposed algorithms. Fig. 5.3(a) shows the training convergence of the reward for a network of 20 nodes with $\alpha = 101$ ms. As in previous setups, the proposed teacher-student framework outperforms others in terms of stability, convergence, and average reward. However, total power consumption results display greater fluctuations as the number of nodes decreases. This is because, in more relaxed scenarios, intricate approaches

like the teacher-student framework may overfit during training, hindering optimal outcomes during testing. The rule-based method shows an unstable reward curve due to its reliance on random action selection, emphasizing the importance of action advice from a control mechanism. Meanwhile, DDQN and DQN benchmarks exhibit negligible differences in performance. Fig. 5.3(b) presents the testing outcomes using the saved training parameters. Except for the random method, all algorithms demonstrate similar performance in total power consumption due to the relaxed constraints. Teacher-student, rule-based, DDQN, and DQN methods consume less total power than the random benchmark. However, the teacher-student framework achieves the best results. In general, total power consumption decreases as the number of nodes is reduced.

As shown in Fig. 5.3(c), only teacher-student and rule-based methods consistently satisfy the power constraint. In contrast, the D3QN, DDQN, and random approaches exhibit episode violation rates of 77.7%, 77.9%, and 1.75%, respectively, with corresponding average node violation rates of 3.89%, 4.01%, and 0.90%. Similar to earlier scenarios, D3QN and DDQN fail to adequately prioritize constraints in the reward design, leading to higher violation rates. The reduced number of nodes results in lower episode violation rates, as the likelihood of encountering a violation in an episode decreases. However, average node violation rates increase because each node's behavior has a more significant impact on the overall metric. With only 20 nodes, the scheduling requirement becomes more relaxed and easier to satisfy, as reflected in Fig. 5.3(d). Unlike previous cases, none of the algorithms violate the scheduling constraint due to the less stringent setup. However, D3QN, DDQN, and random approaches lack safety mechanisms, which could lead to constraint violations under different parameter settings.

5.2.3 Runtime Performance

Fig. 5.4 shows the average simulation time per iteration, measured in milliseconds, for each algorithm. The execution times of all algorithms increase nearly linearly with the number of nodes. As expected, the random approach exhibits the lowest

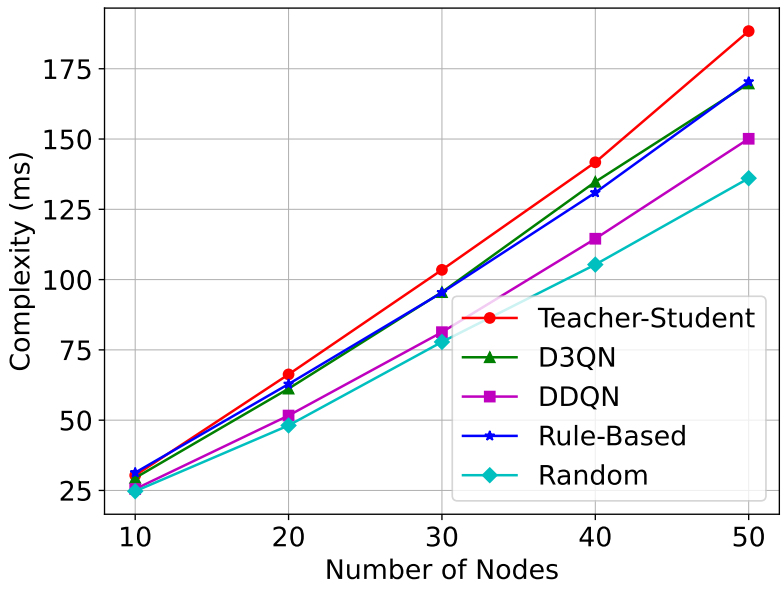

Figure 5.4: Average execution time (ms) per step for different algorithms and number of nodes.

complexity. DDQN follows with slightly higher complexity due to its architecture. D3QN and the rule-based benchmarks require more time as they introduce additional processing steps. The proposed teacher-student method incurs a marginal increase in simulation time due to its advice mechanism. However, this minor overhead is justified by its faster convergence and stable system performance, making it highly suitable for real-time applications.

Chapter 6

CONCLUSION

This thesis introduces a novel optimization theory-based safe DRL approach for the joint optimization of control and communication systems, incorporating a uniquely derived PAoI violation probability constraint. After formulating the joint optimization problem, we derive the optimality conditions to simplify and decompose the problem. The simplified problem is then solved using D3QN within a teacher-student learning framework, where the teacher guides the student by providing action advice that aligns best with the student's actions while satisfying system constraints. To benchmark our approach, we utilize rule-based D3QN, D3QN, DDQN, and random methods. The constraint violation percentages for D3QN, DDQN, and random approaches are presented for different parameter settings, showing that D3QN and DDQN exhibit similar violation rates, indicating that improvements in architecture have minimal impact on constraint violations. In contrast, the rule-based D3QN consistently satisfies the constraints but exhibits unstable performance due to its random advice mechanism. Our proposed algorithm outperforms the benchmarks across various node configurations and PAoI violation probability constraints, all while adhering to system requirements with only a negligible increase in computational complexity. The algorithm demonstrates stable performance due to the strategic advice mechanism integrated with the D3QN model's action selection behavior. As future work, we aim to enhance the algorithm by incorporating advanced learning techniques such as meta-learning and transfer learning to improve the adaptability of the safe DRL method to new environments, reducing the need for extensive retraining.

BIBLIOGRAPHY

[ANS,] Ansi/isa-100.11a-2011 wireless systems for industrial automation: Process control and related applications. `https://www.isa.org/products/ansi-isa-100-11a-2011-wireless-systems-for-industr`. Accessed: 2023-01-21.

[Access, 2010] Access, E. U. T. R. (2010). Further advancements for e-utra physical layer aspects (release 9). *Eur. Telecommun. Stand. Inst.*

[Ali et al., 2024] Ali, H. Q. et al. (2024). Optimization theory based deep reinforcement learning for resource allocation in ultra-reliable wireless networked control systems. *IEEE Trans. Commun.*, pages 1–1.

[Ashraf et al., 2022] Ashraf, K. et al. (2022). Joint optimization via deep reinforcement learning in wireless networked controlled systems. *IEEE Access*, 10:67152–67167.

[Boyd and Vandenberghe, 2004] Boyd, S. and Vandenberghe, L. (2004). *Convex Optimization*. Cambridge University Press.

[Cao et al., 2023] Cao, J., Zhu, X., Sun, S., Popovski, P., Feng, S., and Jiang, Y. (2023). Age of loop for wireless networked control system in the finite blocklength regime: Average, variance and outage probability. *IEEE Transactions on Wireless Communications*, 22(8):5306–5320.

[Champati et al., 2021] Champati, J. P. et al. (2021). Minimum achievable peak age of information under service preemptions and request delay. *IEEE J. Sel. Areas Commun.*, 39(5):1365–1379.

[Costa et al., 2016] Costa, M. et al. (2016). On the age of information in status update systems with packet management. *IEEE Trans. Inf. Theory*, 62(4):1897–1910.

[Cui et al., 2005] Cui, S., Goldsmith, A. J., and Bahai, A. (2005). Energy-constrained modulation optimization. *IEEE Trans. Wireless Commun.*, 4(5):2349–2360.

[Dertouzos, 1974] Dertouzos, M. (1974). Control robotics: The procedural control of physical processes. *IFIP Congr. 1974*.

[Devassy et al., 2019] Devassy, R. et al. (2019). Reliable transmission of short packets through queues and noisy channels under latency and peak-age violation guarantees. *IEEE J. Sel. Areas Commun.*, 37(4):721–734.

[Durisi et al., 2016] Durisi, G. et al. (2016). Toward massive, ultrareliable, and low-latency wireless communication with short packets. *Proc. IEEE*, 104(9):1711–1726.

[Gupta et al., 2017] Gupta, J. K. et al. (2017). Cooperative multi-agent control using deep reinforcement learning. In *Int. Conf. Auton. Agents Multiagent Syst.*, pages 66–83. Springer.

[Hessel et al., 2018] Hessel, M. et al. (2018). Rainbow: Combining improvements in deep reinforcement learning. In *Proc. AAAI Conf. Artif. Intell.*, volume 32.

[International Telecommunication Union – Radiocommunication Sector (ITU-R), 2013] International Telecommunication Union – Radiocommunication Sector (ITU-R) (2013). Technical Characteristics and Spectrum Requirements of Wireless Avionics Intra-Communications Systems to Support Their Safe Operation. Recommendation ITU-R M.2283-0, International Telecommunication Union, Geneva, Switzerland. Available online.

[Kartal et al., 2023] Kartal, O. T. et al. (2023). Resource allocation in the finite blocklength regime under paoi and delay violation constraints. In *2023 21st Int. Symp. Modeling Optimization Mobile, Ad Hoc, Wireless Netw. (WiOpt)*, pages 539–546.

[Kosta et al., 2021] Kosta, A. et al. (2021). The age of information in a discrete time queue: Stationary distribution and non-linear age mean analysis. *IEEE J. Sel. Areas Commun.*, 39(5):1352–1364.

[LeCun et al., 2015] LeCun, Y. et al. (2015). Deep learning. *Nature*, 521(7553):436–444.

[Nguyen et al., 2020] Nguyen, T. T. et al. (2020). Deep reinforcement learning for multiagent systems: A review of challenges, solutions, and applications. *IEEE Trans. Cybern.*, 50(9):3826–3839.

[Nguyen et al., 2022] Nguyen, T. T., Thao Nguyen, T., Nguyen, T.-H., and Nguyen, P. L. (2022). Fuzzy q-learning-based opportunistic communication for mec-enhanced vehicular crowdsensing. *IEEE Transactions on Network and Service Management*, 19(4):5021–5033.

[Park et al., 2018] Park, P. et al. (2018). Wireless network design for control systems: A survey. *IEEE Commun. Surveys Tuts.*, 20(2):978–1013.

[Paszke et al., 2019] Paszke, A. et al. (2019). Pytorch: An imperative style, high-performance deep learning library. *Adv. Neural Inf. Process. Syst.*, 32.

[Pister et al., 2009] Pister, K. et al. (2009). Industrial Routing Requirements in Low-Power and Lossy Networks. RFC 5673.

[Popovski et al., 2022] Popovski, P., Chiariotti, F., Huang, K., Kalør, A. E., Kountouris, M., Pappas, N., and Soret, B. (2022). A perspective on time toward wireless 6g. *Proceedings of the IEEE*, 110(8):1116–1146.

[Ren et al., 2019] Ren, H. et al. (2019). Joint power and blocklength optimization for urllc in a factory automation scenario. *IEEE Trans. Wireless Commun.*, 19(3):1786–1801.

[Sadi and Coleri Ergen, 2013] Sadi, Y. and Coleri Ergen, S. (2013). Optimal power control, rate adaptation, and scheduling for uwb-based intravehicular wireless sensor networks. *IEEE Trans. Veh. Technol.*, 62(1):219–234.

[Sadi and Coleri Ergen, 2017] Sadi, Y. and Coleri Ergen, S. (2017). Joint optimization of wireless network energy consumption and control system performance in wireless networked control systems. *IEEE Trans. Wireless Commun.*, 16(4):2235–2248.

[Sadi et al., 2014] Sadi, Y., Coleri Ergen, S., and Park, P. (2014). Minimum energy data transmission for wireless networked control systems. *IEEE Trans. Wireless Commun.*, 13(4):2163–2175.

[Saurav and Vaze, 2023] Saurav, K. and Vaze, R. (2023). Online energy minimization under a peak age of information constraint. *IEEE J. Sel. Areas Inf. Theory*, 4:579–590.

[Schmoeller Roza et al., 2022] Schmoeller Roza, F. et al. (2022). Safe robot navigation using constrained hierarchical reinforcement learning. In *2022 21st IEEE Int. Conf. Machine Learning Applications (ICMLA)*, pages 737–742.

[Wang et al., 2024] Wang, X., Du, H., Feng, L., Niyato, D., Zhou, F., Yang, Z., and Li, W. (2024). Resource allocation and user pairing for rate splitting multiple access based wireless networked control systems. *IEEE Transactions on Communications*, pages 1–1.

[Witrant et al., 2010] Witrant, E. et al. (2010). Limitations and performances of robust control over wsn: Ufad control in intelligent buildings. *IMA J. Math. Control Inf.*, 27(4):527–543.

[Yang et al., 2014] Yang, W., Durisi, G., Koch, T., and Polyanskiy, Y. (2014). Quasi-static multiple-antenna fading channels at finite blocklength. *IEEE Transactions on Information Theory*, 60(7):4232–4265.

[Yates et al., 2021] Yates, R. D., Sun, Y., Brown, D. R., Kaul, S. K., Modiano, E., and Ulukus, S. (2021). Age of information: An introduction and survey. *IEEE Journal on Selected Areas in Communications*, 39(5):1183–1210.

[You et al., 2023] You, X., Sheng, B., Huang, Y., Xu, W., Zhang, C., Wang, D., Zhu, P., and Ji, C. (2023). Closed-form approximation for performance bound of finite blocklength massive mimo transmission. *IEEE Transactions on Communications*, 71(12):6939–6951.

[Zhang et al., 2021a] Zhang, X. et al. (2021a). Aoi-driven statistical delay and error-rate bounded qos provisioning for murllc over uav-multimedia 6g mobile networks using fbc. *IEEE J. Sel. Areas Commun.*, 39(11):3425–3443.

[Zhang et al., 2021b] Zhang, X. et al. (2021b). Joint optimization and tradeoff modeling for peak aoi and delay-bound violation probabilities over urllc-enabled wireless networks using fbc. In *ICC 2021 - IEEE Int. Conf. Commun.*, pages 1–6.

[Zhang et al., 2020] Zhang, X.-M., Han, Q.-L., Ge, X., Ding, D., Ding, L., Yue, D., and Peng, C. (2020). Networked control systems: a survey of trends and techniques. *IEEE/CAA Journal of Automatica Sinica*, 7(1):1–17.

[Zhang et al., 2021c] Zhang, Z. et al. (2021c). Energy-efficient secure video streaming in uav-enabled wireless networks: A safe-dqn approach. *IEEE Trans. Green Commun. Netw.*, 5(4):1892–1905.

[Zhao et al., 2024] Zhao, Z. et al. (2024). Deep learning for wireless-networked systems: A joint estimation-control–scheduling approach. *IEEE Internet Things J.*, 11(3):4535–4550.

[Zheng et al., 2022] Zheng, Y. et al. (2022). Enabling robust drl-driven networking systems via teacher-student learning. *IEEE J. Sel. Areas Commun.*, 40(1):376–392.

[Östman et al., 2019] Östman, J. et al. (2019). Peak-age violation guarantees for the transmission of short packets over fading channels. In *IEEE INFOCOM 2019 - IEEE Conf. Comput. Commun. Workshops*, pages 109–114.