Doctoral Dissertation

Cooperative AI-driven Communication Framework for Wireless Multihop Networks

CUI Zhihan

Supervisor: LIM Yuto

Graduate School of Advanced Science and Technology

Japan Advanced Institute of Science and Technology

[Information Science]

March, 2026

Abstract

The continuous evolution from fifth-generation (5G) toward Beyond 5G (B5G) and sixth-generation (6G) systems introduces stringent performance requirements in terms of network capacity, end-to-end latency, and energy efficiency under dynamic and densely deployed wireless environments. Multi-server and multihop network architectures, together with the emergence of artificial intelligence (AI) and multi-access edge computing, provide opportunities for adaptive and scalable communication. However, they also create challenges involving interference-coupled throughput degradation, latency escalation caused by load fluctuations, and inefficient transmit power usage.

To address these challenges, this dissertation proposes a cooperative AI-driven communication (CORE) framework for wireless multihop networks. The framework enables cross-layer and cooperative optimization through three AI-based schemes that operate progressively according to network conditions. First, the efficient capacity management (eCAP) scheme enhances throughput by integrating factor-graph-assisted topology formation, two-phase Q-learning routing, and compute-and-forward transmission to construct interference-resilient multihop paths. Second, the extreme low latency transmission (eLOW) scheme reduces service latency in multi-server wireless networks by employing a Broad Learning System (BLS) for fast and adaptive server allocation under varying traffic loads and link quality. Third, the unified power management (uPOW) scheme improves energy efficiency and interference resilience in multi-server wireless multihop networks by combining BLS-based server allocation, SINR-driven Q-learning path selection, and Consensus Transmit Power Control.

The proposed framework is evaluated through numerical simulations conducted in Python and MATLAB under diverse density, interference, and traffic conditions. Experimental results demonstrate that:

- The eCAP scheme improves network capacity by up to 1.8 times compared with baseline multihop routing protocols through learning-based topology construction and cooperative forwarding.
- The eLOW scheme reduces service latency by up to 47% and alleviates congestion effects in multi-server deployments through prediction-driven BLS server allocation.

- The uPOW scheme reduces interference power by approximately 40% and achieves 35% energy savings while maintaining high network capacity via distributed cooperative power coordination.

By integrating distributed learning, cooperative decision-making, and adaptive optimization, the CORE framework establishes a foundation for scalable, energy-efficient, and low-latency communication in B5G and 6G wireless systems. The framework demonstrates the feasibility of AI-native network control and offers a conceptual and algorithmic basis for future autonomous communication architectures.

Keywords: Cooperative communication, AI-driven framework, Wireless multihop network, Multi-server edge computing, Capacity optimization, Latency reduction, Power control, 6G.

Acknowledgement

Foremost, I would like to express my deepest gratitude to my supervisor, Associate Professor **LIM Yuto** for his invaluable guidance. His sincerity and supervision have deeply inspired me a lot, his supervision gives me lots of motivation, and his high generosity was a big help to make my enjoyable time in Japan, especially at JAIST. Additionally, this endeavor would not have been possible without the generous support from Professor **TAN Yasuo**, the respectful second supervisor, and the defense examiner.

I would like to extend my sincere thanks to Professor **TALEB Tarik**, the advisor for the Internship, for his invaluable patience and guidance. Besides, I am also thankful to Dr. **CHEN Yan**, supervisor of the internship, for his kindness during the internship.

Words cannot express my gratitude to my defense examiners, Professor **KURKOSKI Brian Michael** and Professor **BEURAN Razvan Florin**. I could not have undertaken this PhD journey without their support.

I am also grateful to my classmates, lab mates, and researchers from TAN Laboratory and QuWi Laboratory (LIM Laboratory), for their kind support and sharing in collaboration meetings. With their friendliness, my life in JAIST was fun every day. Then, I would like to thank unknown significant others (USO) who help me directly and/or indirectly to complete my study.

Last but not least, I would be remiss in not mentioning my family for their belief and letting me find my own self in education.

Contents

List of Figures

List of Tables

List of Abbreviations

1G	First Generation
6G	Sixth Generation
5G	Fifth Generation
ACK	Acknowledgment
AI	Artificial Intelligence
ALM	Airtime Link Metric
AP	Access Point
AWGN	Additive White Gaussian Noise
B5G	Beyond 5G
BLS	Broad Learning System
BLS/MS	BLS with Maximum SINR
BLS/MT	BLS with Minimum Time Delay
BLS/NS	BLS with Nearest MEC Server
BLSO	Broad Learning System Optimization
BLSQ	BLS-based Q-learning
BS	Base Station
CPU	Central Processing Unit

CoF Compute-and-Forward

CT Concurrent Transmission

CTPC Consensus Transmit Power Control

DIFS Distributed Coordination Function Inter-Frame Space

D2D Device-to-Device

DL Deep Learning

DNN Deep Neural Network

DRL Deep Reinforcement Learning

E2E End-to-End

E8 E8 Lattice Code

ES Edge Server

FD Full-duplex

FG Factor Graph

FMDP Finite Markov Decision Process

GNN Graph Neural Network

GHz Gigahertz

IEEE Institute of Electrical and Electronics Engineers

INLPS SINR-based Learning Path Selection

IoT Internet of Things

LC Lattice Code

LPS Learning Path Selection

MAC Medium Access Control

MEC Multi-access Edge Computing

MIMO Multi-Input Multi-Output

ML Machine Learning

MDP Markov Decision Process

MSWN Multi-server Wireless Network

MWMN Multi-server Wireless Multihop Network

NLC Nested Lattice Code

NLPS SNR-based Learning Path Selection

OS Orchestration System

PHD Predefined-Path High-speed Device

PHY Physical Layer

QoS Quality of Service

RL Reinforcement Learning

RLD Random-Path Low-speed Device

RSSI Received Signal Strength Indicator

SPA Sum-Product Algorithm

SIFS Short Inter-Frame Space

SINR Signal-to-Interference-plus-Noise Ratio

SNR Signal-to-Noise Ratio

SPST Shortest Path Spanning Tree

TDMA Time Division Multiple Access

THz Terahertz

TPC Transmit Power Control

UAV Unmanned Aerial Vehicle

UD User Device

uPOW Unified Power Management Scheme

WMN Wireless Multihop Network

WLAN Wireless Local Area Network

eLOW Extreme Low Latency Transmission

List of Symbols

The following list describes the symbols that are used within the body of this dissertation:

α	Learning rate of Q-learning
β	Biases of broad learning
$\boldsymbol{y}$	Adjacency matrix
δ	Weighting coefficient
ϵ	threshold of Q-learning
η_j	Noise level of node j
γ	Discount factor of Q-learning
$\hat{t}^{down}$	Resultant downloading time
$\hat{t}^{up}$	Task uploading time
λ	Regularization parameter
$\mathbb{I}(\cdot)$	Indicator function
$\mathcal{C}$	Network capacity
$\mathcal{D}_i^j$	Decision made by j for i
$\mathcal{E}$	Set of edge server
$\mathcal{I}$	Set of transmitting node
$\mathcal{J}$	Set of receiving node

$\mathcal{K}$	Set of interfering node to ongoing transmission
$\mathcal{O}$	Channel access overhead
$\mathcal{U}$	Set of user device
μ	Consensus coefficient
$\mu*$	Optimal consensus coefficient
Ω	Gini impurity
ω_{ij}	Wall attenuation of the transmission from node i to node j
$\phi(\cdot)$	Activation function of broad learning
ψ	Shadowing attenuation
σ_e	Processing speed of e
τ	Task tolerable time
Θ	Network latency
θ	Task completion time
$\tilde{r}$	Enhanced reward
$\xi(\cdot)$	Nonlinear activation function
ζ	Pathloss attenuation constant
A	Concatenation of feature and enhancement nodes
a	Action of Q-learning
a'	The best action
B	Channel bandwidth
b	Row vector of binary indicators
C_m	The mth layer

$Cost$ Normalized cost

CW_{min} Minimum of contention window

D Destination node

d_0 Decorrelation distance

d_{ij} Euclidean distance between node i and node j

E No. of edge server

e Edge server

F_m No. of nodes in mth layer

f_m nodes in mth layer

FER Frame error rate

G_{ij} Power ratio of the transmission from node i to node j

$GF(\cdot)$ Global function

H Enhancement node matrix of broad learning

I No. of transmitting node

i Transmitting node of ongoing transmission

J No. of receiving node

j Receiving node of ongoing transmission

K No. of interfering node

k Interfering node to ongoing transmission

L Packet, or task size

l Air time cost

L_{res} Estimated size of the processed task

L_{test} Size of test frame

M No. of layers

m Layers

N Total number of nodes in the uploading path

n Intermediate nodes on the path

O Total energy consumption

$p(\cdot)$ Proportion of samples

P^{int} Total interference power

P^{new} Updated power after applying the consensus scaling

P_i Transmit power of node i

PG Power reduction gain

PL_0 Pathloss under Friis free space model

PL_{ij} Channel gain between node i and node j

Q Q-value of Q-learning

q Quality of serves

Q_{max} Maximum Q-value

r Reward of Q-learning

R^{ave} Average throughput

R^{down} Download transmission rate

R_0 Basic rate for test packet

R_{SD}^{e2e} E2E throughput from S to D

S Source node

s State of Q-learning

SB Sub-branch of broad learning

$SINR_{ij}$ Signal-to-interference-plus-noise-ratio of transmission from node i to node j

SNR_{ij} Signal-to-noise ratio of transmission from node i to node j

t^{down} Result download time

t^{est} Task estimation time

t^{proc} Task processing time

t^{ser} Task service time

t^{up} Task upload time

T_v Transmission time of v

TG Throughput gain

U No. of user device

u User device

V Total No. of time slots

v No. of time slots

W Random weight of broad learning

WP Weight factor of PG

WT Weight factor of TG

X Input matrix of broad learning

Y Labeled dataset of broad learning

y Binary indicator variable in adjacency matrix

Z Mapped feature node matrix of broad learning

z Raw feature value

z' Normalized feature

z_{max} The maximum value of each feature across the dataset

z_{min} The minimum value of each feature across the dataset

Chapter 1

Introduction

From the first generation (1G) of analog communication systems to the fifth generation (5G), mobile networks have continuously evolved to support higher data rates, broader coverage, and increasingly diverse services. With the worldwide deployment of 5G now well established, research efforts from both academia and industry have shifted toward Beyond 5G (B5G) and sixth-generation (6G) wireless networks. Unlike previous generations that primarily focused on throughput enhancement, B5G and 6G networks are envisioned to support massive connectivity, ultra-low latency, high energy efficiency, and intelligent autonomous operation [4, 5]. These ambitious requirements are driven by the rapid growth of connected devices and the increasing complexity of wireless environments.

The evolution toward B5G and 6G is closely tied to emerging application domains in which communication networks play a critical and often mission-sensitive role. Representative examples include disaster recovery and emergency communication systems, where rapidly deployable and infrastructure-independent networks are required to maintain connectivity under damaged or unavailable infrastructure [6]; industrial Internet of Things (IIoT) applications, where time-sensitive data exchange among sensors, controllers, and edge servers is essential for safe and efficient operation [7]; and large-scale smart city deployments, such as intelligent transportation systems and public event monitoring, where a massive number of devices must be supported simultaneously under stringent latency and energy constraints [8]. In these scenarios, the ability to provide scalable, low-latency, and energy-efficient wireless communication is a fundamental requirement.

However, conventional centralized network architectures face inherent limitations in

such dynamic and large-scale environments. The reliance on fixed infrastructure and centralized control may lead to performance problems, single points of failure, and poor adaptability to rapid topology changes. As a result, wireless multihop networks (WMNs), combined with multi-access edge computing (MEC), have emerged as a promising paradigm for future communication systems. By enabling devices to relay data cooperatively and offload computation to nearby edge servers, WMNs with MEC support flexible deployment, extended coverage, and low-latency service provisioning in dynamic environments [9].

Despite these advantages, WMNs introduce a set of fundamental wireless-specific challenges. First, multihop transmission over shared spectrum leads to strong interference coupling among neighboring links, which severely limits network capacity in dense deployments, as established in classical capacity analysis of wireless networks. Second, E2E latency accumulates rapidly over multiple hops and server interactions, particularly in multi-server environments with time-varying traffic loads and heterogeneous computational resources. Third, excessive energy consumption caused by simultaneous transmissions and interference poses a major obstacle to sustainable operation, especially for battery-powered devices. These challenges are intrinsic to wireless multihop communication and cannot be effectively addressed by traditional static or centralized optimization approaches.

To cope with the increasing complexity and dynamics of future wireless networks, artificial intelligence (AI) has been widely investigated as a powerful tool for wireless optimization. Learning-based techniques, particularly reinforcement learning, have demonstrated strong potential in tasks such as routing, resource allocation, and power control. However, most existing AI-driven solutions focus on isolated optimization objectives and operate independently. In complex WMNs, such isolated decision-making may result in conflicting actions, unstable convergence, and inefficient utilization of network resources. This observation motivates a fundamental question: how can multiple AI-driven components be coordinated to operate cooperatively under shared wireless constraints?

This dissertation addresses this question by proposing a framework for WMNs. Rather than designing independent algorithms for individual problems, the proposed framework systematically integrates capacity optimization, latency-aware decision-making, and power management into a unified architectural structure. Through coordinated learning

and information sharing, the framework enables different optimization modules to work toward common network objectives while adapting to dynamic network conditions.

The contributions of this dissertation are threefold. First, an extreme low-latency transmission scheme is developed to suppress latency accumulation in multihop wireless networks. Second, an efficient capacity management scheme based on the Broad Learning System (BLS) is proposed for multi-server environments, enabling adaptive server selection and load balancing. Third, a unified power management scheme integrating BLS, reinforcement learning, and consensus-based power control is designed to mitigate interference and improve energy efficiency. Together, these schemes form a coherent cooperative framework that supports scalable, adaptive, and intelligent operation in wireless multihop networks.

Beyond technical contributions, this research has broader social implications. By improving the reliability, efficiency, and adaptability of WMNs, the proposed framework can support mission-critical services such as disaster response, public safety communication, and industrial automation. Moreover, the emphasis on cooperative and energy-efficient operation contributes to the sustainable development of large-scale wireless infrastructures in the B5G and 6G era.

1.1 Niche of Communication Systems

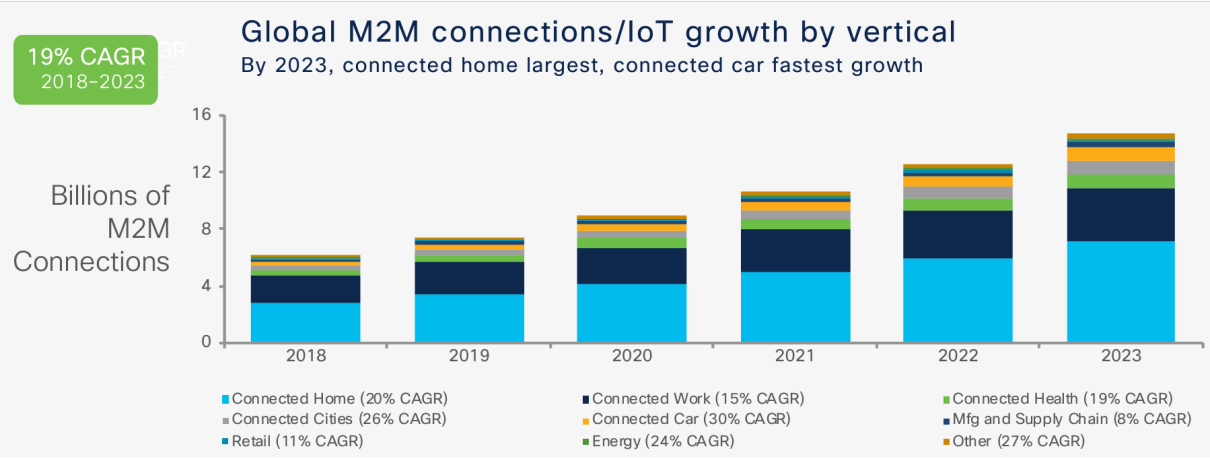


Figure 1.1: Global M2M connection growth by industries (2018-2023) [1].

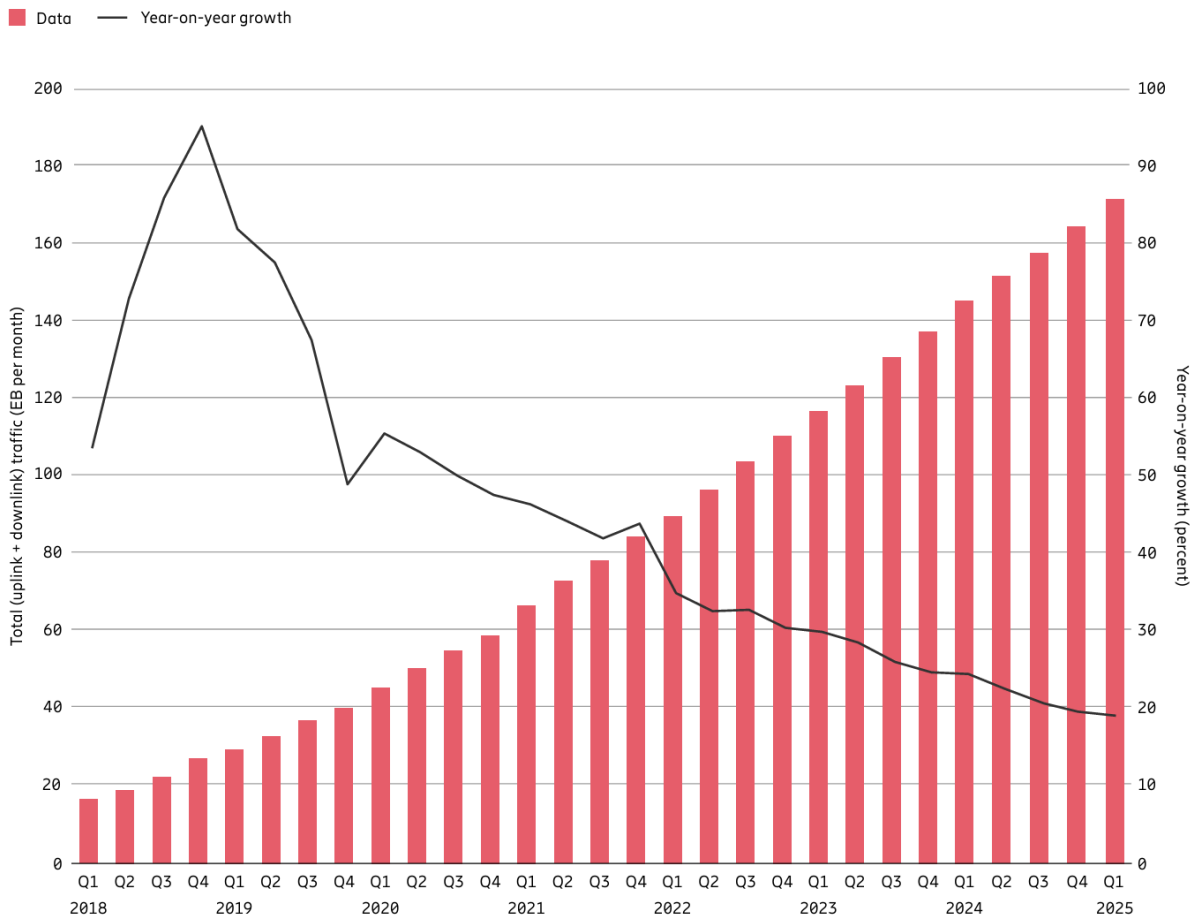


Figure 1.2: Forecast of global mobile data traffic growth through 2030 [2].

Recent years have witnessed a fundamental shift in the purpose and scale of wireless communication systems. According to Cisco's Annual Internet Report (2018-2023), global machine-to-machine connections have grown 2.4 times, from 6.1 billion in 2018 to 14.7 billion in 2023, representing a 19% compound annual growth rate. As illustrated in Figure 1.1, different industrial sectors contribute unevenly to this expansion. Connected home applications account for nearly half of the total M2M connections, including home automation, security, and surveillance. Connected car services, such as fleet management and in-vehicle infotainment, exhibit the fastest growth at 30% annually, while connected cities rank second with 26%. This trend demonstrates that wireless networks are no longer limited to human communication; they are rapidly becoming the backbone for pervasive Internet of Things (IoT) and machine-type communication services across all industries.

In parallel, the overall mobile data traffic is undergoing exponential expansion. According to the Ericsson Mobility Report [2], global mobile data traffic continues to grow at

a double-digit annual rate and is projected to exceed several thousand exabytes per month by 2030. Such a drastic increase is fueled not only by broadband consumption but also by emerging edge-intelligent applications such as real-time analytics, extended reality, and cooperative autonomous systems. As shown in Figure 1.2, traffic intensity grows faster than the number of connected devices, signaling a paradigm shift from connection-centric to computation-centric communication.

These transformations define the new niche of communication systems in the Beyond 5G and 6G eras. Future networks must support ultra-dense device deployments, high-throughput data flows, and stringent latency requirements while maintaining energy efficiency. To achieve this, communication systems are expected to integrate AI and edge computing for adaptive resource management. In this context, this dissertation investigates AI-driven frameworks for capacity optimization, transmit-power control, and multi-hop routing in multi-server wireless networks, aiming to realize intelligent, scalable, and energy-efficient connectivity for next-generation communication systems.

1.1.1 Definition of Wireless Multihop Network

A wireless multihop network (WMN) is a set of wirelessly connected nodes that communicate without relying on centralized infrastructure such as base stations (BSs) or access points (APs). Each node can act not only as a source or destination but also as a relay, forwarding packets for other nodes in a multihop manner. Through device-to-device (D2D) communication, packets originating from a source node can be delivered to a distant destination by traversing multiple relay nodes. In this way, WMNs can effectively extend network coverage and improve spectrum utilization by reusing intermediate nodes as relay points.

Compared with traditional infrastructure-based wireless networks, WMNs do not require every user to be directly connected to a BS or AP. Instead, each user terminal can behave as a small access point, helping to forward data to nearby nodes. This decentralized architecture reduces the deployment cost of BSs for network operators and provides high flexibility in environments where infrastructure is difficult or expensive to install. Figure 1.3 shows an example of a WMN, where packets are delivered over several hops from sources to their corresponding destinations.

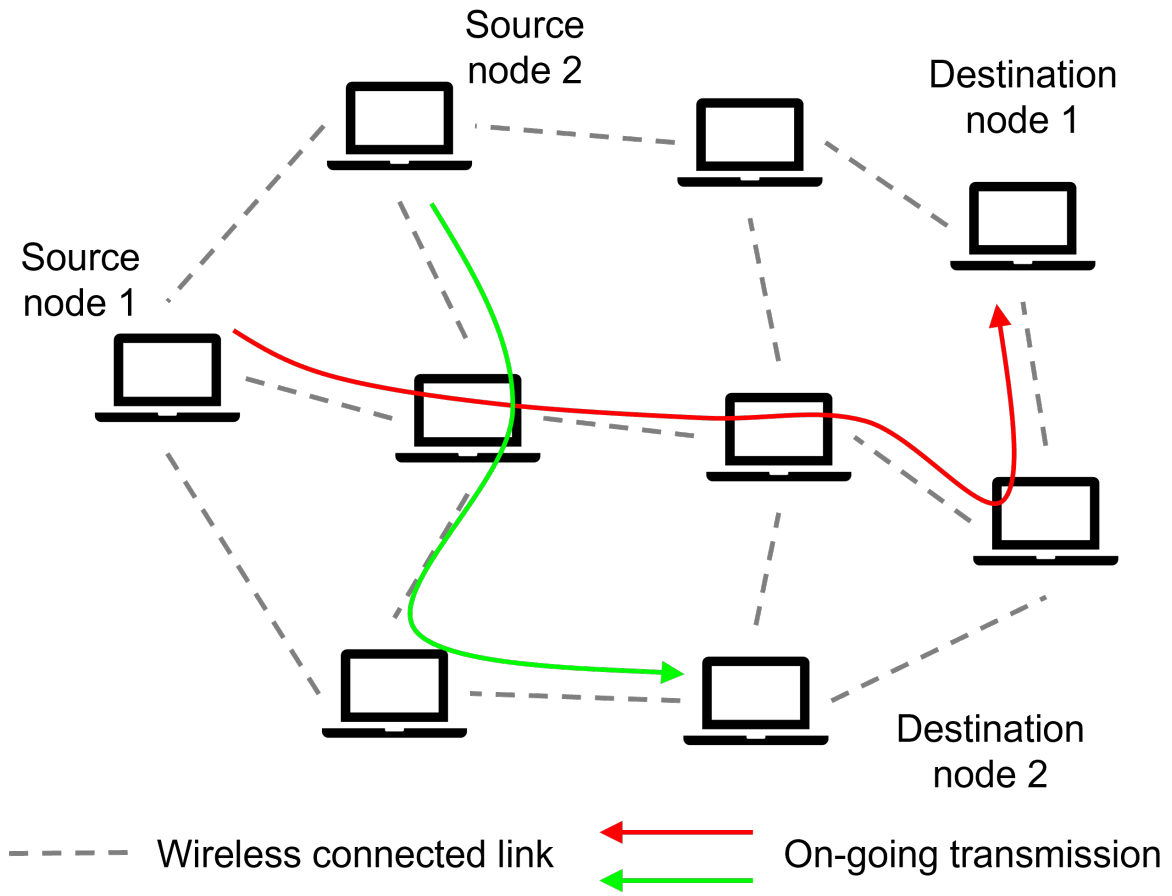


Figure 1.3: Example of a WMN.

Although the WMN has advantages such as decentralization, high network capacity, and long transmission distance, it also has many problems. The first problem is that, because of the increasing number of nodes in the network, there are many data transmission paths to be selected from the source nodes to the corresponding destination nodes by an efficient path selection algorithm. Since the criteria and conditions of each path differ, the results of path selection have a significant influence on the entire network capacity problem. Therefore, it is necessary to design a path selection algorithm for WMN that can effectively select path for each nodes and optimize network capacity, which is one of the objectives of this research. Second problem is occurred when too many nodes send messages to a single node, in which it can lead to high latency for the sent message to reach its destination node. Besides that, the queuing time and processing time will increase drastically when the number of nodes is large.

In practice, WMNs have been widely considered for various application scenarios where

flexible, resilient, and rapidly deployable communication is required. Typical examples include emergency and disaster-recovery networks, where temporary multihop links among rescue teams and command centers can be established without relying on damaged infrastructure; industrial monitoring and control systems, where sensors and controllers form multihop links inside factories or plants; and large-scale IoT deployments in smart cities, where streetlights, meters, and environmental sensors forward data cooperatively to edge servers. Furthermore, WMNs also play an important role in unmanned aerial vehicle (UAV) swarms, vehicular networks, and temporary event coverage, where infrastructure support is limited or constrained. These diverse application scenarios motivate the need for intelligent, AI-assisted mechanisms to manage routing, capacity, and power control in WMNs, as will be addressed in the following sections of this dissertation.

1.1.2 Future Challenges for Wireless Multihop Networks

As highlighted in the IMT-2030 framework [3], future 6G systems are expected to support ultra-high data rates, extremely low latency, massive connectivity, and superior energy efficiency across a wide range of usage scenarios, as illustrated in Figure 1.4. These ambitious capability targets place stringent requirements on the design of large-scale WMNs, which are envisioned as a key enabler for flexible, dense, and cost-effective connectivity in future communication infrastructures. While multihop communication offers extended coverage and improved scalability, it also introduces several critical technical bottlenecks that must be addressed before such networks can fully meet IMT-2030 expectations.

2) Capacity limitation and interference coupling. As device density grows toward the massive connectivity envisioned in Fig. 1.4, spectrum reuse and interference become dominant constraints on network capacity. In WMNs, many links share the same channel resources, and simultaneous transmissions can cause strong mutual interference. Moreover, the multihop structure introduces spatially correlated interference patterns, making throughput optimization and scheduling highly non-trivial. Without careful coordination of path selection, link activation, and channel usage, the aggregate network capacity can degrade significantly as the network scales.

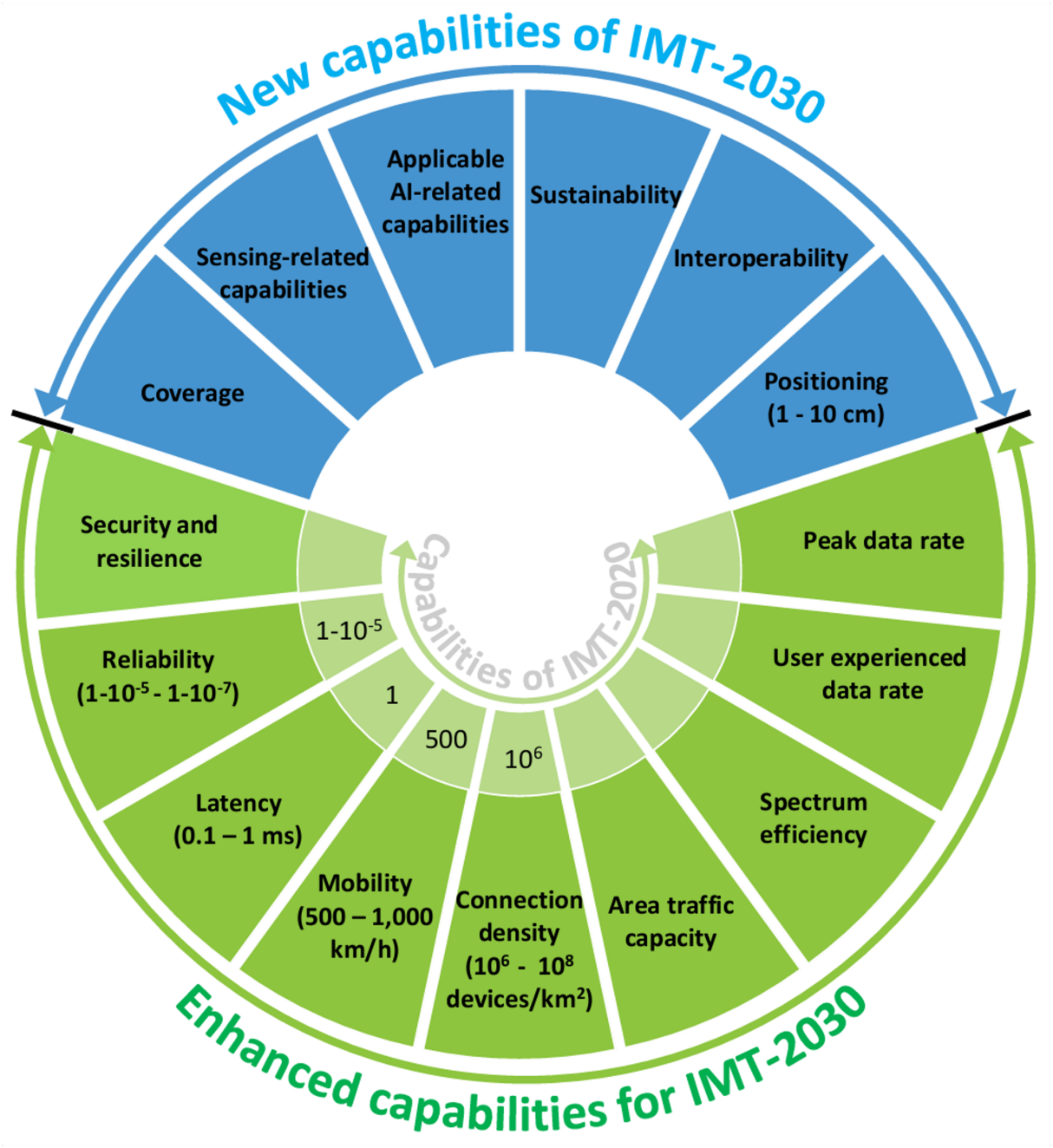


Figure 1.4: The capabilities of IMT-2030 [3].

1) Latency accumulation and dynamic topology. In multihop transmission, end-to-end (E2E) delay accumulates as data traverses multiple relay nodes. Each hop contributes transmission time, queuing delay, and potential route discovery or update overhead. Under dynamic conditions such as node mobility, variations in channel quality, or sudden traffic surges, these effects become more pronounced, resulting in significant fluctuations in service latency. Guaranteeing predictable low-latency performance in such

environments remains a fundamental challenge, especially for delay-sensitive applications like autonomous control, industrial automation, and immersive services.

3) Energy consumption and power efficiency. The maintenance of high data rates and low latency often requires increased transmit power, which in turn raises energy consumption and interference levels. In dense multihop deployments, excessive power usage shortens device lifetime and undermines the energy-efficiency goals emphasized in the IMT-2030 framework [3]. Balancing reliable communication with constrained energy budgets, therefore becomes a key design issue, particularly for battery-powered or resource-limited devices.

In summary, latency accumulation, capacity limitation, and power efficiency represent three tightly coupled challenges that constrain the scalability and sustainability of wireless multihop networks. Objectively understanding and quantifying these issues provides the foundation for the subsequent chapters of this dissertation, which focus on designing transmission, capacity management, and power control schemes that are more suitable for future large-scale wireless multihop environments.

1.2 Research Problems and Motivation

With the rapid expansion of wireless communication networks, the evolution toward Beyond 5G and 6G introduces unprecedented challenges in capacity management, latency control, energy efficiency, and intelligent coordination. Future networks are expected to support massive connectivity, ultra-low latency, and high energy efficiency while simultaneously enabling autonomous and cooperative decision-making among distributed nodes. However, achieving these ambitious goals is particularly difficult in dynamic and densely deployed environments, where multiple devices and servers interact under limited spectrum, energy, and computation resources. From a research perspective, the challenges faced by future wireless multihop networks can be broadly categorized into two classes. The first class consists of wireless-specific challenges, including network capacity degradation under interference, latency accumulation in multi-server and multihop transmission, and excessive energy consumption in dense deployments. The second class comprises AI-

related challenges arising from the application of learning-based techniques to wireless networks, such as stability, scalability, and coordination among multiple AI-driven components. This dissertation addresses both classes of challenges, with a particular focus on enabling effective cooperation among AI-driven entities to support efficient, scalable, and sustainable wireless multihop networks.

1.2.1 Wireless-Specific Challenges in Multihop Networks

WMNs exhibit a number of fundamental challenges that arise from the physical and architectural characteristics of wireless communication. Unlike challenges introduced by learning-based control mechanisms, these issues exist regardless of whether artificial intelligence is employed, and are primarily caused by shared spectrum access, interference coupling, multihop transmission dependency, and distributed network topology. In the context of Beyond 5G and future 6G systems, these wireless-specific challenges become more pronounced due to increasing node density, heterogeneous traffic demands, and the coexistence of multiple edge servers. The following subsections summarize three representative wireless-specific challenges that motivate the need for advanced optimization and coordination mechanisms in WMNs.

Network Capacity Problem

In wireless multihop networks, the large number of nodes and dense spatial distribution of links create strong mutual interference among simultaneous transmissions. Because multiple neighboring links contend for the same wireless channel, the achievable data rate of each hop is highly dependent on the interference conditions of nearby links. As a result, identifying and maintaining a routing path that consistently delivers high throughput becomes extremely difficult. Even paths with a small hop count may suffer from hidden terminals, bottleneck relays, or localized interference clusters, leading to unexpectedly low E2E throughput. This scarcity of high-throughput multihop routes causes the overall network capacity to remain low even when rich connectivity exists. Therefore, designing cooperative and learning-based transmission mechanisms that can form high-throughput paths under dense interference is a critical challenge for improving capacity in wireless multihop networks.

Network Latency Problem

In multi-server environments, the E2E service latency experienced by a device is determined by multiple factors, including wireless transmission delay, server-side queueing delay, computational load, and feedback latency. While the availability of multiple servers provides flexibility for task execution and data offloading, it also introduces dynamic latency behaviors: a server that offers low delay initially may quickly become congested as more devices associate with it, causing sudden and unpredictable escalation of service time.

Latency amplification is further exacerbated by time-varying traffic loads, heterogeneous server capacities, channel fluctuations, and asynchronous computation patterns across servers. As a result, selecting a low-latency server or transmission route becomes highly challenging, because a decision that is optimal at the moment of association may become suboptimal shortly afterward. Traditional selection or scheduling strategies rely on static metrics—such as minimum distance, maximum signal strength, or instantaneous queue length—which fail to capture temporal variations and therefore cannot prevent latency accumulation proactively.

Consequently, a fundamental research challenge is how to accurately anticipate latency variations and update server- or transmission-related decisions before delay builds up. Achieving low and stable latency under dynamic network conditions requires a prediction-driven mechanism capable of learning latency patterns from real-time network features and using these insights to guide adaptive decision-making.

Energy Efficiency and Power Control

As wireless networks become denser, simultaneous transmissions generate high interference levels, increasing energy consumption and reducing transmission reliability. Traditional centralized power control schemes require global network information and lead to excessive signaling overhead, making them impractical for large-scale networks. Moreover, simple power reduction alone may mitigate interference but at the expense of throughput degradation. Therefore, achieving energy-efficient operation in future wireless multihop networks requires distributed and cooperative power control mechanisms that can balance interference mitigation and capacity preservation. Such mechanisms are essential to

ensure sustainable communication in large-scale, interference-prone environments.

1.2.2 AI-related Challenges and Intelligent Cooperation

In addition to the wireless-specific challenges discussed above, the integration of AI into wireless communication systems introduces a new class of research problems. Applying AI techniques to wireless networks involves a wide range of challenges, including data availability, learning stability, model generalization, online adaptation, and coordination among multiple learning entities. Each of these aspects can significantly affect the performance and reliability of AI-driven wireless systems.

Among these AI-related challenges, this dissertation focuses specifically on the problem of intelligent cooperation. In WMNs, multiple AI-driven modules operate over shared spectrum and are tightly coupled through interference and network dynamics. When such modules are designed or optimized in isolation, their decisions may conflict, leading to unstable behavior, slow convergence, or degraded network performance.

Therefore, enabling effective cooperation and coordination among AI-driven components is a fundamental requirement for applying AI to wireless multihop networks. This work investigates how cooperative learning and coordinated decision-making can be systematically incorporated into a unified framework, allowing different AI modules to work toward common network objectives rather than competing for locally optimal solutions.

1.3 Research Vision, Purpose and Objectives

With the rapid evolution toward B5G and 6G systems, future wireless networks are expected to support extremely high data rates, massive connectivity, and stringent QoS requirements under highly dynamic and dense deployment scenarios. In particular, multi-server and multihop architectures are anticipated to play a central role in extending coverage, enhancing capacity, and enabling computation offloading at the network edge. Meanwhile, the increasing integration of AI into network control and resource management introduces both new opportunities and challenges for coordination and cooperation among distributed intelligent entities. However, as discussed in the previous sections, these advantages come at the cost of new challenges in network capacity management,

E2E latency, energy efficiency, and intelligent coordination.

The vision of this dissertation is to contribute to the development of an intelligent, cooperative, and scalable communication framework for WMNs. Such a framework should be capable of managing capacity, minimizing latency, optimizing energy efficiency, and enabling AI-driven cooperation among network entities in a distributed and adaptive manner. The long-term goal is to provide a conceptual and algorithmic foundation for future network architectures that integrate communication, computation, and intelligence in a unified way, thereby realizing autonomous and self-organizing network operations.

The purpose of this dissertation is to investigate how WMNs can be systematically optimized through cooperative and AI-driven schemes. Rather than treating capacity, latency, and energy efficiency as independent problems, this research seeks to understand their mutual interdependence and to develop cross-layer strategies that jointly improve network throughput, service delay, and power performance. Furthermore, the study aims to design an AI-driven cooperative communication framework that enables distributed decision-making and coordination among intelligent network entities. By leveraging adaptive and self-optimizing schemes, the framework aspires to achieve global optimization, thereby ensuring scalability and resilience in future 6G environments.

In line with the research problems identified earlier, the main objectives of this dissertation are:

- **Objective 1:** To design a multihop transmission scheme that maximize network capacity while maintaining reliable connectivity in dynamic wireless environments (the disseminated publications on this objective are [2], [6], [11], [12], and [13])

- **Objective 2:** To establish a network latency reduction scheme that mitigates E2E delay accumulation and maintains stable network performance under dynamic conditions (the disseminated publications on this objective are [4], [5], and [9])

- **Objective 3:** To develop a distributed power control and energy management scheme that mitigates interference and reduces overall power consumption without degrading capacity performance (the disseminated publications on this objective are [1], [3], [7], and [8])

By pursuing these objectives, this dissertation seeks to tackle four key challenges in

future AI-driven wireless multihop networks: improving capacity, reducing latency, enhancing energy efficiency, and enabling intelligent cooperation. In doing so, it contributes to the development of adaptive, autonomous, and high-performance communication systems for Beyond 5G and 6G networks.

1.4 Research Significance

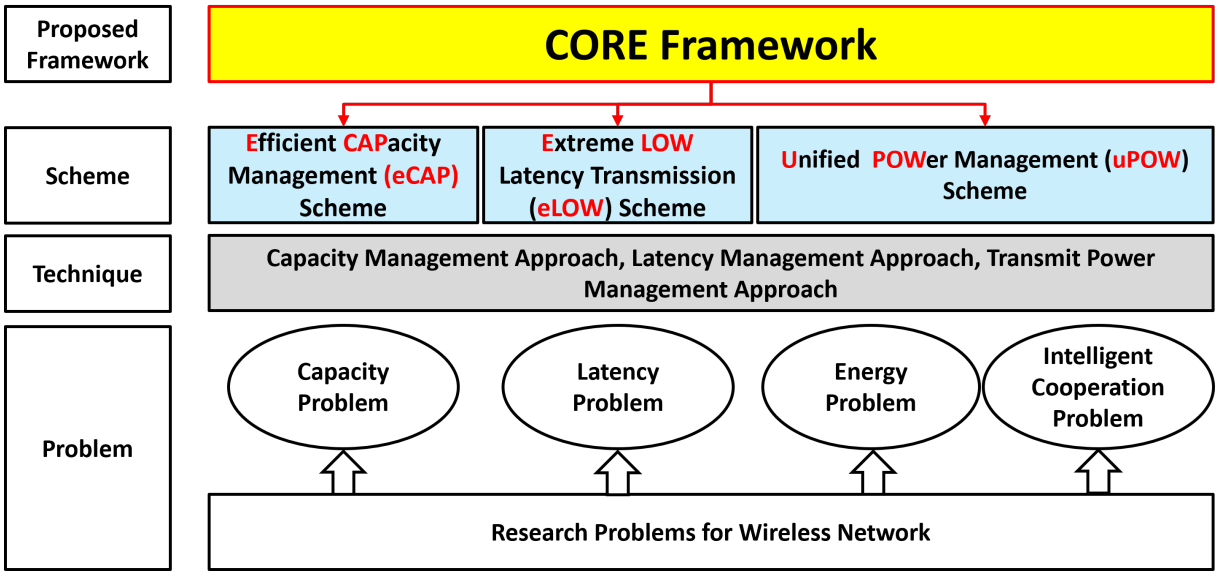


Figure 1.5: Proposed CORE framework for WMNs optimization.

Existing studies on wireless multihop networks have mainly addressed capacity, latency, and energy issues independently, often relying on static network assumptions or predefined optimization objectives. Such isolated approaches limit the overall performance when the network scale, mobility, and interference level dynamically change. In addition, current frameworks seldom integrate AI techniques with cooperative communication principles to adaptively manage the growing complexity of future B5G and 6G environments. Consequently, there remains a pressing need for a unified and intelligent framework that can jointly optimize network capacity, latency, and power efficiency.

To address these challenges, this dissertation proposes a Cooperative AI-driven Communication (CORE) Framework for wireless multihop networks, as illustrated in Figure 1.5. The framework is structured around three primary schemes: efficient capacity management (eCAP), extreme low latency transmission (eLOW), and unified power

management (uPOW). Each is designed to tackle a specific research problem in capacity, latency, and energy optimization, respectively. These schemes are supported by three corresponding management techniques: capacity management, latency management, and transmit power management, which together form the foundation of the proposed cooperative and adaptive communication system.

The significance of this research lies in its holistic approach to optimizing wireless multihop networks. By integrating capacity, latency, and energy management into a unified framework, the proposed approach achieves a balance among throughput enhancement, delay reduction, and power efficiency under dynamic network conditions.

This framework not only advances theoretical understanding of cross-layer optimization but also provides a scalable solution for practical 6G applications such as autonomous driving, UAVs, industrial automation, and large-scale IoT deployments. Hence, the proposed framework contributes to the realization of intelligent, sustainable, and high-performance wireless communication systems for the next generation of networks.

1.5 Dissertation Organization

The remainder of this dissertation is organized as follows:

- **Chapter 2: Cooperative AI-driven Communication Framework** This chapter introduces the proposed CORE framework, which serves as the architectural backbone of this dissertation. It first reviews the key enabling technologies and research foundations, then describes the overall framework architecture and the interaction between protocol layers. The three cooperative AI schemes are also outlined to establish the conceptual and functional links to the subsequent chapters.

- Chapter 3: Efficient Capacity Management Scheme
  This chapter addresses the capacity problem in wireless multihop networks. It presents the design and formulation of the eCAP scheme, which focuses on optimizing network capacity and reducing computation time of algorithms under dynamic network conditions.

- Chapter 4: Extreme Low Latency Transmission Scheme
  This chapter deals with the latency problem in multi-server wireless networks. It

describes the eLOW scheme that manages server selection and resource distribution to enhance overall network latency and network capacity caused by high-density connections.

- Chapter 5: Unified Power Management Scheme

  This chapter focuses on the energy problem in dense wireless environments. It introduces the uPOW scheme, which performs distributed transmit power coordination to achieve a balance between interference mitigation and energy efficiency across the network.

- Chapter 6: Conclusion and Future Work

  The final chapter summarizes the findings of this dissertation, highlighting the contributions of each proposed scheme within the CORE framework. It also discusses future research directions toward more intelligent and sustainable multihop wireless networks in B5G and 6G systems.

Chapter 2

Cooperative AI-driven Communication Framework

This chapter provides an overview of the proposed framework, which plays a central role in shaping the future communication architecture for WMNs. First, the key technologies related to the framework are discussed. Second, the overall design of the proposed framework and its three cooperative schemes is presented. Finally, the assumptions and constraints underpinning the framework are explained.

2.1 Related Key Technologies to the Framework

The proposed framework is built upon several interrelated technologies that collectively define the foundation for intelligent, cooperative, and distributed wireless communication in the B5G and 6G eras. In particular, WMNs provide flexible connectivity and network scalability; the IoT enables massive sensing and automation; MEC offers localized computation and reduced latency; and AI empowers adaptive and cooperative decision-making. Together, these technologies converge to realize a network ecosystem capable of autonomous learning and real-time optimization. This section discusses these key technologies and their relationships to the core research challenges identified in Chapter 1, which are capacity optimization, latency reduction, and energy efficiency.

2.1.1 Internet of Things

The IoT represents one of the most transformative paradigms driving the evolution of B5G and 6G communication systems. By connecting billions of devices ranging from sensors and wearables to autonomous vehicles and industrial robots, IoT enables seamless data collection, intelligent control, and automation across a wide range of domains, including healthcare, manufacturing, transportation, and smart cities. In particular, the IIoT has become a critical enabler for Industry 4.0, supporting machine-type communication, predictive maintenance, and process automation in factory environments.

However, in large-scale IoT or IIoT environments, the vast number of connected devices and heterogeneous data streams can easily lead to network congestion, latency, and energy inefficiency. Traditional centralized cloud architectures struggle to handle such massive connectivity due to long transmission distances and heavy backhaul loads. To overcome these limitations, distributed architectures, such as WMNs integrated with MEC, have been proposed. By enabling localized data processing and device-to-device (D2D) relaying, these systems achieve lower latency and higher reliability, which are particularly essential in emergency rescue communication scenarios where network infrastructure may be unavailable.

Nevertheless, IoT networks remain highly dynamic, with fluctuating link quality and limited device power. Incorporating AI-driven coordination becomes indispensable for maintaining adaptive routing, balanced task allocation, and interference control. By combining IoT with WMN and MEC infrastructures, future communication systems can realize large-scale, self-organizing, and resilient network environments that provide both massive connectivity and reliable low-latency communication.

2.1.2 Multi-access Edge Computing

MEC has emerged as a key enabling technology for B5G and 6G systems, aiming to bring computation and storage resources closer to end users. By deploying edge servers (ESs) at the network periphery, MEC minimizes the distance between data generation and processing, thereby reducing latency and alleviating backhaul congestion. This paradigm supports delay-sensitive and computation-intensive applications such as autonomous driving, industrial automation, and immersive extended-reality services.

In the context of WMNs, integrating MEC enables distributed task offloading and cooperative computation. Devices can dynamically select nearby ESs for computation offloading, while intermediate relay nodes forward data through multiple hops. Such a multi-server environment provides high flexibility and scalability, as multiple ESs can cooperatively serve dense clusters of devices in real time. However, when multiple ESs coexist, devices face complex decisions: which server to connect to and how to distribute computation efficiently. Unbalanced server selection or inefficient routing can lead to congestion, high interference, and reduced network utilization.

To address these issues, AI-driven methods have been increasingly adopted for resource management and server coordination in MEC-enabled WMNs. These approaches allow devices and ESs to autonomously learn optimal task-offloading and routing strategies from environmental feedback, thereby improving throughput, minimizing delay, and achieving balanced network performance. This dissertation extends such concepts further by introducing a Broad Learning System (BLS)-based capacity management scheme to enable efficient, adaptive, and scalable optimization across multiple servers.

2.1.3 Extensions of Wireless Multihop Network

WMNs represent a fundamental architectural paradigm for next-generation mobile communication systems. Unlike traditional infrastructure-based networks, where base stations or access points act as centralized coordinators, WMNs operate in a decentralized manner, allowing each node to function as both a transmitter and a relay. Through device-to-device communication, nodes forward packets cooperatively via multiple hops until the destination is reached. This multihop relaying mechanism effectively extends transmission range, enhances spectrum utilization, and improves overall network resilience.

The WMN topology consists of multiple nodes interconnected through dynamic wireless links. Each node can adaptively forward data based on local observations of channel quality, link interference, or queue state. This decentralized and self-organizing nature enables WMNs to operate flexibly in scenarios such as disaster recovery, intelligent transportation, or temporary event networks, where infrastructure deployment is limited or unavailable.

Despite these advantages, WMNs face several intrinsic challenges. As the number of

participating nodes increases, the network experiences frequent topology changes, complex path selection, and severe interference coupling. Moreover, achieving efficient transmission and power allocation under uncertain environments becomes increasingly difficult with traditional static control schemes. To address these challenges, intelligent optimization mechanisms that leverage AI and MEC are essential for enhancing network capacity utilization, reducing network latency, and adaptively managing energy consumption. Therefore, this dissertation focuses on addressing three interrelated issues in WMNs, including efficient capacity management under dynamic connectivity, latency reduction through intelligent multihop path selection, and energy-aware coordination via adaptive power control.

Over time, the WMN concept has evolved into several specialized forms to support the increasing diversity of wireless services. One important extension is the multi-server wireless network (MSWN). As illustrated in Figure 2.1, which multiple servers coexist in the network and user devices may choose and offload computation tasks to different servers. MSWNs enhance service coverage, processing capability, and load balancing com-

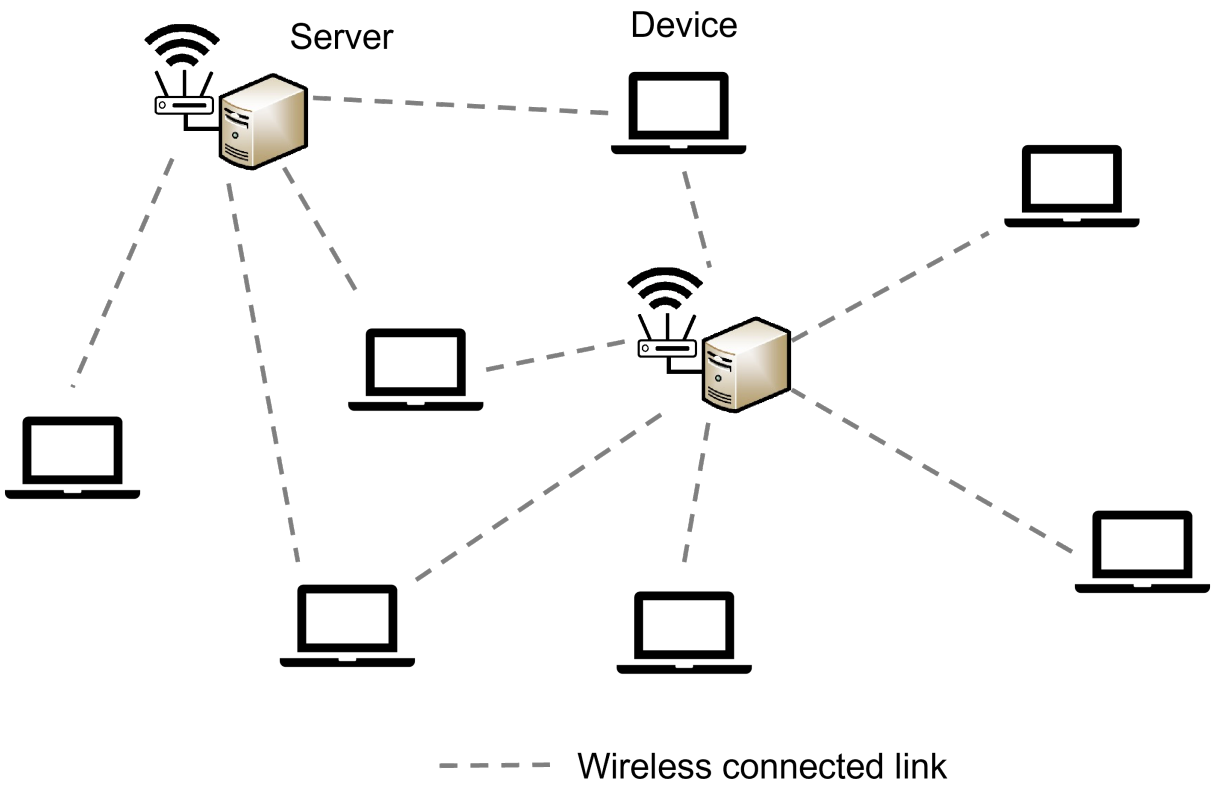


Figure 2.1: Example of a MSWN.

pared to conventional single-server designs. However, they introduce a new optimization dimension because choosing a suboptimal server under dynamic interference and traffic loads can drastically degrade system capacity.

Building on MSWNs, a more advanced paradigm has recently emerged: the multi-server wireless multihop network (MWMN). As illustrated in Figure 2.2, devices communicate with servers not only directly but also via multihop relaying through intermediate nodes. This architecture is particularly relevant in MEC-enabled systems, where both

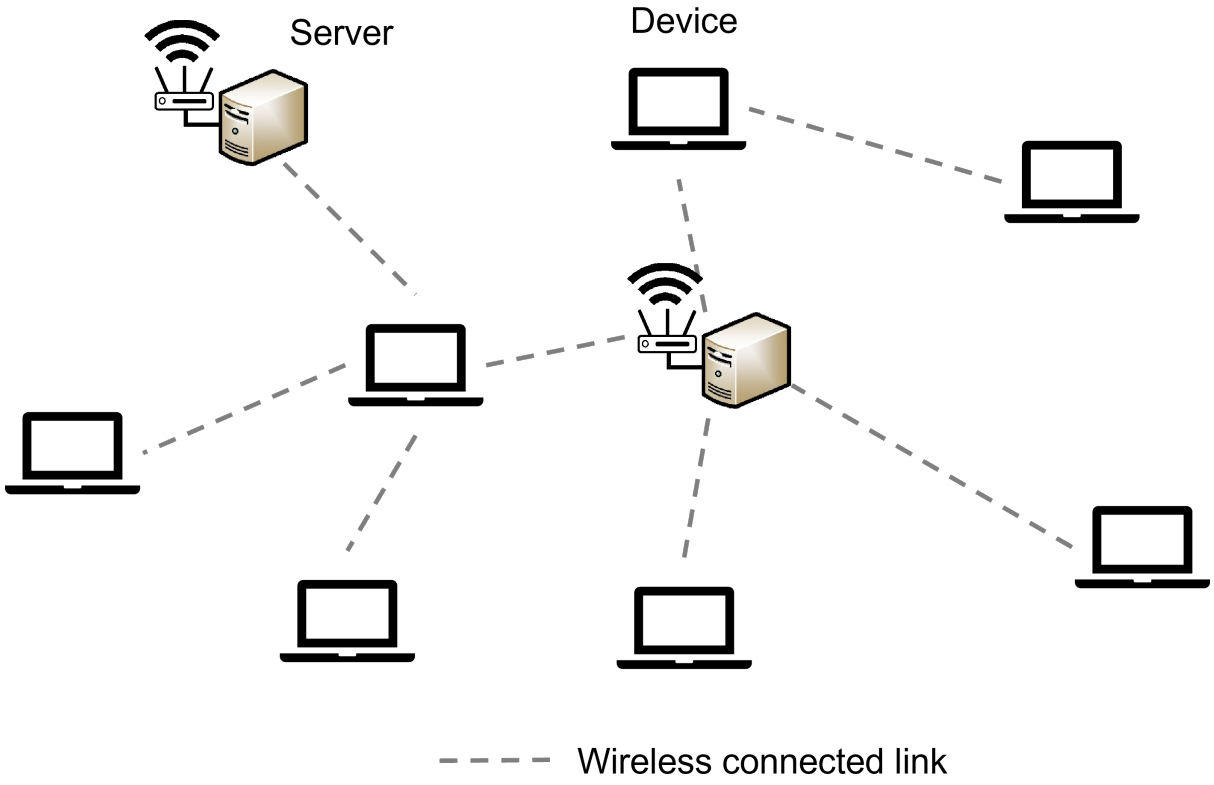


Figure 2.2: Example of a MWMN.

communication and computation paths may dynamically depend on wireless connectivity, interference conditions, and device mobility. In such environments, server allocation, routing, and transmit power control become tightly interdependent and must be jointly optimized.

These network variants form the technical background of this dissertation. Chapter 4 investigates intelligent server selection in MSWNs to improve network capacity in multi-server environments. Chapter 5 further extends this research to MWMNs, addressing the combined challenges of server allocation, path selection, and power control in large-scale

dense deployments.

It is worth clarifying that different network models and evaluation conditions are adopted across Chapters 3 to 5. This design choice does not indicate inconsistent system assumptions but rather reflects a methodological decision to isolate and optimize different performance objectives within the proposed framework.

Specifically, each component of the CORE framework is designed to address a distinct but interrelated optimization target: network capacity, network latency, and interference-aware power efficiency. Using a unified but overly complex network model for all components would obscure the individual contribution of each scheme, since multiple tightly coupled factors would vary simultaneously. Therefore, representative network models are selected in each chapter to emphasize the dominant bottleneck relevant to the corresponding optimization objective, while maintaining common architectural assumptions, including wireless multihop communication, interference-limited operation, and distributed AI-driven decision-making.

Through this approach, the proposed framework enables systematic and focused optimization of different performance dimensions, while preserving conceptual consistency at the framework level.

2.1.4 Artificial Intelligence in Wireless Networks

AI has become a pivotal enabler in the design and optimization of next-generation wireless communication systems. Traditional networks rely on deterministic algorithms and static configurations, which are insufficient for managing the complexity, heterogeneity, and dynamic nature of B5G and 6G environments. In contrast, AI introduces learning, prediction, and autonomous decision-making capabilities that enable adaptive and self-optimizing network operation.

AI techniques, including machine learning (ML), deep learning (DL), and reinforcement learning (RL), are being applied across all protocol layers. At the physical layer, AI supports channel estimation, interference prediction, and signal classification. At the MAC and network layers, it facilitates adaptive routing, intelligent resource allocation, and spectrum management. At the application layer, AI enables efficient task offloading, service migration, and energy-efficient computation through cross-layer optimization.

When integrated with WMNs, MEC, and IoT, AI enables distributed intelligence and cooperation among network entities. In such systems, multiple intelligent agents deployed across devices, edge servers, and orchestrators can collaboratively learn and optimize policies for transmission, routing, and power control. Mechanisms such as federated learning and multi-agent reinforcement learning further enable decentralized coordination, where nodes share knowledge without centralized control. These cooperative AI paradigms align directly with the core vision of this dissertation, providing the foundation for a scalable, autonomous, and energy-efficient communication framework for future 6G wireless networks.

2.2 Literature Review

Recent research has focused on incorporating AI into wireless networks to enhance spectrum efficiency, reduce latency, and enable dynamic resource management. This section reviews recent developments in AI-driven communication frameworks and highlights the growing trend toward cooperative, multi-layer, and multi-server optimization paradigms.

For instance, Chen et al. [10] introduced the concept of Wireless Big AI Models (wBAIM), a unified AI-driven architecture designed to enable multi-task, multi-scenario, and cross-layer cooperation in 6G wireless networks. Their work demonstrates how pre-trained large-scale models can facilitate collaborative resource management, cloud–edge coordination, and low-latency intelligent communication, providing valuable insights for future cooperative AI frameworks.

Deep reinforcement learning (DRL) has recently emerged as a powerful solution for radio resource allocation and management (RRAM) in large-scale heterogeneous wireless networks. Alwarafy et al. [11] conducted a comprehensive survey on DRL-based RRAM schemes, highlighting the limitations of conventional optimization, heuristic, and game-theoretic approaches in dynamic and ultra-dense environments. Their work emphasizes that DRL enables network entities to autonomously learn optimal power control, spectrum allocation, and user association strategies under incomplete network information. The study also categorizes major DRL algorithms, including Deep Q-Network (DQN), Deep Deterministic Policy Gradient (DDPG), and asynchronous advantage actor–critic, and discusses their applicability in both discrete and continuous action spaces, providing

important theoretical and methodological references for AI-driven resource management frameworks.

Ismail et al. [12] presented a comprehensive survey of deep reinforcement learning (DRL)-based resource scheduling mechanisms in MEC environments. The study classified DRL applications into three main areas: content caching, computation offloading, and resource management, and evaluated existing models according to architecture, objectives, algorithms, and learning structure. The paper highlights the significance of cooperative device–edge–cloud architectures and discusses how DRL techniques can achieve near-optimal scheduling and efficient utilization of communication, computation, and caching resources in dynamic MEC systems. This work provides essential insights for the development of AI-driven cooperative frameworks in multi-server edge environments.

Overall, the reviewed studies collectively emphasize the critical role of AI, particularly DRL and foundation model paradigms, in enabling cooperative and distributed intelligence in future wireless systems. However, most existing works remain limited to either single-hop or single-server environments and rarely consider the integration of multi-server, multihop, and cross-layer optimization. Addressing these limitations, this dissertation aims to design a unified cooperative AI-driven framework that jointly optimizes latency, capacity, and energy efficiency for wireless multihop networks.

2.3 Overview of Cooperative AI-driven Communication Framework

In the context of AI-enabled wireless communication systems, a framework is commonly understood as a unified architectural structure that integrates multiple functional components under shared system assumptions, information interfaces, and coordination mechanisms, rather than a single algorithm targeting an isolated objective [13]. Such a framework provides an architectural basis for organizing and orchestrating heterogeneous learning-driven modules in complex wireless environments.

Following this general notion, framework-level designs have been widely adopted in wireless networking research to address the intrinsic coupling among multiple network functions. For example, cross-layer optimization frameworks have been proposed to jointly

coordinate routing, power control, and resource management as distinct but interrelated modules within a unified system architecture [14, 15]. These studies demonstrate that a wireless networking framework commonly consists of multiple specialized components, each targeting a specific performance aspect, while operating coherently within a shared optimization structure.

Following this established line of research, this dissertation adopts the term "framework" to denote a unified cooperative control architecture and "scheme" to denote multiple functional components for WMNs. Within the proposed Cooperative AI-driven Communication (CORE) framework, capacity optimization, latency reduction, and power management are not treated as independent problems, but as interrelated components operating on a common network model with shared information representations and coordinated decision-making logic. Accordingly, the proposed AI-driven schemes are designed as modular components that interact coherently within the same framework, rather than as separate or isolated methodologies.

2.3.1 Architecture of the CORE Framework

Figure 2.3 illustrates the overall architecture of the proposed CORE framework. In the 5G Mobile Model [16], at the application layer, specified applications (e.g., large-area Wi-Fi coverage, factory device task offloading, emergency rescue communication, and so on) determine the communication request based on service requirements such as data volume, destination, and latency constraints.

This request is then delivered to the Open Transport Protocol (OTP) layer, where the transport protocol is selected. Transmission Control Protocol (TCP) and Multipath TCP are supported at this layer and are activated under different conditions: TCP for single-path reliable transmission and Multipath TCP for multiple paths when higher reliability or throughput is required.

The processed request is then passed to the Network Layer, where the routing protocol selects candidate paths between the communicating nodes. At this stage, only logical connectivity and hop-to-hop reachability are determined; the quality of each hop, interference state, and server association is not yet optimized. The network layer therefore, exports the routing table and the multihop transmission request to the Open Wireless

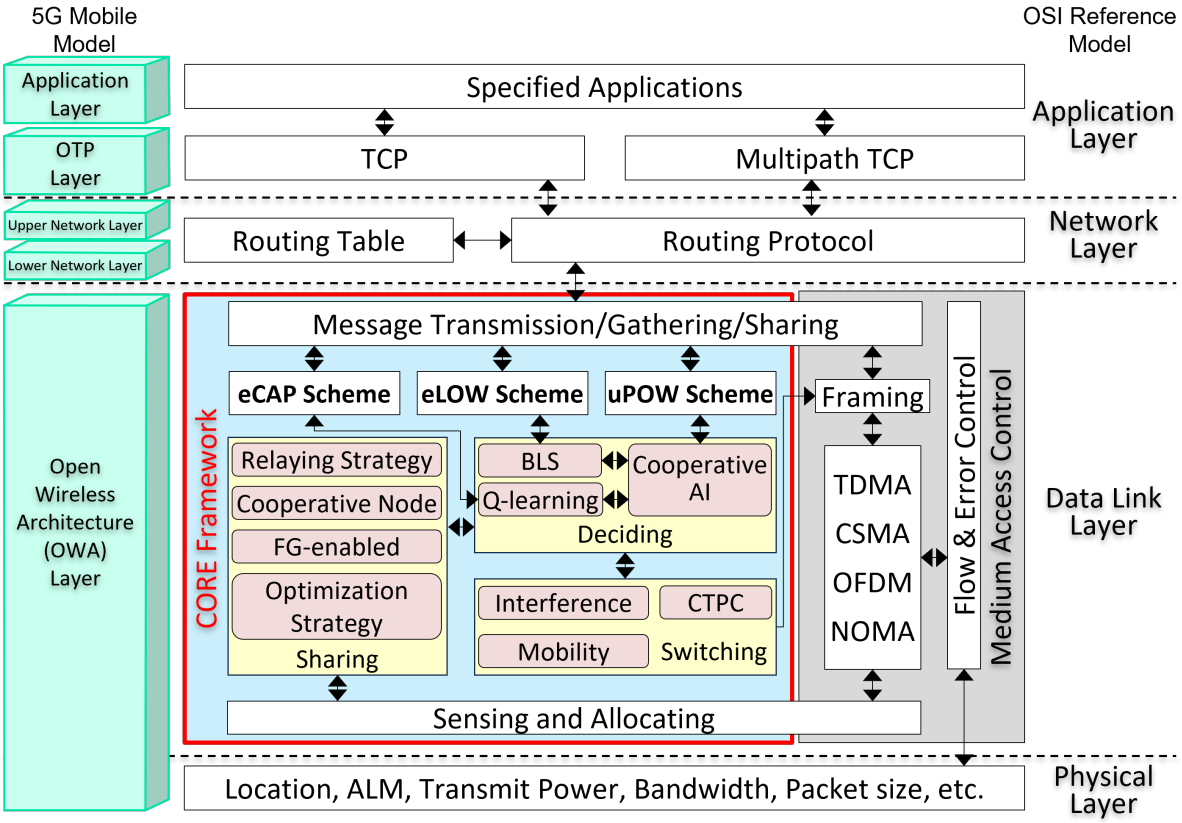


Figure 2.3: Proposed CORE framework.

Architecture (OWA) layer for further refinement.

The OWA layer is where the proposed CORE framework operates. CORE acts as an intelligent AI-driven control layer that performs cooperative optimization before actual transmission takes place. It integrates three functional modules to provide optimized communication decisions: Deciding, Sharing, and Switching. In the deciding module, CORE interprets the incoming request and selects the appropriate AI strategy based on network requirements and application goals. For capacity-critical traffic, the framework activates the eCAP scheme, where Q-learning performs path selection to maximize E2E throughput. For multi-server environments where congestion and load imbalance dominate, the framework activates the eLOW scheme, in which a BLS determines the optimal server allocation. For dense scenarios where interference and energy usage are key challenges, the framework activates the uPOW scheme, where BLS and Q-learning cooperate to jointly optimize server allocation and path selection.

In the sharing module, the selected decisions are broadcast to all participating nodes

to guarantee coherent execution. Under eCAP, the FG determines the appropriate server association, and CoF strategy establishes cooperative node pairing for simultaneous uplink forwarding. Under eLOW, server selection results are synchronized across devices and edge servers to prevent conflicting offloading decisions. Under uPOW, the server allocation and path selection results are shared through the orchestration system. Cooperative AI ensures that path selection and server allocation are aligned.

In the switching module, CORE preserves communication quality during data delivery by dynamically adjusting decisions in response to evolving network states. eCAP performs topology-dependent switching when bottleneck links change; eLOW triggers switching when server loads fluctuate; and uPOW conducts mobility- and interference-aware switching using Consensus Transmit Power Control (CTPC) to maintain stable throughput and energy efficiency in dense deployments. All schemes depend on the sensing and allocating module, which continuously collects real-time network parameters from the lower layers, including node location, airtime link metric (ALM), wireless link quality, transmit power, packet size, bandwidth, and interference level. These parameters are transformed into usable AI inputs and pre-assigned to the relevant scheme to minimize decision latency.

Once CORE finalizes the optimized communication decision, it is exported to the Medium Access Control (MAC) layer. The MAC layer performs framing and applies the appropriate access method, including Time Division Multiple Access (TDMA), Carrier Sense Multiple Access (CSMA), Orthogonal Frequency Division Multiplexing (OFDM), or Non-Orthogonal Multiple Access (NOMA). It depends on spectrum availability, node density, and coexistence requirements. Flow and error control mechanisms are then applied to ensure reliable data delivery before transmission enters the Physical Layer, where the optimized communication strategy is executed in the wireless medium. Meanwhile, the physical layer continuously reports link-level measurements to the sensing and allocating module, such as ALM, SINR, and transmit power, creating a closed-loop interaction. Due to this modular design, the CORE framework is compatible with a diverse range of wireless systems, including WLAN, B5G, IoT, D2D, MEC, WMN, and FD communication, making it a scalable foundation for future intelligent multihop networks.

Importantly, the software architecture design of CORE allows new learning paradigms

or AI-based optimization modules to be incorporated without modifying the protocol stack. As future wireless systems evolve, additional cooperative schemes can be integrated into the CORE architecture as additional modules, ensuring long-term scalability and adaptability. By integrating distributed sensing with AI-driven optimization and unified architectural coordination, the CORE framework provides a scalable and adaptive foundation for efficient communication in future wireless networks.

2.3.2 AI-driven Schemes of CORE Framework

The three AI-driven schemes of the proposed framework are designed to operate as a progressive optimization chain. The process begins with the eCAP scheme, which tackles the fundamental challenge of network capacity through FG and Q-learning. Building on the improved path selecting efficiency, the eLOW scheme applies BLS to allocate a server for each device based on strategies to reduce network latency. Finally, as network density increases, interference and energy constraints become dominant, and the uPOW scheme integrates BLS, Q-learning, and CTPC to achieve interference mitigation and energy-efficient operation. Together, these three schemes unified a unified and adaptive communication framework for WMNs.

Efficient Capacity Management Scheme

The eCAP scheme is introduced as the capacity-oriented module of the proposed framework, aiming to jointly improve network capacity and reduce transmission time in WMNs. eCAP exploits cooperative relaying and AI-driven path selection to construct efficient multihop paths toward a designated root node.

At the topology level, eCAP first employs a FG-based root node selection of a shortest-path spanning tree using the ALM, so that a root node and tree structure with link quality are chosen. On top of this structure, a two-stage Q-learning-based Learning Path Selection (LPS) mechanism is applied: an SNR-based phase to quickly discover high-quality paths, followed by a signal-to-interference-plus-noise ratio (SINR)-based refinement phase that explicitly accounts for interference. Finally, nested lattice coding combined with a CoF strategy enables paired transmissions toward common parents, effectively serving multiple flows within fewer time slots.

As a result, eCAP provides an AI-assisted cooperative transmission scheme that optimizes network capacity, shortens computation time, and reduces the number of required time slots. The detailed system model, algorithms, and performance evaluation of eCAP are presented in Chapter 3.

Extreme Low Latency Transmission Scheme

The eLOW scheme is introduced as the latency-oriented module of the proposed cooperative AI-driven framework, targeting MSWNs where many devices can associate with multiple edge servers. Its main goal is to reduce network latency, alleviate congestion, balance traffic loads, and enhance overall network throughput under dense and interference-prone conditions.

In eLOW, a BLS is adopted to learn the mapping from network states, such as device–server distances, channel conditions, task volumes, and server conditions, to server-association decisions. Owing to its shallow but "broad" architecture and incremental learning capability, BLS can be trained efficiently and updated online as devices or traffic patterns change.

On top of this learning engine, BLS-based strategies are considered. A minimum-delay-oriented variant focuses on reducing the E2E service time experienced by each device. A distance-oriented variant encourages association with nearby servers to lower propagation and access delay. A SINR-oriented variant emphasizes link quality and interference conditions to improve achievable data rates and stabilize performance at high densities. By combining these strategies, eLOW provides an AI-assisted latency reduction scheme that reduces network latency while maintaining scalability in multi-server deployments. The detailed system model and performance evaluation of eLOW are presented in Chapter 4.

Unified Power Management Scheme

The uPOW scheme functions as the module responsible for power control and interference mitigation within the proposed CORE framework. It focuses on achieving balanced performance between throughput enhancement, energy efficiency, and interference mitigation in wireless multihop networks. To this end, uPOW integrates three coordinated layers of optimization: BLS-based server allocation, SINR-driven Q-learning path selection, and

CTPC.

The BLS module first determines optimal server assignments by considering device–server distance, task load, and communication quality, thereby minimizing task delay under resource constraints. The Q-learning-based path selection then constructs adaptive multihop paths that maximize SINR and throughput while satisfying latency requirements. Finally, the CTPC layer coordinates transmit power adjustments across nodes through consensus optimization, reducing interference and improving network-wide energy efficiency. This unified cross-layer design enables cooperative decision-making among communication and computation entities, providing scalable and adaptive control for dense and dynamic MEC-enabled multihop environments.

Through these integrated components, uPOW effectively enhances network capacity, reduces task completion time, and maintains high QoS with lower energy consumption. A detailed formulation, algorithmic design, and simulation analysis of uPOW are provided in Chapter 5.

The three proposed schemes are not independent methodologies but modular components embedded within the same CORE framework. Each scheme targets a dominant performance problem under specific network conditions. These schemes share a common system model, network entities, and information exchange mechanisms, and can be selectively activated or combined according to the operational requirements of the network. Therefore, the CORE framework provides a unified and extensible structure in which different optimization objectives are handled through coordinated AI-driven modules rather than isolated solutions.

2.4 Assumptions and Constraints

The proposed CORE framework is developed under several fundamental assumptions to maintain analytical tractability and highlight its cooperative design principles. All entities are interconnected through wireless multihop links and can participate in data forwarding and information exchange. Each device is working in the FD model, which means a device can transmit and receive at the same time. Each device is capable of both data transmission and relaying. The communication process is assumed to operate under perfect channel state information (CSI) for link adaptation, routing, and power

control. Interference among nodes is modeled using the SINR, while self-interference and hardware imperfections are ignored for simplicity, assuming that sufficient interference cancellation and synchronization mechanisms are applied. Each node is equipped with an adaptive transmission capability that supports power control and cooperative relaying. Furthermore, queuing delay, synchronization errors, and the signaling overhead required for AI model updates are omitted to focus on the theoretical design and performance analysis of cooperative schemes.

Under these assumptions, several constraints are imposed to ensure the operability of the framework and the compatibility of the three cooperative schemes. First, all nodes are required to support the sensing and information collection functions used by the sensing and allocating module. Parameters such as node position, ALM, transmit power, packet size, interference level, and link quality must be measurable or obtainable either locally or through neighbor broadcasting. Second, all servers and devices are assumed to follow a unified time reference for scheduling and switching operations, which enables synchronous execution during cooperative transmission, server allocation, and power adjustment. Third, server allocation is assumed to complete within one decision epoch, meaning that each device must be associated with one server at a time.

Additional constraints arise from the functional requirements of each scheme. For the eCAP scheme, at least one device must be reachable to form the FG-based spanning structure, and node mobility during a single transmission cycle is assumed to be negligible. For the eLOW scheme, servers must possess sufficient computational resources to support BLS-based decision models and store training updates. For the uPOW scheme, nodes are assumed to be capable of adjusting their transmit power continuously or by predefined discrete levels to enable CTPC convergence.

As this dissertation represents a foundational study of the proposed CORE framework, all simulations and theoretical derivations are conducted under TDMA. This choice removes randomness introduced by contention-based access and allows us to isolate and analyze the performance gains produced solely by the cooperative AI-driven decision modules. In realistic wireless environments, other multiple access mechanisms, such as CSMA, OFDM, and NOMA, introduce channel contention, subcarrier scheduling, and power-domain multiplexing. Supporting these mechanisms requires additional control logic for

grant-free access, collision resolution, and user separation, which significantly increases the complexity of cooperative learning and real-time optimization. Future extensions of the CORE framework will incorporate these advanced radio access mechanisms, and the modular scheme design ensures that additional control modules can be integrated without revising the upper-layer learning architecture.

Although these assumptions and constraints simplify implementation and enable theoretical analysis, they do not limit the extensibility of the CORE framework. In practical deployments, several relaxed conditions, such as imperfect CSI, asynchronous signaling, mobility-induced topology changes, or heterogeneous transmission capabilities, can be incorporated progressively in future studies. The modular design of CORE ensures that extensions can be made either by modifying the learning models or by integrating new cooperative schemes without altering the protocol stack.

2.5 System Feasibility and Practical Considerations

This dissertation proposes a cooperative AI-driven communication framework for WMNs with edge computing. To ensure the feasibility of the proposed system in practical deployments, the framework is designed with explicit consideration of architectural scalability, computational complexity, and compatibility with existing wireless and edge infrastructures.

First, the CORE framework adopts a modular architecture with an orchestration system (OS) that performs network-wide decision-making. In the uPOW scheme, server selection, multihop path selection, and transmit power control are jointly optimized at the OS based on network information collected via edge servers. Therefore, the proposed framework avoids requiring each device to solve complex optimization problems locally, while still supporting scalable operation by leveraging the OS and edge servers for coordination and policy distribution in dense and dynamic wireless multihop environments.

Second, the computational complexity of the learning components is kept manageable. BLS employed in server allocation adopts a broad network structure with fast training and inference, which is well suited for edge computing environments with limited computational resources. The FG-based topology evaluation and Q-learning operate on discrete decision spaces and are executed periodically rather than continuously, ensuring that the

overall processing overhead remains practical for real-time operation.

Third, the proposed system is compatible with existing wireless and edge computing architectures. The optimization mechanisms are implemented as higher-layer control functions and do not require modifications to physical-layer signaling or standardized MAC procedures. As a result, the CORE framework can be incrementally deployed on top of current wireless multihop and MEC-enabled systems, supporting practical adoption in B5G and future 6G scenarios.

2.6 Deployment Conditions

The deployment of the proposed CORE framework relies on several practical conditions that are commonly satisfied in edge computing–enabled wireless multihop networks.

First, in uPOW the network is assumed to support a multi-server edge computing architecture, where edge servers are interconnected with an OS. The OS is responsible for collecting network status information from edge servers and distributing coordinated optimization decisions to network nodes.

Second, wireless nodes are assumed to be capable of basic link-quality measurement, such as estimating SNR or SINR, which are required for routing evaluation and power control. These measurements are standard functionalities in contemporary wireless systems and do not require specialized hardware.

Third, nodes are assumed to support adjustable transmit power within a predefined range. This assumption is consistent with existing wireless standards, where transmit power control is commonly available to manage interference and energy consumption.

Fourth, network status information, including topology changes, traffic demand, and interference levels, is assumed to be periodically reported to the OS through the edge infrastructure. The proposed framework does not rely on instantaneous or perfectly accurate global information; instead, it operates based on aggregated and periodically updated measurements.

Finally, the proposed framework is intended for static or moderately mobile wireless multihop environments, such as emergency communication networks, industrial IoT, or edge-assisted sensing systems. Highly dynamic scenarios with extremely fast topology changes are beyond the current scope of this dissertation and are considered as future

work.

2.7 Summary

This chapter has presented the CORE framework, which serves as the architectural foundation of this dissertation. First, the key enabling technologies were reviewed to clarify how each technology contributes to the design principles of the proposed framework and how they collectively relate to the core research challenges of latency, capacity, and energy efficiency.

Second, the overall architecture of CORE was introduced in detail. The interactions across protocol layers were explained, from application-driven communication requests to transport selection, routing formulation, and finally AI-driven cooperative optimization in the OWA Layer. The internal operation of CORE was also described, together with the roles of its supporting components, such as the sensing and allocating module and the closed-loop feedback interaction with lower physical-layer measurements.

Third, the three cooperative AI schemes that compose the CORE framework were summarized. The eCAP scheme addresses capacity through FG-assisted root node selection and Q-learning-based path selection. The eLOW scheme tackles the latency problem in multi-server environments through BLS-based server allocation. The uPOW scheme integrates BLS, Q-learning, and CTPC to mitigate interference and reduce energy consumption in dense deployments. Together, these schemes form a progressive and unified optimization chain that enables adaptive and efficient wireless communication under diverse network conditions.

Finally, the assumptions and constraints of the CORE framework were outlined to establish the theoretical foundation on which subsequent chapters are developed.

Based on the architectural structure presented in this chapter, the next three chapters provide an in-depth treatment of each cooperative scheme, covering their system models, algorithmic designs, and performance evaluations.

Chapter 3

Efficient Capacity Management Scheme

Following the introduction of the CORE framework in Chapter 2, this chapter presents the first proposed scheme, eCAP, which serves as the capacity-oriented optimization engine in the sharing and deciding modules of the CORE pipeline. The primary goal of eCAP is to mitigate the degradation of network capacity caused by dense interference and the difficulty of forming high-throughput routes in WMNs.

Although multihop relaying improves coverage and spectrum reuse, it also creates strong mutual interference when many neighboring links transmit concurrently. Even when multiple routing options exist, only a very limited subset of paths can sustain consistently high throughput; hidden terminals, bottleneck relays, and spatially correlated interference often cause unexpected throughput drops along seemingly short routes. Consequently, the aggregate capacity of WMNs remains low under dense deployments, as discovering and maintaining truly high-throughput multihop paths is extremely challenging. Traditional routing strategies based on hop count, static link cost, or fixed heuristics neglect interference coupling and dynamic link variations, and therefore fail to maximize throughput in realistic deployments.

To overcome these limitations, the eCAP scheme formulates high-throughput routing as a learning-enhanced cooperative transmission problem. The scheme integrates three complementary components: (i) a FG approach that constructs a convergent tree-structured topology and selects a root node; (ii) a two-stage Q-learning path selection

algorithm that first discovers promising links based on SNR and then refines decisions using SINR to explicitly consider interference; and (iii) a CoF strategy that enables paired relay nodes to forward simultaneously toward a common parent, increasing spatial reuse and reducing the number of required time slots.

By combining these mechanisms, eCAP provides an AI-assisted cooperative transmission architecture that significantly enhances route throughput and overall network capacity while maintaining transmission reliability.

It should be noted that the core ideas in this chapter are based on the master's work, which was originally reported in [17].

The remainder of this chapter is organized as follows. Section 3.1 reviews the related research. Section 3.2 introduces the system model of the proposed eCAP scheme. Section 3.3 presents the complete algorithmic design of eCAP, followed by performance evaluation and numerical results in Section 3.4. Finally, Section 3.5 concludes this chapter.

3.1 Research Background

It is expected that approximately 65% of the world's population will be using the 5G networks by 2026 [18]. In other words, it means the number of people using mobile devices such as smartphones, smartwatches, and so on, will drastically increase in use. As the next generation, 6G communication is expected to provide better services for users than 5G, such as a larger network coverage, higher throughput, and to accommodate a larger number of users and lower latency communication at the same time. Some novel technologies will be applied to 6G, including extremely large bandwidth (more than 1 THz waves) and AI empowered, including network analysis, network resource management, and network planning.

Meanwhile, 6G network is expected to attend a remarkable revolution in Internet. This revolution will completely differentiate 6G from previous networks and evolve wireless communication from the "Internet of Things" to "Internet of Intelligence" [18]. Particularly, 6G needs to support ubiquitous AI services from the cloud servers to the edge devices and beyond the specification of current mobile networks. AI will promote the development of 6G in designing and optimizing architectures, protocols, and operations.

6G is a research field for serving data transmitting services by applying all new tech-

niques. These various techniques are expected to be applied in 6G in the future, like a full-duplex system, a distributed massive multi-input multi-output (MIMO) system, a reconfigurable intelligent surface (RIS), and so on. Also, G. Trivedi et al. [19] explained that wireless multihop networks are also one of the core technologies in 6G, which can decentralize the computations, organize the network automatically, reach high network capacity, and be deployed in a short time.

WMN can form a self-configuring network by cooperating nodes and allows any two distant nodes to send their packet directly with the help of other nodes over a long distance. In other words, the WMN can extend the network coverage. However it can drastically degrade the entire network capacity due to a source node choosing an uncertain path to send its packet via the way of multihop communication.

In details, the first problem is happened when the number of nodes increases, there are many data transmission paths via different intermediate nodes to be selected from a source node to the corresponding destination node by an efficient path selection algorithm. Since the criteria and conditions of each path are different, the result of the path selection has a great influence on the entire network capacity. The second problem occurs when each node has to send not only its own packets, but also other nodes' packets in the network, in which it can lead to high latency for the sent packet to reach its destination node. Thus, the latency due to the queuing time and the processing time will increase drastically when the number of nodes is large.

To mitigate these two problems, we consider a RL with both the FG and nested lattice code (NLC) approaches in our research study. For the first aforementioned problem, we apply the eCAP scheme, which allows each source node to select its intermediate node with the best path to the corresponding destination node. With these best-metric paths, the resultant network topology that can achieve the optimum network capacity can be established. To solve the heavy iteration problem in the RL, we use FG to select the best root node of the network topology. As a result, FG can reduce the computation time of the DRL. For the second problem, NLC, which is used to reduce the link error probability of a channel, can correct the errors through the integration of the CoF strategy. By the way, the entire network capacity can be improved further by reducing the required number of time slots. In the eCAP scheme, we manage to propose two novel learning path selection

(LPS) algorithms, i.e., SNR-based learning path selection (NLPS) algorithm that focuses on increasing the E2E throughput from each source node to one destination root node, considering SNR as a reward. SINR-based learning path selection (INLPS) algorithm that uses the results obtained from NLPS and considers SINR as a reward during the training phase of RL to look for viable, appropriate path for each source node. In this dissertation, our main contributions are as follows:

- Proposing a novel eCAP scheme to achieve an optimum network capacity while reducing the computation time in WMNs
- With the introduction of FG approach, the eCAP scheme not only selects the best root node of a WMN, but also reduces the computation time of the RL
- Applying an NLC into the data transmission of a tree-based topology network, the results eCAP scheme can reduce the total number of transmissions in the entire WMN

3.1.1 Related Works

An extensive research works on the network capacity in wireless networks. These research works can be mainly divided into three categories, i.e., analytical modeling, network routing, and transmit power control. In analytical modeling, C. Fujimura et al. [20] proposed an analytical expressions for maximum E2E throughput varying number of hops and payload length in the string topology network. Besides that, S. Rezaei et al. [21]considered a routing policy and nodes' distribution as well as the MAC layer together into an analytical modeling of E2E throughput. From the viewpoint of network routing, W. Lee et al. [22] and J. Gui et al. [23] jointly consider node fairness and energy saving when designing the routing policy for wireless networks. Some researchers also look into the routing problem in the tree-based topology network. For example, D. Eliiyi et al. [24] proposed a parallel algorithm to find all root nodes of a network, in which the root node can considerably reduce the overall energy consumption and increase the network lifetime. Through this algorithm, the root node can greatly reduce the computation time as well. Recent studies have also investigated tree-based routing and topology optimization in wireless multihop and mesh networks under interference-aware formulations. For example, Xu et al. [25]

proposed the Minimum Interference Steiner Tree (MIST), in which multicast routing is modeled as a Steiner tree optimization problem with explicit interference considerations. A two-stage submodular relaxation algorithm (TSSR) is developed to construct multicast trees that simultaneously minimize tree cost and interference, and theoretical approximation guarantees are provided. These approaches demonstrate the effectiveness of tree-structured network models and interference-aware optimization in wireless mesh networks.

A few of research works on the network capacity analysis using FG approach and coding theory. Using the FG approach, Y. Mao et al. [26] studied a low-complexity algorithmic framework for link loss monitoring in a centralized manner in wireless sensor networks. The proposed algorithm iteratively updates the estimates of link losses upon receiving or detecting the loss of recently sent packets by the sensors. Similarly, W. Li et al. [27] focused on the nonparametric variant of the sum-product algorithm (SPA), called sequential particle-based SPA (SPSPA), for FG to infer the multi-sensor target states over time in a distributed manner of wireless sensor networks. Both studies show the great achievement of FG in wireless networks. In recent year, C. Jiang et al. [28] applied FG into the real smartphone navigation system, i.e., pedestrian dead reckoning exploring human walking gaits, in which FG can effectively solve practical application problems in the wireless communication. In the coding theory, X. Bu et al. [29] applied the network coding in the WMN to solve maximum-minimum optimization problem of cooperative communication. Their research work can significantly increase the capacity of wireless networks. Besides that, they also considered jointly optimizing intermediate node selection, scheduling, and flow routing for cooperative communication in the WMN environment. As for the lattice coding theory, there are no studies that apply to the WMN environment. But according to J. Xue et al. [30], the lattice decoder can achieve low word error rate for power-constrained wireless communications.

With the popularity of AI, there have been many studies on wireless networks empowered with AI in recent years. J. Rosenberger et al. [31] applied a deep reinforcement learning multi-agent system in different devices to decentralize the calculation to locate the resources in the industrial Internet of Things. Even though the systems and resources are keep changing, their method runs very well and time is very low, which inspired our

research to use DRL to reduce computation time.

In recent developments, Y. Wang et al. [32] proposed a multigranularity DRL for reliability optimization in channel resource allocation in the WMN. Their approach, which combines mobile-edge computing and DRL, addresses the challenge of highly reliable data transmission in complex multihop and multichannel environments. This research significantly contributes to the field by enhancing network performance and reliability using advanced computational techniques. In a similar vein, T. Ho et al. [33] developed the DRL for optimizing server selection, cooperative offloading, and handover in MEC environments within 5G networks, significantly enhancing network efficiency and reducing computation costs.

Among all kinds of AI, DL is one type that has seen numerous applications [34, 35] in recent years. RL is another type that is the most suitable for solving path selection problems, and some researchers have applied it to the WMN. D. A. Dugaev et al. [36] presented an application of RL-based algorithms to the routing task in wireless multihop topologies, and a flexible, reliable, adaptive packet forwarding scheme was developed, which showed significantly better results in packet loss ratio and route recovery time values, compared to the classical routing approach, widely used in the WMN. RL has many variant algorithms. Among them, Q-learning, as a model-free RL algorithm, processes data without an environment model and adaptation, is widely used in the path selection problem of wireless networks. Researchers have applied Q-learning to various situation. One example focuses on the interference channel like T. Wongphatcharatham et al. [37], who proposed the multi-agent Q-learning to optimize the transmit power of transmitters within interference channel by maximizing SINR. Their results show that transmitters are able to allocate their own transmit powers and get a better sum-rate than the traditional methods, such as the maximum power allocation and the random power allocation. Another example focuses on the E2E transmission rate like X. Wang et al. [38], who proposed a Q-learning-based intermediate nodes selection algorithm to decentralize the computation of nodes in the multihop clustered networks and result in a near-optimal E2E rate and better performance than traditional decentralized solutions from one source node to one destination node. Other example focuses on the intermediate nodes selection in cooperative wireless sensor networks like Y. Su et al. [39] developed a

deep-Q-network-based scheme, which significantly enhances system capacity and energy efficiency in cooperative communications.

3.1.2 Motivation

From the aforementioned studies, it is evident that the research community has made significant progress in analytical modeling, routing optimization, and transmit power control in WMNs. However, several limitations remain unresolved, motivating the objectives of this study.

First, most existing analytical models focus on evaluating E2E throughput or transmission capacity under fixed topology assumptions. Although these models provide useful theoretical insights, they often overlook the dynamic and decentralized characteristics of modern WMNs, where node mobility, link instability, and interference patterns fluctuate in real time.

Second, while numerous routing algorithms have been proposed—ranging from fairness- or energy-aware schemes to cooperative and hierarchical strategies—most rely on static or heuristic optimization. These methods struggle to maintain optimal performance in large-scale, dynamic, and heterogeneous networks, especially when multiple users transmit data concurrently and cause mutual interference.

Third, current transmit power control mechanisms achieve notable improvements in interference mitigation and throughput. However, they often require global coordination or centralized information exchange, which is impractical for distributed and large-scale WMNs.

In parallel, emerging research has applied AI to enhance routing and resource allocation in WMNs. These methods have shown promising results in improving adaptability and efficiency. Nevertheless, they typically treat communication links and nodes as independent entities, failing to capture the underlying probabilistic dependencies and network-wide interactions that influence overall performance.

Meanwhile, the FG and NLC approaches have demonstrated strong potential in other wireless communication domains for probabilistic inference and capacity enhancement, yet their integration into multihop networks and AI-driven frameworks remains largely unexplored.

To address these research gaps, this chapter aims to investigate the integration of FG and NLC into Q-learning to jointly optimize path selection and network capacity in WMNs. By combining the low-complexity inference capability of FG, the structural coding efficiency of NLC, and the adaptive decision-making of Q-learning, we seek to develop a novel hybrid framework capable of achieving high E2E throughput, reduced interference, and improved network scalability under dynamic wireless environments.

3.2 System Model

In this section, we describe the system model, which includes the network model, channel model, interference model, network capacity calculation, and airtime link metric.

3.2.1 Network Model

In eCAP scheme, the network model is a WMN, which has only one root node. We assume that the network topology of the WMN can be modelled as a graph network model. In this model, all nodes have the same transmit power, bandwidth, antenna gain, and received signal strength indicator (RSSI). Also, nodes are stationary, and each node must have a connection to other nodes. For transmitting data, a node can send to or receive from only one other node in a single time slot. A WMN example of network model that is used in this chapter is shown in Figure 3.1. In the network model, we assume that all nodes are working in full-duplex mode. Among these nodes, one node will be selected as the root node, and it will be connected to the wired connection, like Ethernet. Other nodes will be connected to the root node either directly or in a multihop manner.

3.2.2 Channel Model

In channel model, the wireless nodes are randomly and uniformly distributed in the coverage area, and the distance between two wireless nodes i and j (d_{ij}) is computed by the Euclidean distance.

The signal attenuation of the transmission link over a communication channel depends on the log-distance pathloss model. By assuming PL_0 as the Friis free space model, the

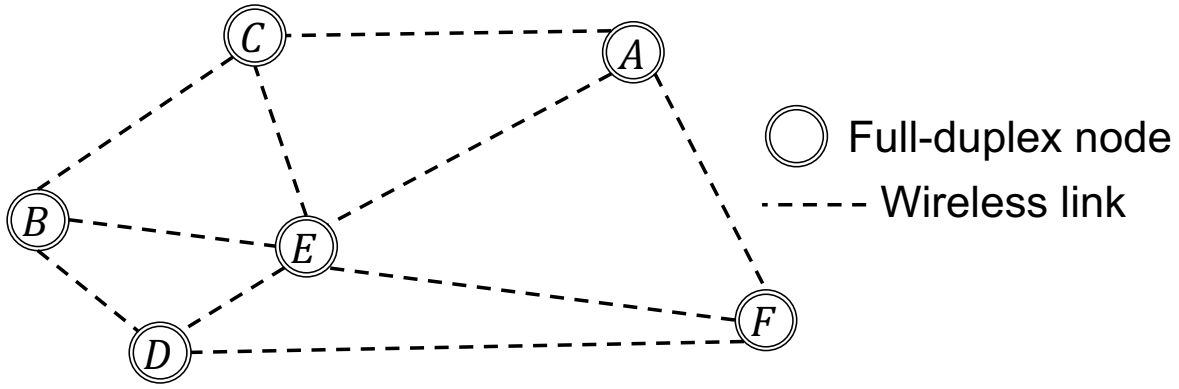


Figure 3.1: An example of WMN network model.

channel gain in terms of decibel dB unit between a transmitting node i and a receiving node j is shown as

$$PL_{ij} = PL_0 + 10 \cdot \zeta \cdot \log_{10}(\frac{d_{ij}}{d_0}) - \omega_{ij} + \psi \tag{3.1}$$

where PL_0 is assumed as Friis free space model, $PL_0 = 20 \cdot \log_{10}(d_0)$. β is attenuation constant and W_{ij} is the wall attenuation. d_0 is decorrelation distance which is set to ten meters in this research. ψ is shadowing attenuation. The power ratio at the receiving node j with the signal attenuation between wireless node i and node j according to the pathloss PL_{ij} is

$$G_{ij} = \frac{1}{10^{\left(\frac{PL_{ij}}{10}\right)}} \tag{3.2}$$

3.2.3 Interference Model

The interference model is used to determine the amount of power level of interfering nodes in an ongoing transmission, and the power capture model is one of them. In the power capture model, the SINR is considered to indicate whether the reception of the ongoing transmission is successfully achieved or not. In this dissertation, we refer to the power capture model, which is similar to [40] and define it as SINR-based interference model to obtain the set of interfering nodes of the ongoing transmission.

Figure 3.2 depicts the SINR-based interference model. Node i is sending its packets to node j. At the same time, nodes 1 to k are also sending packets to other nodes besides node i and node j, where $\mathcal{K}$ will be the total number of interfering nodes in the network of WMN. The SINR at node j from node i is given by

$$SINR_{ij} = \frac{G_{ij} \cdot P_i}{\eta_j \cdot B + \sum_{k \in \mathcal{K}, k \neq i} G_{kj} \cdot P_k} \tag{3.3}$$

where P_i is the transmit power of node i and η_j is noise level of node j, and B is the channel bandwidth.

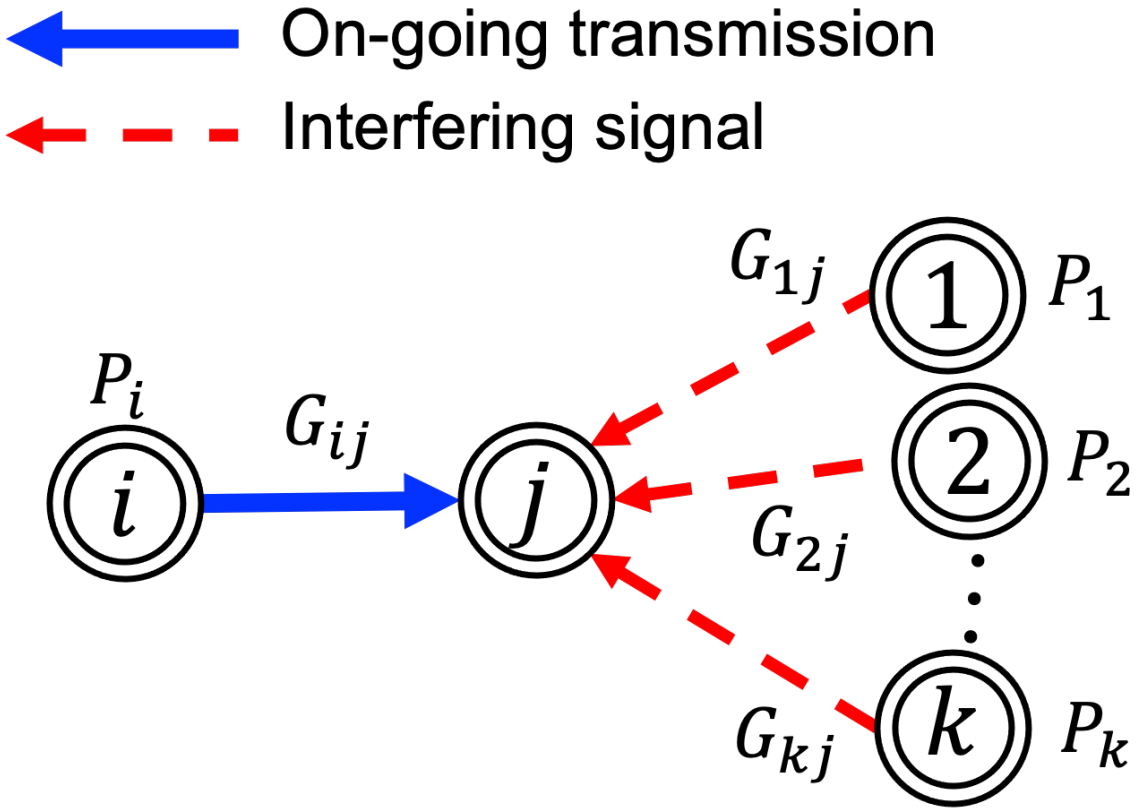


Figure 3.2: Illustration of SINR-based interference model.

3.2.4 Link Capacity Model

We can obtain the link rate from node i to node j as

$$R_{ij} = B \cdot \log_2 (1 + SINR_{ij}) \tag{3.4}$$

Then, the E2E throughput (R_{SD}^{e2e}) from the source node S to the destination node D is defined as the sum of the link rates of all links on the path, which is

$$R_{SD}^{e2e} = \sum_{m=0}^{M} R_{n^m \; n^{m+1}}, n = \{1, 2, \cdots, N-1\} \tag{3.5}$$

where N represents the total number of nodes in the uploading path from S to D. n denotes one of the intermediate nodes on the path from S to D, including S. The network capacity ($\mathcal{C}$) is computed based on the link rate and required time slot of all the on-going transmissions. For each link, the transmission time of a time slot (T_v) is defined as the maximum length of transmission time of all the involved links, which are transmitted the packets in the same time slot, and is given as

$$T_v = \max_{i \in \mathcal{I}} \left\{ \frac{L}{R_{ij}} \right\} \tag{3.6}$$

Therefore, the network capacity is the sum of the sizes of all packets sent divided by the sum of the transmission time of all time slots, which is

$$\mathcal{C} = \frac{(N-1) \cdot L}{\sum_{v=1}^{V} T_v} \tag{3.7}$$

where v is the number of time slots, L is packet size and $\mathcal{I}$ is the set of nodes sending packets in the same time slot. V is the total number of time slots. For the sake of simplicity in the evaluation part, we assume that each node has only one packet to be sent to the destination root node.

3.2.5 Airtime Link Metric

Airtime reflects the amount of channel resources consumed by transmitting the frame over a particular link, and the extensive framework allows this metric to be overridden by any path selection metric as specified in the mesh profile. The IEEE802.11s mesh WLAN specification of airtime link metric (ALM) [41] can captures the link quality as a function of the estimated frame loss probability as follows:

$$l = \left(\mathcal{O} + \frac{L_{test}}{R_0} \right) \frac{1}{1 - FER} \tag{3.8}$$

where l is the airtime cost with no units, $\mathcal{O}$ is channel access overhead, which includes frame headers, training sequences, access protocol frames, and so on. Here $\mathcal{O}$ equals to the sum of PHY header, MAC header, acknowledgement (ACK), distributed coordination function inter-frame space (DIFS), short inter-frame space (SIFS), slot time and minimum of contention window (CW_{min}). L_{test} is size of test frame, R_0 is basic rate for test packet and FER is frame error rate (FER). ALM can use a numerical value to express the quality

of the link. l is small means the quality of the link is better. For example, a node send a test frame (8192 bits) through a link with a data rate of 1 Mbps. The channel access overhead is 65 μs, PHY header is 192 μs, ACK is 304 μs and slot time is 9 μs. The total time is 570 μs. If the frame error rate is 80%, the airtime cost (l) is 4278.3 (rounded to 4280). The airtime cost is used by FG as a measure of the link quality between two communication nodes.

3.2.6 Network Optimization Targets

The objective is to determine which wireless links and intermediate relay nodes should be selected to form an E2E transmission path between a source node and its destination. From a network perspective, the path selection process constitutes a discrete decision-making problem, in which each candidate wireless link is either selected or excluded from the constructed transmission path. Different choices of transmission paths lead to different E2E throughputs, interference levels, and achievable transmission capacities.

It should be emphasized that transmission power, bandwidth allocation, and other physical-layer parameters are not directly optimized in this chapter. Instead, their impact is implicitly captured through the reward evaluation of different path selection outcomes under dynamic network conditions.

Accordingly, the scheme developed in this chapter learns effective path selection policies by interacting with the network environment, aiming to maximize the network capacity.

3.3 Efficient Capacity Management Scheme

In this section, we propose and describe the novel eCAP scheme. The structure diagram of the eCAP scheme is shown in Figure 3.3. In this WMN environment, various wireless communication devices, such as smartphones, robots, laptops, vehicles, and so on are connected to each other wirelessly. The information of these devices is sent via a multihop fashion can be efficiently determined by the proposed eCAP scheme, which consists of three parts. In the root selection part, a shortest path spanning tree (SPST) algorithm and an FG approach are adopted to select the best root node of the tree-based network

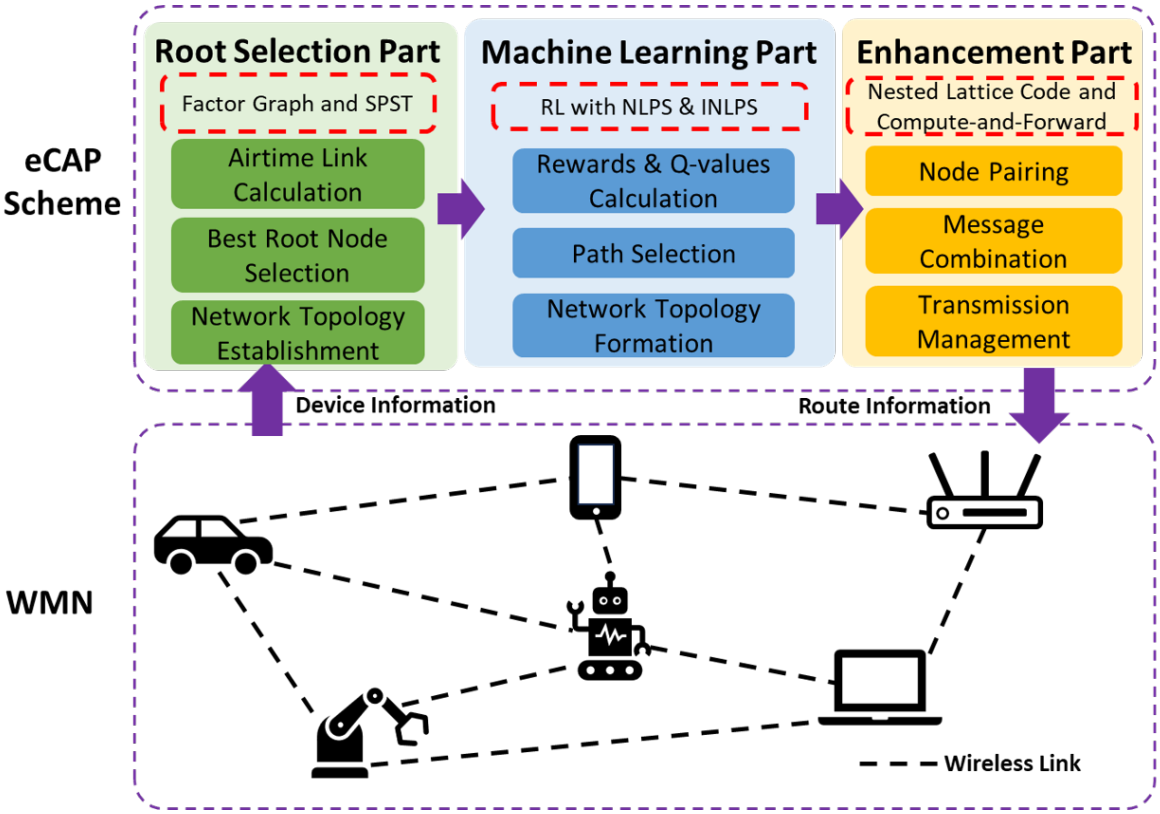


Figure 3.3: The structure diagram of eCAP scheme.

topology using the ALM metric. Next, in the learning part, two LPS algorithms based on Q-learning, i.e., NLPS and INLPS algorithms are used to select the best path for forwarding the packets of each devices. Last, in the enhancement part, NLC and CoF are applied to manage and combine the data transmission of two pairing nodes to not only improve the reliability of data transmission in the presence of noise and interference, but also reduce the number of time slots. The application scenario of the proposed scheme in this dissertation is expected for the network system of the large area coverage, such as exhibition halls, stadiums, and so on, in which the communicating devices are quite static and form a single root tree topology for their data transmissions.

3.3.1 Factor-graph

An FG approach is a bipartite graph representing the factorization structure of a global function into a product of smaller local functions; each local function contains the product from the other factors. In particular, FG has two types of nodes: variable nodes and factor nodes. Variable nodes can be either evidence variables when their value is known, or query

variables when their value should be predicted. Factor nodes define the relationships between variable nodes in the graph and represent functions on subsets of the variables. Each factor node can be connected to many variable nodes and comes with a factor function to define the relationship between these variables. Each factor function has a weight associated with it, which describes how much influence the factor has on its variables in relative terms. The weight can be learned from training data, or assigned manually. Because many optimization problems in robotics have the locality property, FG can model a wide variety of problems across the fields of AI and robotics. Using the sum-product algorithm, the global function can represent the whole FG, which can determine the best network capacity from a tree-based structure topology.

In the LPS algorithms, the role of FG is to select the best root node, which is most likely to generate high network capacity for the LPS algorithms, thereby reducing the number of iterations of the LPS algorithms. FG selects the best root node through the following three steps:

(1) For each node as the root node, use Dijkstra's algorithm to find the SPSTs;

(2) FG uses sum-product algorithm to calculate the total ALM link weight of SPSTs; and

(3) Compare the total ALM link weight for all the different root nodes and select the best root node with the smallest weight.

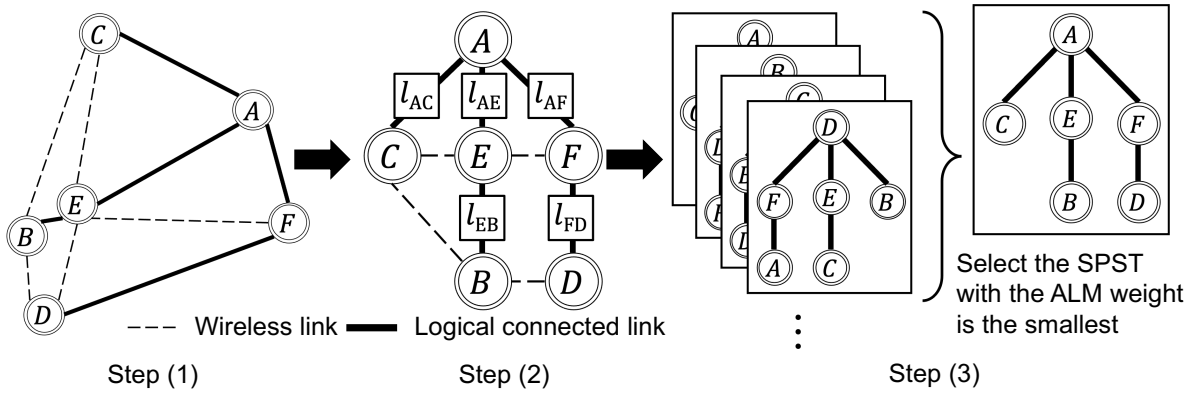


Figure 3.4: SPST computation for node A by Dijkstra's algorithm.

In Step (1), the nodes and links in WMN can be represented into an undirected graph. As shown in Figure 3.4 of Step (1), for each node in WMN as the root node, Dijkstra's

algorithm is used to find the path from the root node to all other nodes. Dijkstra's algorithm is an algorithm for finding the shortest paths between nodes in a graph. If the link is selected by Dijkstra's algorithm, it becomes a logically connected link with a weight calculated by ALM. With this algorithm, the shortest path from the root node to all other nodes is selected to form a tree-based topology. This topology is called SPST topology, which has minimum weight paths from the root node to all the other network nodes. Each node in WMN can be used as the root node and generate its own SPST topology, so the number of nodes is consistent with the number of SPST topologies.

Next, in Step (2), the total ALM link weight of each SPST topology is calculated. The weight of each link is represented by the airtime cost, which is calculated by ALM. The lower the value, the better the quality of the link. In general, the total ALM link weight is obtained by adding the weight of each link, and this is also done in Dijkstra's algorithm. But this way is difficult to distinguish which SPST topology is better. Besides that, in the actual network environment, the structure of the network will also have a great impact on the performance of the entire network. Therefore, the simple summation method cannot accurately and completely represent the quality of the network. Instead of this method, in this dissertation we regard the SPST topology as an FG, and use the sum-product algorithm to calculate its total ALM link weight. The sum-product algorithm performs addition and multiplication according to the relationship between nodes; therefore the total ALM link weight is also affected by the network topology. Compared to the simple summation method, the sum-product algorithm can better distinguish which is the best SPST topology. Step (2) in Figure 3.4 is applying FG, and the sum-product algorithm on the SPST topology of node A is shown. l isthe ALM cost of the link and can be calculated by ALM. In FG, we assume all the nodes are equal in the WMN environment, and the factor value of a node can be neglected. For the root node A, the relation between node E and node B is parent node and child node, so the product of node E is multiplied them together, which is $l_{AE} \cdot l_{EB}$. Similarly, the product of node GF is $l_{AF} \cdot l_{FD}$. Node C, node E, and node F are sibling nodes, so the product from them is adding them together, which is

$$GF(A) = l_{AC} + (l_{AE} \cdot l_{EB}) + (l_{AF} \cdot l_{FD}) \tag{3.9}$$

where $GF(A)$ is the global function of root node A. This global function represents to the total ALM link weight. The total ALM link weight is small means good performance of this network topology. It should be noted that the airtime link metric defined in Equation (3.8) evaluates link quality at the individual link level, whereas the total ALM metric computed via the factor graph represents a topology-aware aggregation of these link-level costs.

At last, in Step (3), after the ALM link weights of each node are calculated by the sum-product algorithm, these SPST topologies are compared to each other with their own total ALM link weights, and the smallest one is selected as the best one. As shown in Step (3) of Figure 3.4, by comparing the total ALM link weight of the SPST topology with its corresponding root node, the best-metric network topology with the selected best root node is selected.

3.3.2 Reinforcement Learning and Q-learning

RL is a research field of machine learning (ML), in which an agent learns via trial and error. This problem is often modeled mathematically as a Markov decision process. RL algorithm contains several elements: agent, environment, state s, action a, and reward r. The learning process of the agent starts by receiving the current state s_m from the environment. Agent is the main part of RL, which decides to do which action depending on the current state s_m and reward r_m. After the agent makes a decision with an action a_m, which is sent to the environment. The environment changes to the next state s_{m+1} with its associated reward r_{m+1} based on the action a_m and sends them to the agent again. Then, the agent makes the next decision again based on the received state and reward. By looping this process continuously, the agent tries to find a near-optimal policy by maximizing the expected cumulative reward to determine the best state-action value.

Q-learning, which is known as an independent model of the RL algorithm that is able to learn the action value for a particular state in the defined environment. The independent model of Q-learning does not rely on the model of the environment. Besides that, the model cannot only deal with the problem in the stochastic conditions, but also its reward does not depend on the dynamic change of adaptations or adjustments. F. S. Melo [42] explained that Q-learning is used to obtain an optimal state by maximizing the

expected value of the total reward over any and all successive iterations from the origin of the current state for any Finite Markov Decision Process (FMDP). In other words, Q-learning can search for an optimal action as the best selection policy for any given FMDP with the given infinite exploration time, and in the case of a fully-random policy or a partly-random policy. In Q-learning, 'Q' is a function that is also called the Q-function. This Q-function computes the expected value of the instantaneous reward for an action to be taken in the given state. In general, the Q-function can be written as:

$$Q^{new}(s_m, a_m) = (1 - \alpha)Q(s_m, a_m) + \alpha(r_m + \gamma Q_{max}(s_{m+1}, a)) \tag{3.10}$$

where s_m and s_{m+1} are current state and next state, a_m and r_m are action and reward, α and γ are learning rate and discount factor. The Q-values form the Q-table and are updated with each training. The agent decided the next action according to the Q-values in the Q-table. The Q-values keep updating until the reward reaches to the target value or the reward changes less than the threshold within several iterations.

Q-learning is widely used in the path selection problem. In this problem, nodes need to choose the next neighboring path according to their own situation until they reach the destination node, which is very well-suited to the operation method of Q-learning. In other words, the selection of intermediate nodes from the source node to the destination node can be defined as the path selection problem in Q-learning. Here, each node acts as an agent, and action refers to the next selected intermediate node. In each selection of an intermediate node as the neighboring path, the node needs to make a choice based on its own network environment and gets a reward according to the choice to learn and optimize the predefined policy; in this case, it is to maximize the objective parameter of the WMN.

3.3.3 SNR-based Learning Path Selection Algorithm

The aim of the NLPS algorithm is to focus on increasing the E2E throughput from any source node to a destination node, considering the parameter of SNR as the reward. NLPS algorithm is a modified version of the decentralized Q-learning based intermediate nodes selection scheme [38]. Since the SNR parameter is used as the reward, the NLPS algorithm will look for the path with the highest SNR value and ignore the interference

effect from other transmitting nodes.

In the beginning stage, when the network topology has been formalized, it is not yet possible to determine the result of the total interference level from other ongoing transmissions despite its own noise during the data transmission. Thus, SNR is used as a path decision-making for any network topology to establish its data transmission. To apply the NLPS algorithm, the nodes in WMN need to be layered according to the best root node, which is provided by FG, and the transmission range is determined by RSSI. As shown in Figure 3.5, all nodes are divided into M layers. m is the number of layers and f_m is the nodes in the mth layer. C_m means the mth layer and $n_{f_m}^m$ means the fth node in mth layer. In layer m, the number of nodes is F_m. According to the system model of this dissertation, all the nodes send packets to the root node D, forming to a tree-based topology. Nodes can only send to nodes in the higher layer, but they cannot send their packets to the nodes in the same layer. Next, the NLPS algorithm uses to determine the path selection for all the source nodes in the network is composed of three phases: initialization, training, and selection.

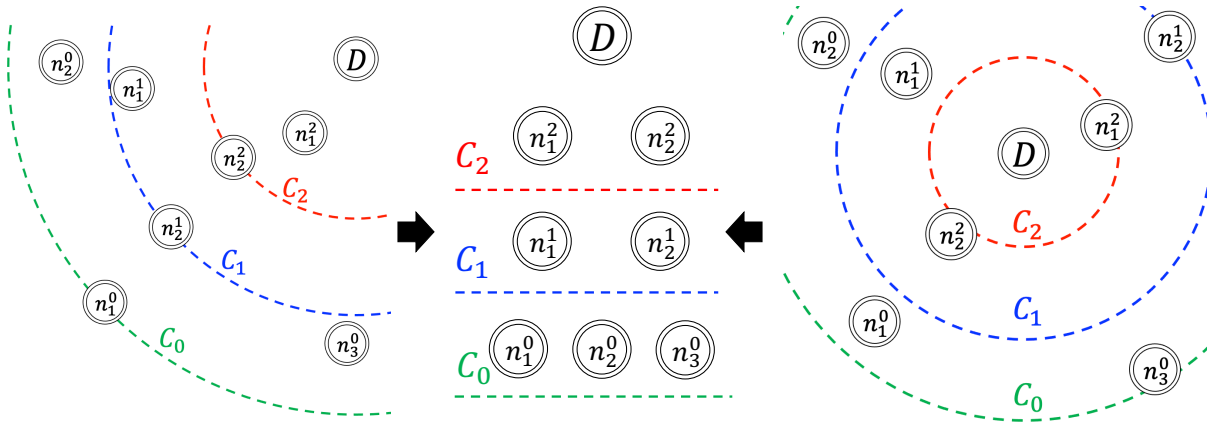


Figure 3.5: Nodes layered by transmission range in WMN.

In the initialization phase, $F_{m-1} \times F_m$ reward tables are generated. Each layer has its own reward table that contains the rewards of all possible pairs of state and action, which correspond to the pair of the transmitting nodes and the receiving nodes.

In the NLPS algorithm, the reward is set to the SNR value, and these reward tables can be obtained before the training phase. The reward table is described by Table 3.1. In this table, m is the layer of receiving nodes, and s_m and a_m mean states and actions, respectively. Since the nodes in the network are static, it simply means that the position

Table 3.1: Reward table for mth layer

s_m \ a_m	n_1^m	n_2^m	...	$n_{H_m}^m$
n_1^{m-1}	r_{11}^m	r_{12}^m	...	$r_{1F_m}^m$
n_2^{m-1}	r_{21}^m	r_{22}^m	...	$r_{2F_m}^m$
...	...	...	...	...
$n_{F_{m-1}}^{m-1}$	$r_{F_{m-1}1}^m$	$r_{F_{m-1}2}^m$	...	$r_{F_{m-1}F_m}^m$

of the nodes has been determined, so the SNR will not change, and the reward table does not need to be updated in the training phase.

Similar to the reward table, each layer has one Q-table, which stores the Q-value of all the possible pairs of state and action, and will be updated after each iteration in the training phase. In the initialization phase, we set every Q-values in every Q-table is zero to allow the agent to select its first action randomly. The Q-table is shown in Table 3.2. Here $Q_m(i,j)$ is Q-value of node i and node j in the mth layer (C_m). For example, $Q_m(n_1^{m-1}, n_2^m)$ means the Q-value of the first node in $(m-1)$th layer and the second node in the mth layer. This Q-value is stored in the Q-table of the mth layer.

Table 3.2: Q-table for mth layer

s_m \ a_m	n_1^m	n_2^m	...	$n_{F_m}^m$
n_1^{m-1}	$Q_m(n_1^{m-1}, n_1^m)$	$Q_m(n_1^{m-1}, n_2^m)$	...	$Q_m(n_1^{m-1}, n_{F_m}^m)$
n_2^{m-1}	$Q_m(n_2^{m-1}, n_1^m)$	$Q_m(n_2^{m-1}, n_2^m)$	...	$Q_m(n_2^{m-1}, n_{F_m}^m)$
...	...	...	...	...
$n_{F_{m-1}}^{m-1}$	$Q_m(n_{F_{m-1}}^{m-1}, n_1^m)$	$Q_m(n_{F_{m-1}}^{m-1}, n_2^m)$	...	$Q_m(n_{F_{m-1}}^{m-1}, n_{F_m}^m)$

The main task of the training phase is to update Q-values according to the Q-function until the agents terminate the training loop, which means the best path is found. Q-function is the key part of the training phase, which calculates Q-values depend on the current values and future values. The standard Q-function in equation (3.10) only considers the current reward, i.e., SNR of the current link. This will cause agents to only focus

on the reward of the current link, while ignoring the status of the entire path. This is similar to the greedy search, which is inefficient. If the SNR of the current link is very large while the SNRs of other links are small, the agent will still choose this link even though it is resulting a very low E2E throughput. Therefore, the Q-function should consider all links. However, this can lead to heavy computation time. To obtain a better result, we make a compromise choice. The Q-function will only select the larger one between the reward of the current link and next link. This makes the agent to consider the link that could result the higher E2E throughput. The modeled Q-function is given by

$$Q_m(s_m, a_m) = \begin{cases} (1-\alpha)Q_m(s_m, a_m) + \alpha(r_m(s_m, a_m) + \gamma Q_{max}(m+1)), \\ \qquad\qquad\qquad\qquad m \leq M, r_m(s_m, a_m) > r_{m+1} \\ (1-\alpha)Q_m(s_m, a_m) + \alpha(r_{m+1} + \gamma Q_{max}(m+1)), \\ \qquad\qquad\qquad\qquad m \leq M, r_m(s_m, a_m) < r_{m+1} \\ r_m(s_m, a_m), \qquad\qquad\qquad m = M+1 \end{cases} \tag{3.11}$$

where

s_m: the state represents the current position of a packet along the multihop transmission path, corresponding to an intermediate relay node in C_{m-1} that is preparing to forward the packet to the next hop in C_m

a_m: the action represents the selection of the next-hop relay node in C_m, which directly determines the wireless link used to extend the end-to-end transmission path. All candidate relay nodes in C_m are considered as feasible actions.

$r_m(s_m, a_m)$: the reward evaluates the quality of the selected wireless link from a network perspective. In eCAP scheme, the SNR or SINR of the selected link is adopted as the reward, reflecting its contribution to reliable transmission along the multihop path.

α: the learning rate controls the balance between newly acquired link evaluation information and previously learned knowledge, where $0 \leq \alpha \leq 1$.

γ: the discount factor determines the weight of future path quality in the Q-value update, and a larger γ encourages the agent to consider long-term path performance.

r_{m+1}: reward of next link in C_{m+1}, which is calculated by

$$r_{m+1} = r_{m+1}(a_m, a'_{m+1}) \tag{3.12}$$

where

$$a'_{m+1} = \underset{a_{m+1}}{\operatorname{argmax}}\ Q_{m+1}(a_m, a_{m+1}) \tag{3.13}$$

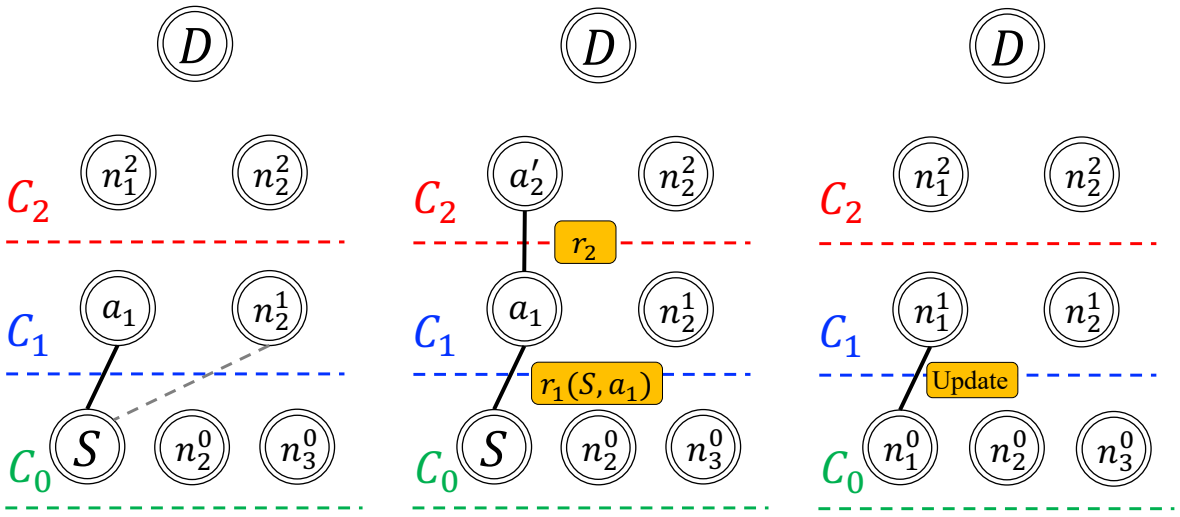
As in equation (3.13), a'_{m+1} is the best action at the state a_m in C_{m+1} with maximum Q-value. As the reward of the best action in the next link, r_{m+1} is compared with the reward of the current link, and the larger one is used to update the Q-value. Through this method, the agent cannot only be limited to the reward of the current link, but also take a long-term view and consider the next step to choose the path.

The maximum Q-value $Q_{max}(m+1)$ in C_{m+1} is calculated by

$$Q_{max}(m+1) = Q_{m+1}(a_m, a'_{m+1}) \tag{3.14}$$

where $Q_{max}(m+1)$ represents the Q-value of the best action in C_{m+1} and participates in the updating of Q-value as new information. The Q-function represents the long-term utility of selecting a specific next-hop relay at a given intermediate node, in terms of the overall quality of the resulting E2E transmission path.

Here we use the example in Figure 3.5 to explain how the Q-function works in the training phase. As shown in Figure 3.6, n_1^0 is set as a source node S. S needs to send packets to destination node (root node) D. $s_1 = S$, $m = 1$. First, a_1 is randomly selected from C_1. Here we assume that node n_1^1 is selected. $a_1 = n_1^1$. Next, the SNR of this link is the reward $r1(S, a_1)$ of doing the action a_1. From the Q-table of C_1 and C_2, we can find the best action at a_1 by equation (3.13) and assume that $a'_2 = n_1^2$. The SNR of this link is calculated as r_2. Also, $Q_{max}(2)$ is calculated by equation (3.14). The rewards of the current link $r_1(S, a_1)$ and next link r_2 are compared to each other to decide which equation will be used to update in equation (3.11). At last, the Q-value of n_1^0 and n_1^1 is updated and stored in the Q-table. Then, for the next layer C_2, the action node is set as a state, $s_2 = n_1^1$, $m = 2$, and repeat the same step as C_1, and the Q-value in the Q-table of C_2 is updated. In this training phase, the process repeats for each layer until $m = M + 1$, which means it has reached to the destination node D and one iteration is finished. If the change of E2E throughput from S to D is lower than the threshold ϵ, the training is stopped and completed. Otherwise, another iteration of training begins. As the number of iterations increases, the Q-table is gradually updated. When the training phase is completed, each layer has its own complete Q-table, including the updated Q-value.



(a) Random selection (b) Reward comparison (c) Update Q-value

Figure 3.6: Steps of updating $Q_1(n_1^0, n_1^1)$.

In the selection phase, the agent needs to find the best path from S to D through Q-tables after the training of all layers is completed. At first, the agent sets $s_1 = S$ and searches all Q-values in the row of s_1 in the Q-table of C_1 and finds the maximum Q-value and selects it as the best intermediate node a_2' from s_1. Next, it sets $s_2 = a_2'$ and C_2 repeats the step above to find the best intermediate node. After all layers have found their own best intermediate node, the best multihop path from S to D is determined.

After completing the three aforementioned phases, the best multihop path of each source node is determined, except for the root node. By combining all the best paths of all source nodes, a tree-based network topology with the optimum network capacity can be accomplished, as shown in Figure 3.7.

3.3.4 SINR-based Learning Path Selection Algorithm

In the NLPS algorithm, the total interference among the links has been ignored because the actual set of transmission links is not yet known before path planning begins.. Before establishing the transmission paths, it is not possible to determine which links will be active; therefore, the interference cannot be computed in advance. Therefore, NLPS performs an initial path selection using only the SNR of each candidate link, providing a preliminary routing structure. Once the initial multihop paths are determined, the total

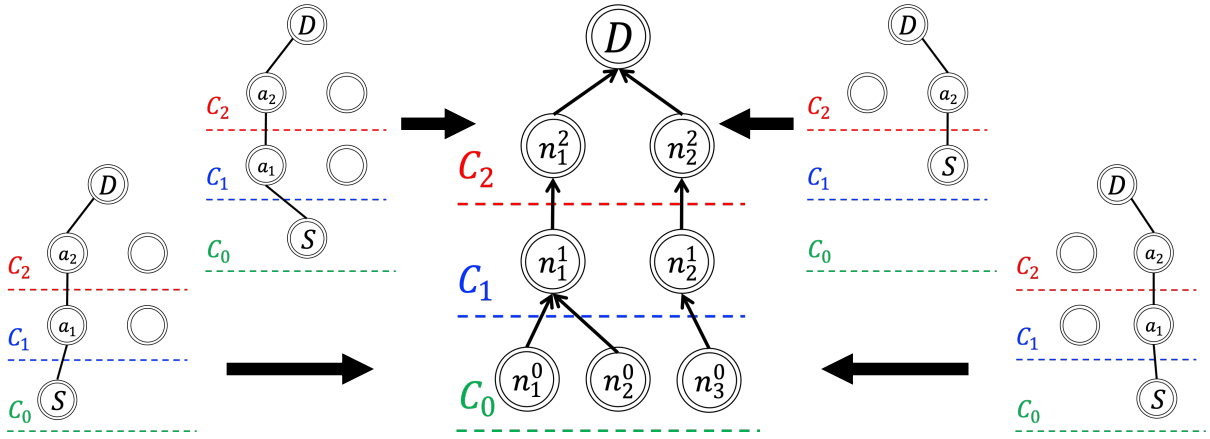


Figure 3.7: Forming the best network topology.

interference generated by concurrently active links can be recomputed. Since interference significantly affects link quality and achievable transmission rates, it must be considered during actual data transmission. To incorporate this essential network parameter, we introduce an enhanced learning-based path selection algorithm that explicitly considers interference. To achieve improved E2E throughput, we propose the INLPS algorithm. INLPS extends NLPS by refining the initial SNR-based routing decisions using SINR information. The objective of INLPS is to identify a more reliable transmission path with higher E2E throughput. INLPS algorithm is performed right after the NLPS algorithm. Using the routing structure generated by NLPS, the SINR of each link can now be computed based on updated interference information. Similar to NLPS, the INLPS algorithm consists of initialization, training, and selection phases. In these phases, the action still represents selecting the next-hop relay node, and the state denotes the current relay position along the multihop path. However, the reward is updated to reflect the SINR of the selected link, enabling the agent to evaluate both link quality and interference effects when extending the transmission path. First, INLPS imports the results of NLPS, including the identified root node and the learned Q-tables. After the NLPS algorithm finishes, the INLPS algorithm saves the root node and all the Q-tables. Then, the INLPS algorithm uses the network topology obtained by the NLPS algorithm to re-calculate the corresponding SINRs and store them in the reward tables. The algorithm then proceeds through the initialization, training, and selection phases following the updated SINR-based reward model. Consequently, the agent favors next-hop decisions that lead to higher-SINR paths, resulting in improved E2E throughput. Thus, incorporating SINR as the reward enables

INLPS to identify multihop paths with superior E2E throughput.

The flowchart of NLPS and INLPS are shown in Figure 3.8.

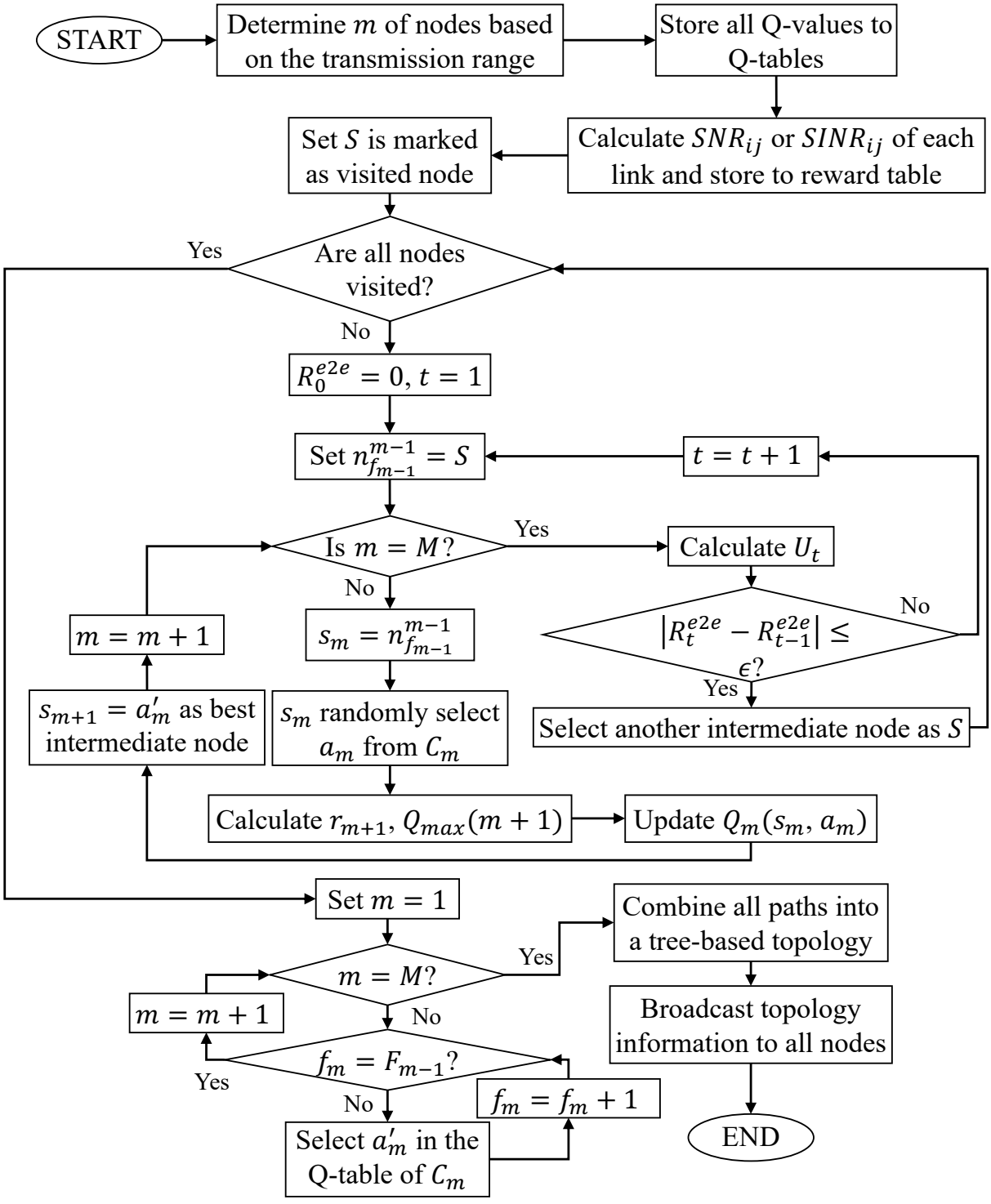


Figure 3.8: Flowchart of NLPS and INLPS algorithms.

3.3.5 Nested Lattice Code

Although the proposed LPS algorithms increase the E2E throughput in sending packets through the tree-based network topology, however this may result in poor network capacity when node density is very high in the network. This is because the total interference level increase drastically when all the nodes is sending their packets. To address this problem, the proposed eCAP scheme adopts NLC and CoF strategy.

Lattice code (LC) can be used to improve the data transmission in the presence of noise. LC is a kind of error-correcting code that enables messages to be transmitted through lattice points. Besides, LC ensures accurate encoding and decoding of messages, even if noise alters some bits of the codewords.

To improve the efficiency of data transmission, the NLC is formulated by using two lattices: a coding lattice and a shaping lattice. The coding lattice is responsible for the error-correcting capabilities of the LC. Meanwhile, the shaping lattice applies a power constraint to the codewords. In this dissertation, we focus on an 8-dimensional LC, specifically the E8 lattice code. The NLC is adopting a method to regulate the transmit power levels by mapping the original transmit power at the shaping lattice point to the constrained power level at the coding lattice point.

3.3.6 Compute-and-forward

The CoF strategy is employed to enable intermediate nodes to decode linear equations of transmitted messages, which are conveyed as noisy linear combinations by the channel. This strategy significantly reduces the number of required time slots when two nodes transmit their messages to the same parent node simultaneously. In the CoF strategy, codewords are linearly combined and transmitted through the channel. Even though in the presence of noise, these codewords can be accurately decoded at the receiver using NLC. In the CoF strategy, messages can be encoded to the corresponding codewords and transmitted over an additive white Gaussian noise (AWGN) channel; they become the receiving codewords, which are decoded to the receiving messages. Because the CoF strategy relies on codes with a linear structure, NLC is recommended to be used in the communication system [43].

In CoF, multiple transmitting nodes may simultaneously send lattice-coded signals,

and a receiving node decodes an integer-valued linear combination of the transmitted codewords rather than decoding each individual message. As a result, different relay nodes may obtain different linear combinations of the same set of transmitted messages, depending on their channel conditions.

By exploiting the linear structure of NLCs, these linear combinations can be reliably decoded at the receivers. Once a sufficient number of linearly independent combinations is collected, the original messages can be recovered by solving a system of linear equations over the underlying finite field. In the proposed eCAP scheme, this inversion process is performed implicitly when the decoded combinations are forwarded and aggregated along the multihop path, allowing the destination node to reconstruct the original information symbols.

In this dissertation, we combine two messages into one. In particular, NLC is used to reduce the error on the messages, whereas the CoF strategy is used to reduce the number of time slots in the data transmission.

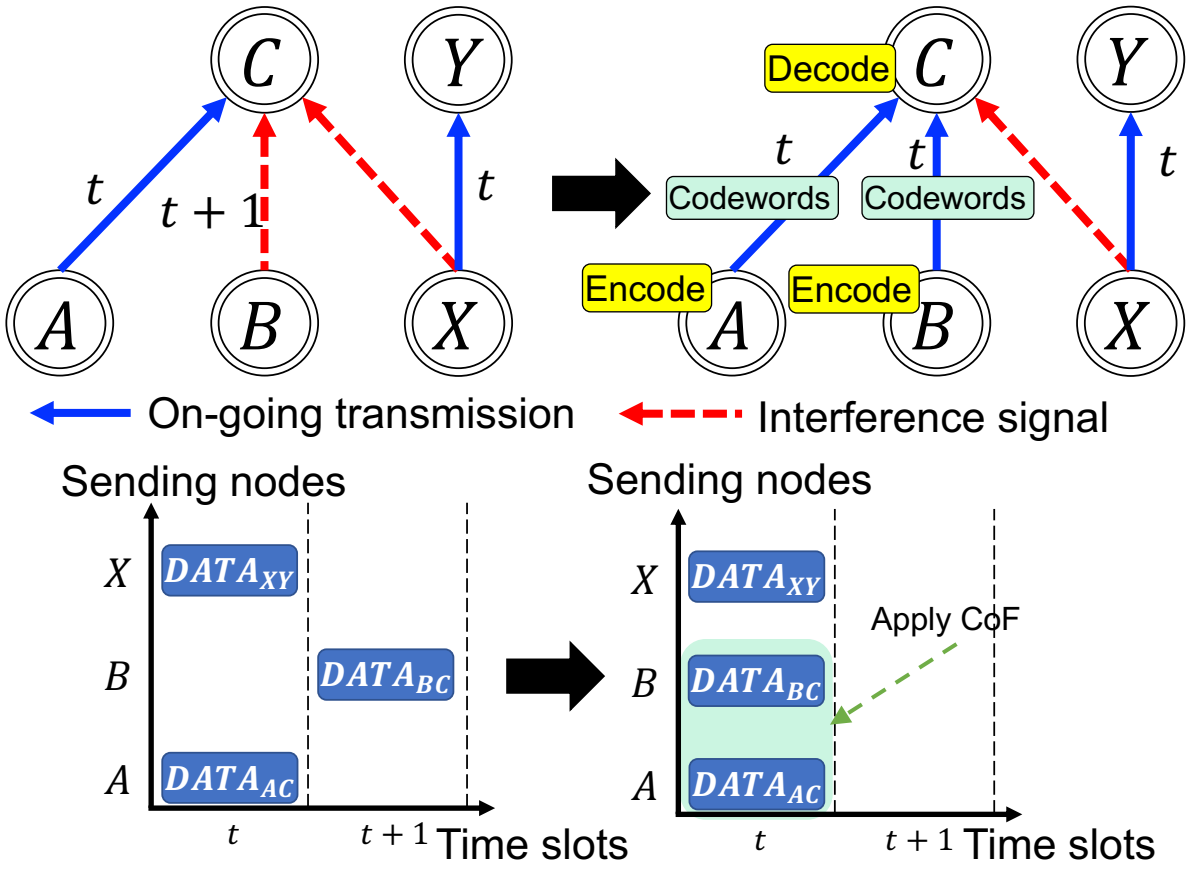


Figure 3.9: The CoF strategy in the data transmission.

As shown in Figure 3.9, in a single time slot, nodes A, B try to send messages to node C, and node X tries to send messages to node Y. Before applying the CoF strategy, the latency can be doubled if these two messages are sent at the same time, which also means it needs more time slots to finish all data transmissions. In this situation, the network capacity will be very low. After applying the CoF strategy, the messages sent at the transmitting nodes A, B at the same time can be encoded separately and decoded at the receiving node C. With the help of NLC, these two encoded messages, including codewords, are decoded together at the receiving node C; therefore, the resultant SINR becomes better. As a result, the data transmission rate of these two messages is increased due to the total number of time slots is reduced.

If more than two transmitting nodes try to send their messages to the same receiving node in a time slot, the receiving node will randomly select two of the transmitting nodes.

3.4 Numerical Simulations

This section evaluates the performance of the proposed eCAP scheme through numerical simulation. Performance evaluation compares the existing path selection method [44] algorithm in the same scenarios and network environment.

3.4.1 Simulation Scenarios and Settings

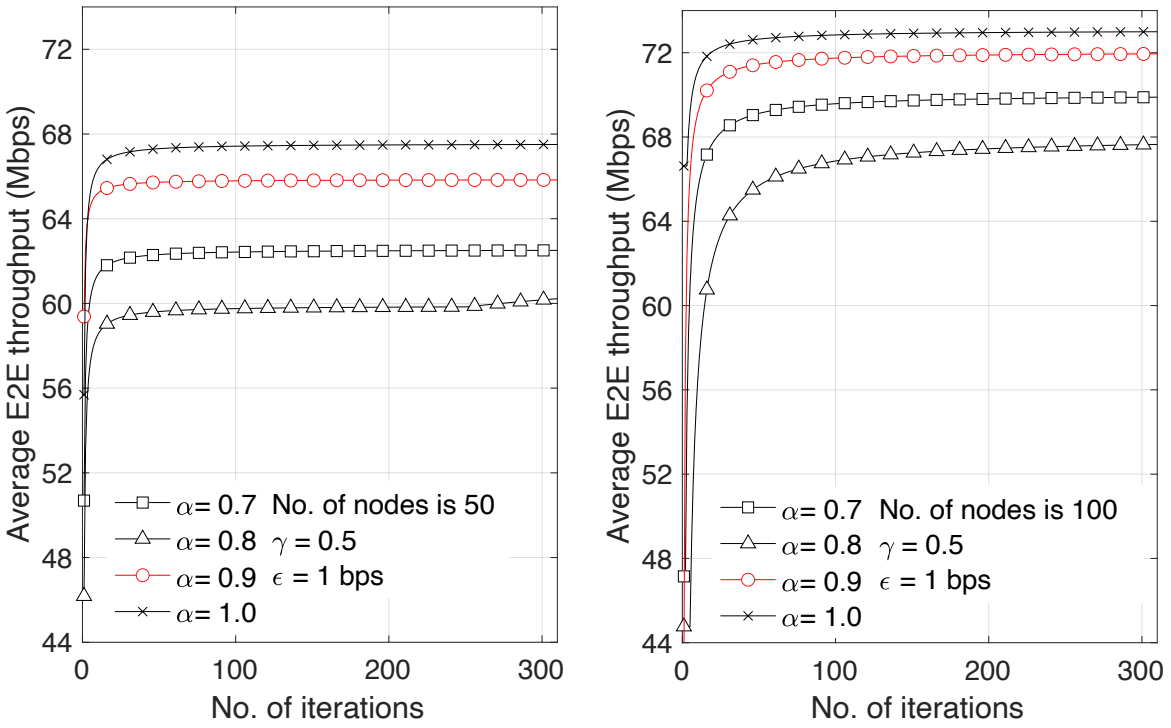
This numerical evaluation is to study the performance of network capacity and computation time with two scenarios. In particular, the first scenario looks at the performance analysis of FG and NLC, whereas the second scenario compares the performance of the proposed LPS algorithms. In this dissertation, we assume that each intermediate node has only one packet to send to the destination root node, and nodes can send one packet and receive at the same time. Besides that, nodes can also receive two packets at the same time. The simulation program is written in MATLAB R2022a, and the desktop environment is an Apple Mac mini (2018) with Intel Core i7 3.2 GHz, 64GB DDR4 RAM. The parameters and values are listed in Table 3.3. The constant attention and shadowing attenuation of the log-distance pathloss model is set as 3.5 and 4.0 dB, respectively, to fit the real-world environment. The values of other parameters are set according to the

Table 3.3: Simulation parameters and settings of eCAP.

Parameter	**Value**
Network coverage size	500 m × 500 m
Transmission range	170 m
Number of nodes	50∼100 nodes
Number of simulations	10,000 times
Transmit power	19 dBm
Attenuation constant	3.5
Wall attenuation	0 dB
Shadowing attenuation	4 dB
Decorrelation distance	10 m
PHY header length	39.2 μs
MAC header size	320 bits
Channel bandwidth	20 MHz
Test payload size	8192 bits
Noise level	-174 dBm/Hz
RSSI	-65 dBm
FER	10^{-4}
DIFS	34 μs
SIFS	16 μs
CW_{min}	15
ACK size	112 bits
Slot time	9 μs
Basic rate	6 Mbps
Learning rate	0.9
Discount factor	0.5
Threshold	1 bps
Data packet size	1000 bytes

IEEE802.11ax standard specification [45].

For Q-learning settings, the default value for learning rate is 1.0. However, the Q-



(a) 50 nodes

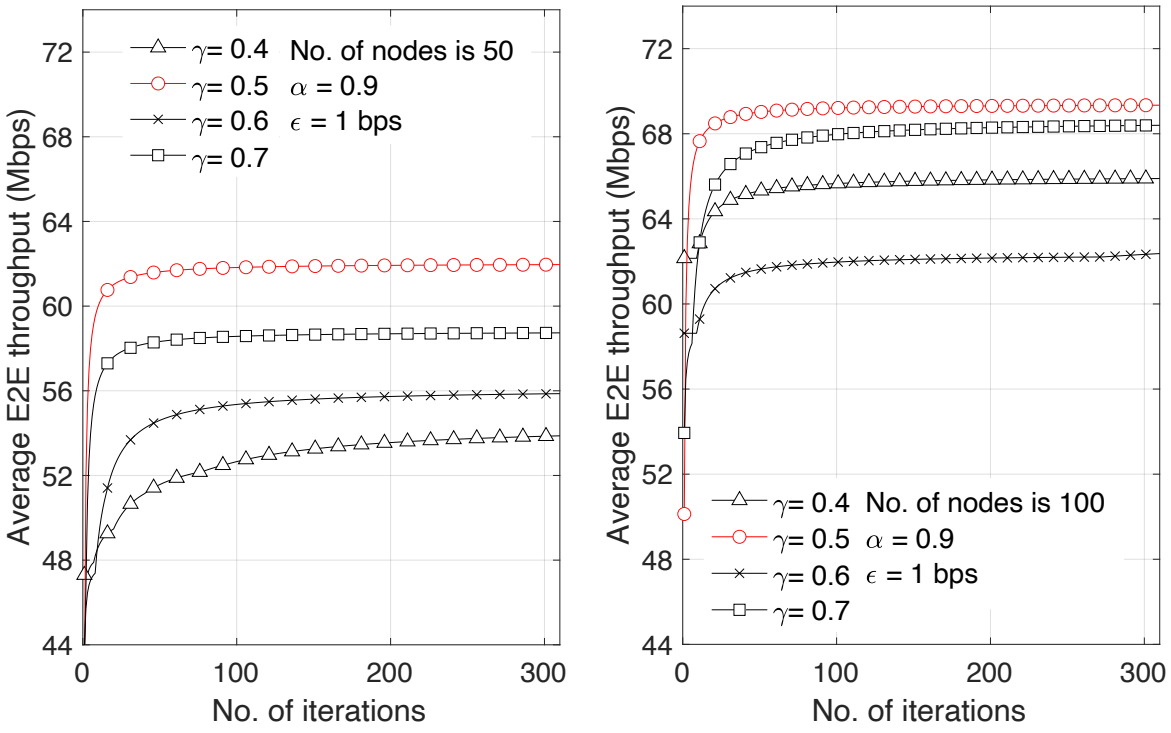
(b) 100 nodes

Figure 3.10: Performance of learning rate in the Q-learning.

function has been modified, so it is necessary to revisit the appropriate value for the learning rate. Figure 3.10 shows the comparison of different learning rate when the number of nodes is 50 and 100 nodes with same discount factor and threshold. The result shows that the best learning rate for our Q-function is 1.0. $\alpha = 1.0$ means that only new information is considered when updating the Q-value, and the influence of current information is completely ignored. For a more realistic environment, it is necessary to give some weight to the current information. Thus, we consider the learning rate α is 0.9.

Next, the discount factor in Q-learning decides the importance of the Q-value for the best action. Simply to say, the discount factor is larger means that the agent's next action will be considered in a long-term manner, otherwise the next action will be considered in a short-term manner. The result of discount factor in the Q-learning is shown in Figure 3.11. When the discount factor $\gamma = 0.5$, the performance of average E2E throughput is better, which also means setting the importance of $Q_{max}(m+1)$ as 0.5 is the best value for our training time of Q-learning.

Finally, the threshold of Q-learning determines when the training should be finished. In



(a) 50 nodes

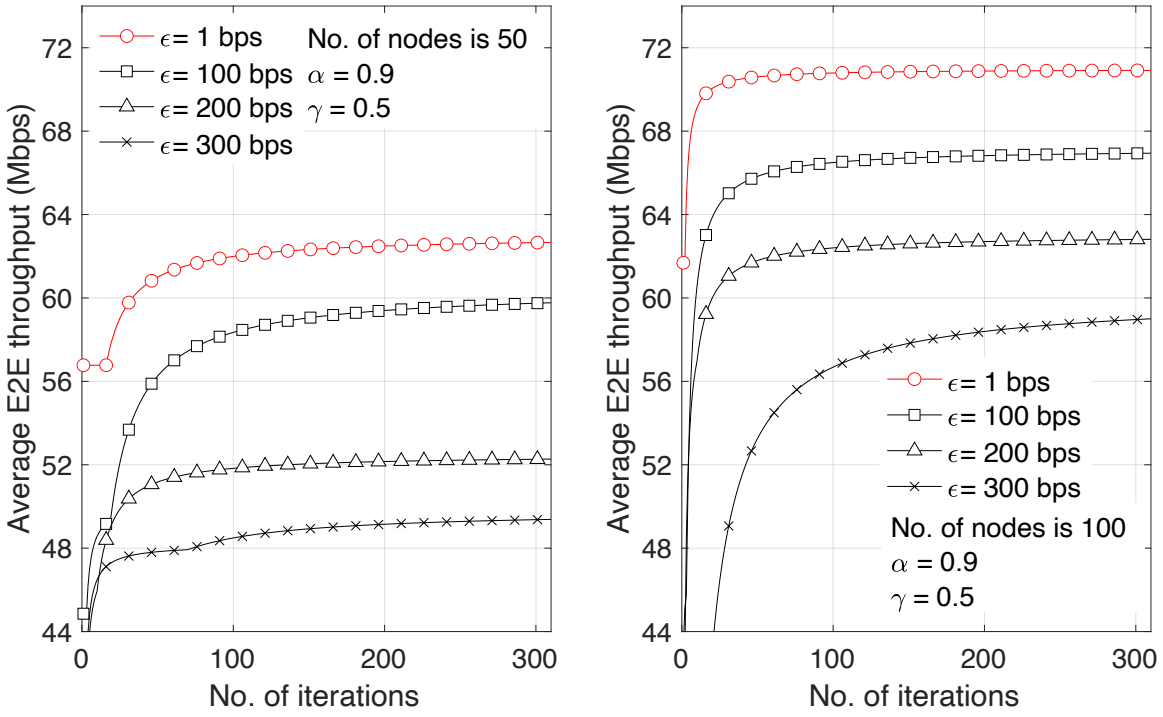
(b) 100 nodes

Figure 3.11: Performance of discount factor in the Q-learning.

common, the smaller the threshold in the Q-learning, it shows the better the performance results. However, the number of iterations can be also increased at the same time.

Figure 3.12 shows the performance result of different thresholds when the number of nodes is 50 and 100. As we can observe from these results, the lower the threshold, the higher the average E2E throughput. When the threshold is set to 1 bps, the training works fine and almost reaches to more than 300 iterations. After 300 iterations, the throughput remains unchanged. Thus, we set the threshold ϵ equivalent to 1 bps in our simulation.

As for the convergence and stability properties, we conclude that the parameters, learning rate, discount factor, and threshold in both algorithms, to balance exploration and exploitation, allow agent's decision for the next action, and prevent over-training or non-stopping scenarios, respectively. However, we need to adjust further according to the dynamic environment. Besides that, we use the best next hop's reward as a reference factor to calculate the Q-value, ensuring that the algorithm consistently finds the best path. We recognize the importance of the algorithm's ability to adapt its performance based on the observed network environment scenario. We plan to modify the NLPS and



(a) 50 nodes

(b) 100 nodes

Figure 3.12: Performance of threshold in the Q-learning.

INLPS algorithms to include other possible feedback parameters in order to not only enhance their robustness, scalability, and efficiency in dynamic environments, but also to be highly feasible in real-world applications.

3.4.2 Simulation Results and Discussion

The numerical simulation mainly investigates the performance of network capacity and computation time by comparing different scenarios and consists of two parts. In the first part, we investigate the performance of network capacity and computation time with/without FG and CoF. Besides that, we also examine how the FG approach is used to find the root node to improve the performance of the proposed eCAP scheme. In the second part, the NLPS algorithm and the INLPS algorithm is applied to find the best network topology in WMN. The scenarios 'NLPS' and 'INLPS' are using the NLPS algorithm and the INLPS algorithm to select the best path, respectively, and NLC with CoF strategy is also used in data transmission to increase network capacity further.

Performance Analysis of FG and CoF

In this section, we discuss the results of FG and CoF. There are three kinds of path selection methods. The 'TSSR' abbreviation means the path selection method is obtained by a tree topology, which is computed by the two-stage submodular relaxation (TSSR) algorithm [25]. For the legends of 'FG without CoF' and 'FG', they mean the path selection method is using FG without CoF and FG with CoF, respectively.

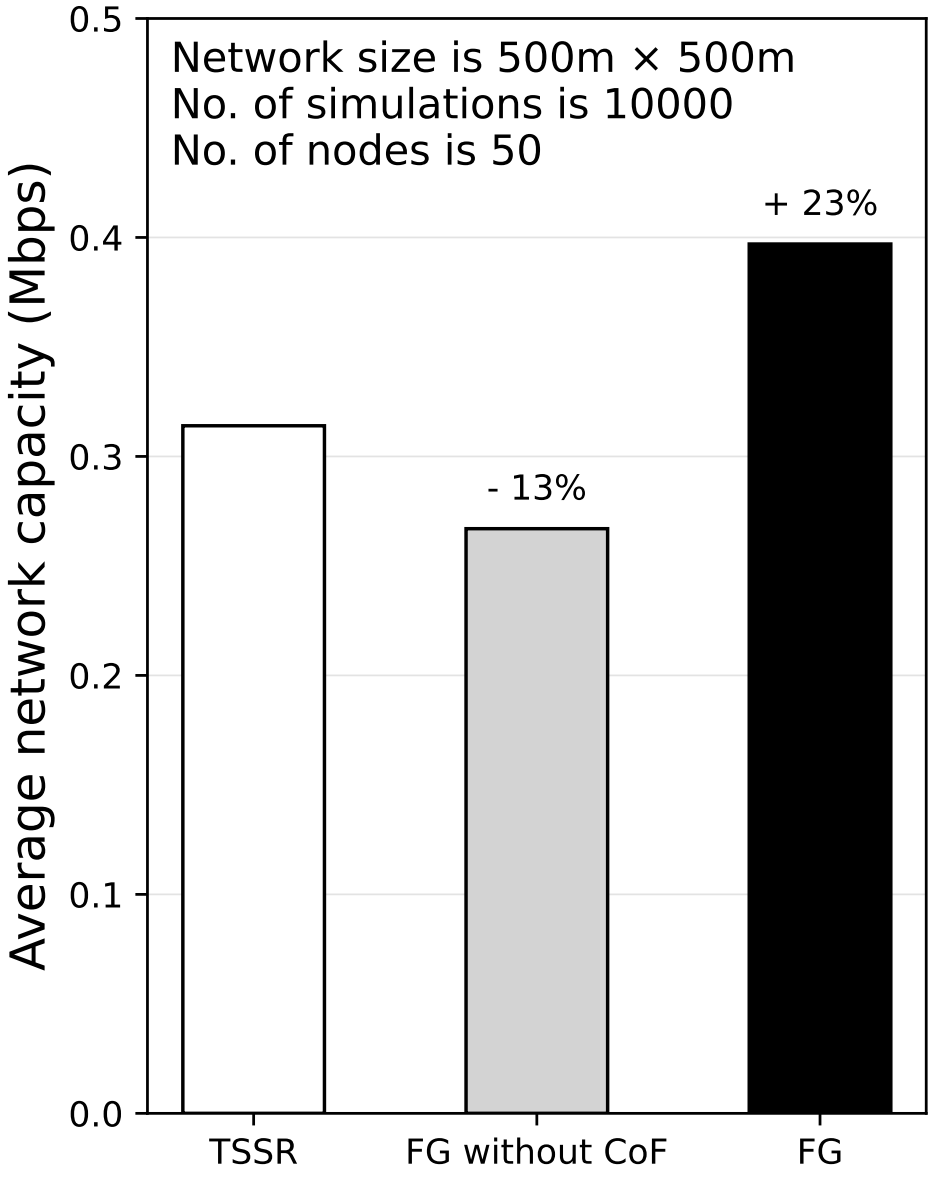


Figure 3.13: Network capacity comparison of the FG approach with and without CoF.

In Figure 3.13, the average network capacity using FG is 0.267 Mbps or about 13% times lower than TSSR (0.314 Mbps) when the number of nodes is 50. If CoF is also

applied, the average network capacity reach to about 23% times higher, i.e., 0.397 Mbps.

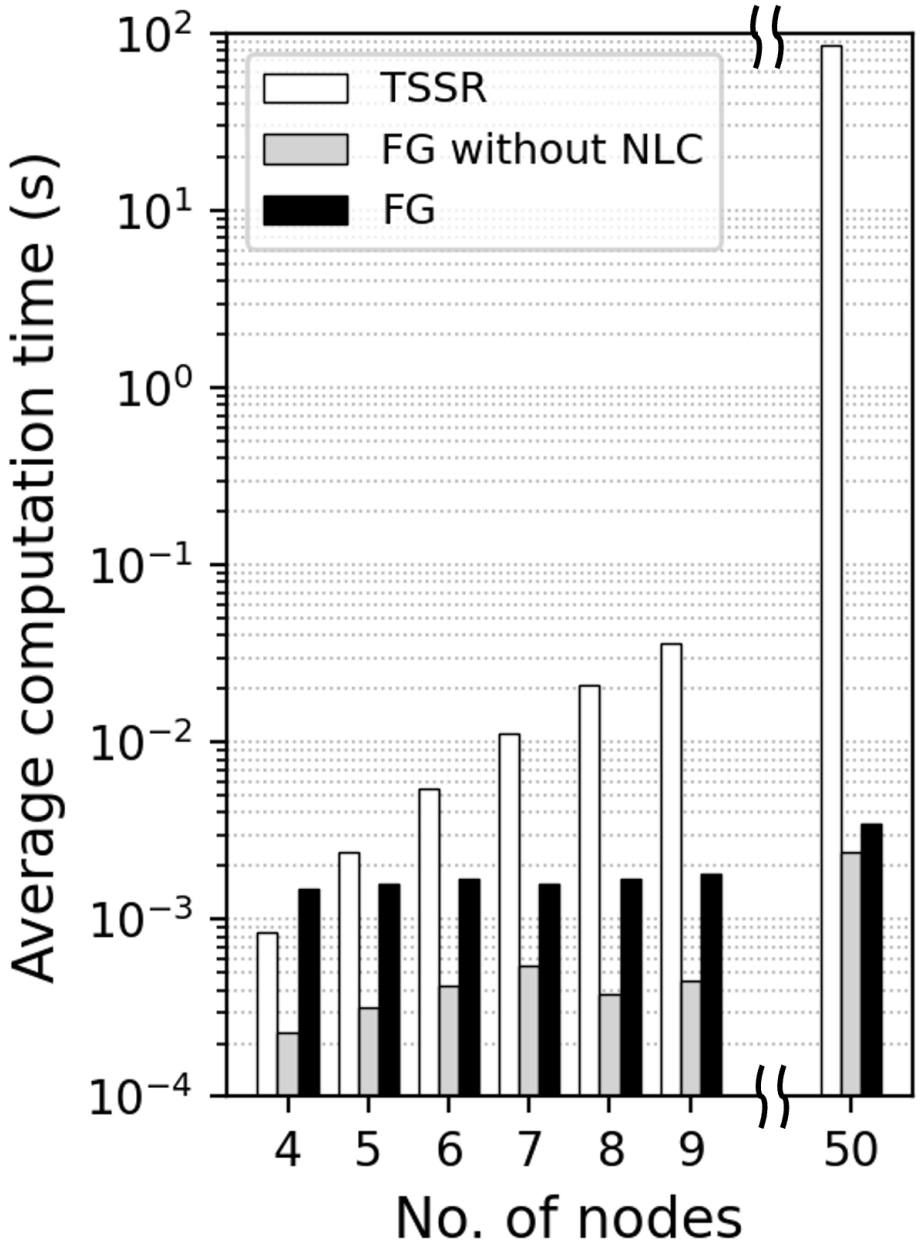


Figure 3.14: Computation time comparison of the FG approach with and without CoF.

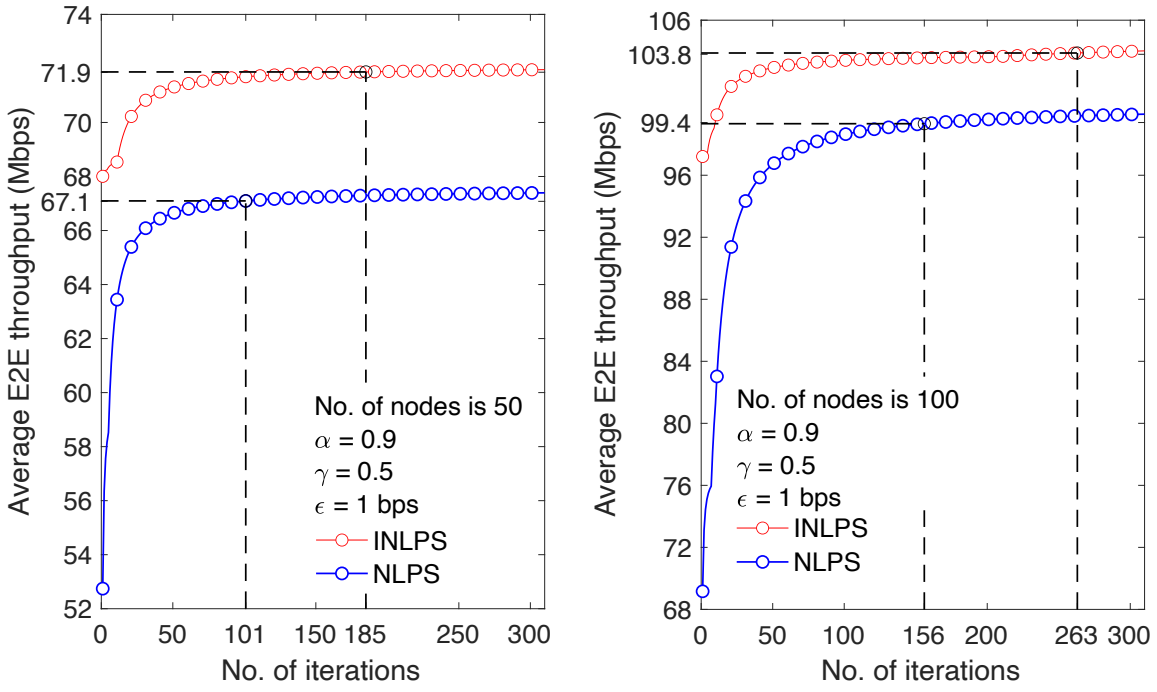
As for computation time in Figure 3.14, when the number of nodes is smaller than four, applying FG and CoF will cost more computation time. Otherwise, the computation time becomes better compared to the 'TSSR'. More specifically, when the number of nodes is 50, it costs about 89.7 s, while FG with CoF only costs 3.44×10^{-3} s.

Through these simulation results, we can conclude that the network topology selected by FG can increase the average network capacity and greatly reduce the computation time.

Besides that, CoF can further increase network capacity. Although the computation time of CoF will increase slightly, it is considered in the acceptable range of latency in the real network environment. We can observe that the network capacity is still very low when both the FG and CoF approaches are used. It is because all the network topologies are in the form of tree-based topology, which means that all the source nodes choose to send packets directly or multihop manner to the one root node. As a result, the network capacity becomes low because all the nodes need a large number of time slots to complete their data transmission to the root node. Thus, it is necessary to use an LPS algorithm to plan a forwarding path to carry all the packets to the destination root node.

Performance Analysis of NLPS and INLPS Algorithms

We first investigate the performance of the number of iterations, network capacity, and computation time. Figure 3.15 shows the results of average E2E throughput versus number of iterations between NLPS and INLPS algorithms when the number of nodes is 50 and 100.



(a) 50 nodes

(b) 100 nodes

Figure 3.15: Performance results of average E2E throughput versus number of iterations between NLPS and INLPS algorithms.

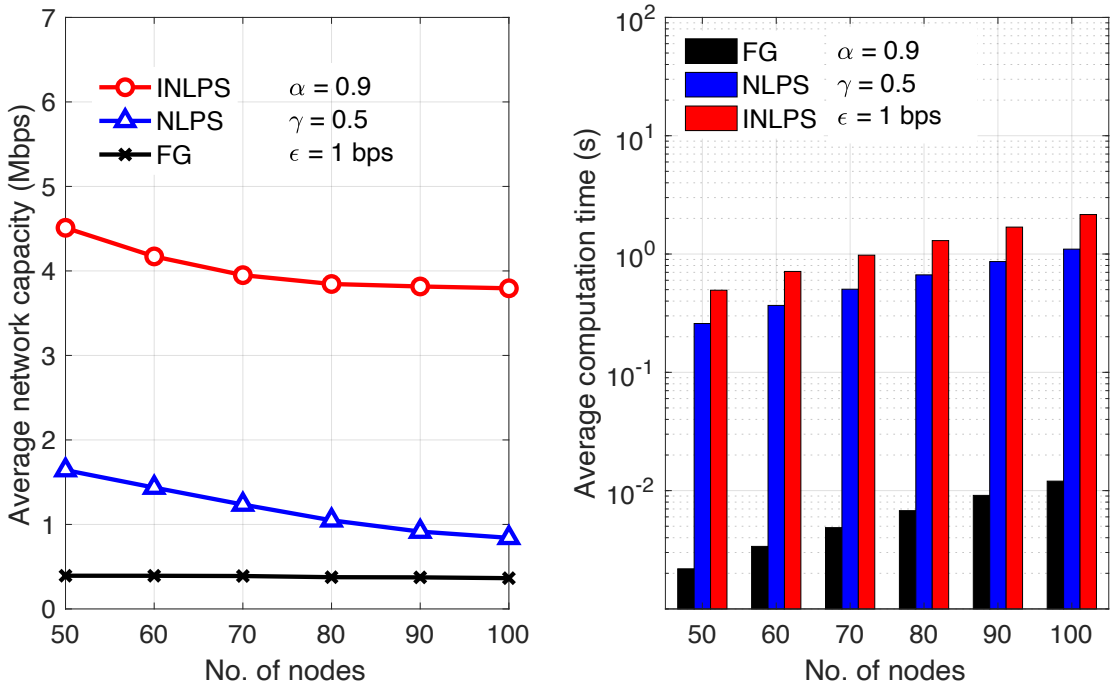
These results reveal that the INLPS algorithm can reach higher average E2E throughput than the NLPS algorithm. For the sake of comparison, in this dissertation, we use the number of settling iterations, which is defined as the number of iterations for the steady state to reach, and stay within 2% of its final value. In Figure 3.15 (a), when number of nodes is 50, the number of settling iterations of NLPS and INLPS is 101 iterations and 185 iterations, respectively. This means that NLPS has fewer iterations to complete the training than INLPS. Similarly, we can observe the same result when the number of nodes is 100, i.e., 156 iterations and 263 iterations, respectively, in Figure 3.15 (b). Therefore, we can conclude that the average E2E throughput of the INLPS algorithm is higher than that of the NLPS algorithm, even though it takes more iterations. The reason is that the INLPS algorithm uses Q-values that is obtained by the NLPS algorithm to initialize its Q-tables. By incorporating the total network interference for the path selection, the INLPS algorithm can achieve higher average E2E throughput.

Table 3.4: Average E2E energy consumption of NLPS and INLPS.

No. of nodes	Algorithms	
	NLPS	INLPS
50	9.45 nJ	8.83 nJ
100	6.39 nJ	6.12 nJ

The comparison of E2E energy consumption between the NLPS and INLPS algorithms, as shown in Table 3.4 reveals an essential aspect of network energy efficiency. Despite the INLPS algorithm's time complexity, it demonstrates a marginal improvement in energy consumption over the NLPS algorithm for both 50 and 100-node scenarios. The results of Figure 3.15 and Table 3.4 show the balance the INLPS algorithm strikes between enhancing network performance and managing power resource efficiency. The slight decrement in average E2E energy consumption by the INLPS algorithm, compared to NLPS, depicts that the enhancements for handling network capacity optimization do not significantly increase energy efficiency. This observation highlights the essential of considering energy efficiency in the design and optimization of network protocols, particularly in cases where reducing energy consumption is as important as improving network performance.

Second, we look at the results of the network capacity, which are shown as Figure 3.16



(a) Network capacity comparison

(b) Computation time comparison

Figure 3.16: Performance evaluation of FG, NLPS, and INLPS.

(a). When the number of nodes varies from 50 to 100, our simulation results show that the average network capacity decreases. Quantitatively, when the number of nodes is 50, the average network capacity of NLPS algorithm and the FG is 1.64 Mbps and 0.39 Mbps, respectively. Meanwhile, the average network capacity of the INLPS algorithm is 4.51 Mbps or about 2.75 times and 11.56 times higher than the NLPS algorithm and the FG, respectively. For the number of nodes is 100, the average network capacity becomes 3.79 Mbps, 0.84 Mbps, and 0.36 Mbps, respectively. This is because the INLPS algorithm first trains its Q-values based on the results of the NLPS algorithm by incorporating the total network interference. Then, the INLPS algorithm uses SINR is the reward for avoiding using the path with low SINR when it is selecting the paths. Therefore, this leads to a higher average network capacity can be achieved by the INLPS algorithm.

Next, as we can see the performance comparison of the computation time in Figure 3.16 (b), The INLPS algorithm is about 2 times higher than the NLPS algorithm because the INLPS algorithm needs to perform the same computation time of the NLPS algorithm first to get results as its Q-value input. The computation time of both the NLPS and INLPS algorithms includes the computation time of FG, in which we can observe that

FG takes very little time to find the best root node. Without FG, both NLPS and INLPS algorithms need to be ran on all the nodes. Consequently, it is impossible to find the best network topology that can maximize the average network capacity for the LPS algorithms. With the introduction of FG, the NLPS algorithm and the INLPS algorithm can find the best network topology, and at the same time they can accomplish the average network capacity closes to the maximum network capacity together with a great reduction of computation time.

Table 3.5: Average computation time versus average latency of FG, NLPS and INLPS.

No. of nodes	Average computation time			Average latency		
	FG	NLPS	INLPS	FG	NLPS	INLPS
50	2.18 ms	258.53 ms	494.48 ms	20.36 ms	4.86 ms	1.77 ms
60	3.37 ms	367.79 ms	712.51 ms	20.39 ms	5.57 ms	1.92 ms
70	4.86 ms	504.31 ms	978.11 ms	20.53 ms	6.48 ms	2.03 ms
80	6.77 ms	666.51 ms	1298.64 ms	21.30 ms	7.63 ms	2.08 ms
90	9.11 ms	863.68 ms	1688.88 ms	21.42 ms	8.74 ms	2.10 ms
100	12.03 ms	1099.96 ms	2156.32 ms	21.98 ms	9.51 ms	2.11 ms

Last, Table 3.5 delineates the average computation time and average latency for the FG, NLPS, and INLPS algorithms when varying node counts, presenting a tradeoff view between computational overhead and network performance. Notably, the INLPS algorithm exhibits high average computation times but results in low average latency, particularly as the number of nodes increases. The overall results of computation time and latency emphasize the importance of strategic decision-making when it is applied to real-world applications, like a network formed by a group of stationary communicating devices, in order to achieve such resource utilization efficiency. In other words, network environment where reducing latency is paramount, the INLPS algorithm emerges as the preferable choice despite its computational time. This analysis accentuates the vital role of algorithm selection in optimizing WMN configurations, especially in applications where latency extreme impact on overall network performance and user experience.

3.5 Summary

This chapter presented the eCAP scheme, which serves as the capacity-oriented module within the proposed CORE framework. The chapter first clarified the root causes of capacity degradation in WMNs, emphasizing the impact of dense interference, hidden terminals, bottleneck relays, and spatially correlated contention on the formation of high-throughput multihop routes. Based on these observations, a system model was formulated that incorporates channel characteristics, interference coupling, link rate, and airtime-based link quality metrics.

To address the capacity limitation problem, eCAP introduces a cooperative and AI-assisted multihop transmission mechanism. A factor-graph-based shortest-path spanning tree is first constructed to determine a convergent topology and parent–child relationships according to link quality, while simultaneously reducing the search space for routing. On top of this structure, a two-stage LPS mechanism is applied: NLPS is used to rapidly explore candidate paths with high transmission reliability, followed by INLPS that explicitly accounts for mutual interference during concurrent transmissions. In addition, NLC combined with a CoF strategy enables paired child nodes to forward packets simultaneously toward a common parent, thereby increasing spatial reuse and reducing the number of required time slots.

Numerical simulations validated the effectiveness of the proposed eCAP scheme under various node densities and interference conditions. The results demonstrated substantial performance gains over conventional tree-based routing, including significant improvements in network capacity, higher average E2E throughput, and considerable reductions in computation time due to the introduction of the FG-based root selection. These gains stem from efficient topology construction, interference-aware learning, and cooperative coded forwarding.

Overall, the eCAP scheme establishes a practical solution to the capacity bottleneck in multihop wireless networks. It provides a solid foundation for the subsequent latency-oriented and power-oriented optimization modules developed in the CORE framework, which are presented in Chapters 4 and 5.

Chapter 4

Extreme Low Latency Transmission Scheme

Following the capacity-oriented optimization developed in Chapter 3, this chapter advances the research to the second stage of the CORE framework: extreme low-latency transmission in multi-server environments. While the eCAP scheme in Chapter 3 enhances network capacity by constructing high-throughput multihop topologies in WMNs, its performance analysis reveals that, once the problem of path capacity is alleviated, E2E service latency becomes the dominant limiting factor—particularly when a large number of devices offload tasks to a limited set of edge servers. This observation motivates the development of the eLOW scheme, which targets the intrinsic challenge of server selection and load balancing in MSWNs. In this chapter, the primary optimization target shifts from path selections to server association decisions, where each device selects one server from multiple candidates to minimize network latency.

In MSWNs, each device is capable of associating with multiple ESs, and each ES may simultaneously serve a large number of devices. Although multiserver deployment increases computational resources and service coverage, it introduces a new decision-making problem: devices must determine not only how to transmit, but also where to offload. Inefficient server allocation can result in queue build-up, uneven resource utilization, excessive interference, and degraded QoS. Traditional static or heuristic methods cannot sufficiently adapt to the highly dynamic characteristics of modern wireless networks, where mobility, fluctuating traffic arrivals, and interference coupling change rapidly

over time.

To address this challenge, the eLOW scheme formulates latency-aware server selection as a learning-driven optimization problem. It leverages the Broad Learning System (BLS) as the primary decision engine in the deciding module of the CORE framework. Owing to its shallow yet expansive architecture and incremental learning capability, BLS enables rapid adaptation to evolving network conditions without expensive retraining, making it particularly suitable for real-time low-latency management. Based on this learning core, two BLS-based strategies are designed to optimize device–server association from different perspectives. These strategies enable the framework to flexibly balance propagation delay, link quality, and throughput while achieving scalable and low-latency resource allocation under dense deployments.

The remainder of this chapter is organized as follows. Section 4.1 introduces related research on BLS-based server allocation and resource management in MEC and MSWN environments. Section 4.2 formulates the system model and performance metrics for the proposed eLOW scheme. Section 4.3 presents the detailed design of the BLS-based optimization strategies. Section 4.4 evaluates the performance of the scheme through numerical simulations. Finally, Section 4.5 concludes this chapter.

4.1 Research Background

As the evolution towards B5G progresses, the burgeoning demands for higher data rates, lower latency, and enhanced area traffic capacity pose significant challenges to the capacities of existing wireless networks, especially in device-to-device communication scenarios and MSWNs. These expectations underscore the necessity for future wireless networks to effectively manage high-density connections, reduce latency, and augment area traffic capacity, highlighting the imperative for pioneering solutions.

In the context of MSWNs, where IoT devices communicate directly with multiple MEC servers within a single hop, optimizing network capacity and reducing latency become particularly challenging. This environment demands innovative approaches to ensure efficient resource allocation and to enhance overall network performance. Integrating AI solutions with advanced techniques emerges as a key strategy to meet these demands. By leveraging AI's capabilities, wireless networks can dynamically adapt, enabling real-

time decision-making for resource allocation and thus significantly improving network performance in various MSWN scenarios.

Among the wireless technologies explored for these purposes, the Broad Learning System (BLS) offers a promising avenue for optimizing network performance in MSWNs. Known for its fast learning speed, incremental learning ability, and robust feature extraction, BLS is particularly well-suited for handling the dynamic and large-scale nature of MSWNs.

Compared to DL, BLS adopts a shallow network architecture instead of a deep one, which substantially reduces computational complexity and training time. In dynamic MSWN environments, where real-time decision-making is crucial, rapid learning capability enables BLS to adapt more efficiently to network changes. In contrast, deep learning requires intensive computational resources and long training times, making it difficult to respond swiftly to network fluctuations.

Another key advantage of BLS lies in its incremental learning capability, which allows it to update the model without complete retraining. This feature is vital for MSWNs, where devices frequently join or leave the network, and communication loads vary continuously. While RL also provides adaptive optimization through interaction with the environment, its exploration–exploitation process often demands numerous iterations and training episodes, leading to inefficiency in fast-changing environments. BLS, on the other hand, achieves effective model adaptation with far fewer updates.

Moreover, BLS demonstrates strong feature extraction and generalization performance by utilizing randomly generated enhancement nodes and ridge regression for output weight optimization. This mechanism ensures robustness and adaptability when dealing with dynamic, high-dimensional wireless data. Unlike RL, which relies heavily on trial-and-error exploration and complex reward design, BLS can efficiently model relationships between network states and optimal actions without extensive interaction or simulation.

In summary, BLS excels in learning efficiency, incremental adaptability, and feature robustness, making it highly suitable for optimizing multi-server wireless networks.

Therefore, this chapter explores the application of BLS to MSWN optimization and introduces tow BLS-based strategies: BLS with Nearest MEC Server (BLS/NS), and BLS with Maximum SINR (BLS/MS). Each strategy is designed to optimize a specific

aspect of network performance, focusing on enhancing connectivity, reducing latency, and improving throughput in dense network deployments. Factors such as energy consumption and hardware configurations are not considered in this study.

This chapter commences with a comprehensive review of existing literature, identifying gaps in current methodologies and setting the stage for our proposed BLS-based scheme. We then present our system model, including the network, channel, and interference models, establishing a robust foundation for our investigation into MSWN optimizations. Subsequent sections detail the development and implementation of the BLS-based strategies, demonstrating their potential to overcome the unique challenges of MSWNs. Through numerical simulations, our analysis evaluates the effectiveness of the proposed strategies, focusing on key performance metrics such as network capacity and latency. This dissertation contributes to the broader field of wireless communications by proposing a novel BLS-based optimization framework for MSWNs, showcasing its potential to enhance network efficiency and reliability in the era of Beyond 5G.

4.1.1 Related Works

In recent years, task offloading and server decision-making in multi-server MEC have received significant attention, aiming to tackle challenges such as latency minimization, optimal resource utilization, and energy efficiency, especially in contexts demanding high mobility. Guo et al. [46] introduced a dynamic computation offloading mechanism for multi-server MEC systems, addressing the trade-offs between energy consumption and task execution delay. Their approach, based on online learning, optimizes transmit power and server selection for mobile devices, aiming to minimize time-average expected task execution delay under energy consumption constraints. Besides that, Gan et al. [47] also suggested a server offloading scheme that can address the concerns of users with limited computational capabilities to minimize the overall latency in a multi-server MEC network. This scheme will be automatic in task scheduling, wireless offloading modes, and server pair selection, which will be based on the power information and the user's power limitation. Lim et al. [48] took this pioneering step by implementing deep reinforcement learning strategies for computational offloading as well as server decision-making based on a soft actor-critic method. They have done an exceptional job by minimizing delay,

saving on energy consumption, and quickly finding a solution that is both stable and convergent with outstanding developments compared to the existing methods.

Aiming the maximization of the effectiveness of wireless networks, recent studies provide machine learning techniques as innovative solutions to the problem of network capacity. Ajayi et al. [49] propose a self-renewing machine learning method dedicated to increasing network throughput by means of a data selection algorithm that allows to classify scheduling structures and retraining models. This method notably improves the efficiency and the computational cost by applying transfer learning. It eventually reduces the computational cost of training models significantly. Hu et al. [50] explored another method of capacity prediction for wireless networks through CNNs, this method of predicting network capacity by artificial data generation from practical network models showed high accuracy. Also, Ajayi et al. [51] proclaimed the capacity optimization for B5G/6G Integrated Access and Backhaul (IAB) networks with a two-stage machine learning framework which brought the throughput improvement of a significant amount and also reduced the computational time. This work reveals an untapped area of machine learning where resource scheduling could be done more efficiently and network performance could be improved.

On the aspect of network latency, innovative approaches have also been explored. Zhohov et al. [52] have applied machine-learning techniques in the handover process in wireless networks, so greatly reducing latency and ensuring the stability of the friend zone service quality when transitions are made between frequencies. The efficiency of this method is underpinned by the actual network measurement and network scenario as both of them essentially make it applicable to real life. As well, Lee and Chung [53] also designed a scheduling system that aims at reducing the FAN latency and the hybrid scheme successfully cuts the maximum unicast and broadcast transmissions. The aforementioned studies in general highlight that machine intelligence is thoroughly involved in wireless network optimization by revealing the costs that necessitate more effective and trustworthy performance in wireless communication systems in the future.

BLS was first proposed by Chen [54] as a flat network designed for efficient, incremental learning. BLS is proposed as a supervised learning and an efficient alternative to deep learning, allowing for rapid adaptation without extensive retraining. Further expand-

ing on the versatility of BLS, Xu et al. [55] demonstrated its application in time series prediction by incorporating recurrent connections within the BLS framework. This adaptation, termed recurrent BLS (RBLS), signifies an advancement in handling sequential data, thus broadening BLS's applicability to domains necessitating accurate prediction over time, such as financial forecasting and weather prediction. Gong et al. [56] provided a comprehensive review of BLS, elaborating on its algorithms, theoretical foundations, and a wide range of applications. Their analysis underscores the system's universal approximation capabilities, showcasing BLS's effectiveness in tasks ranging from classification and regression to semi-supervised and unsupervised learning. The review not only highlights BLS's flexibility and stability but also its practical utility in fields such as computer vision and biomedical engineering, emphasizing the broad applicability and promising future of BLS in advancing AI and machine learning technologies. Furthermore, Chi et al. [57] also proved that BLS can achieve higher accuracy with less training time than other machine learning algorithms and can adapt to changing network environments in the industrial Internet of Things.

4.1.2 Motivation

Although machine learning has been widely applied to task offloading, server selection, and resource optimization in multi-server MEC systems, existing approaches still struggle to ensure stable latency performance under dynamic conditions. Algorithms such as online learning, transfer learning, and deep reinforcement learning have demonstrated improvements in network performance, but their high computational complexity and slow convergence prevent them from reacting quickly to rapid fluctuations in traffic loads, server congestion, and interference patterns.

In MSWNs, the dominant performance problem becomes E2E service latency rather than computational capability. When many devices simultaneously offload tasks to a small set of edge servers, queue buildup and asynchronous processing can trigger sudden latency spikes, even if communication links remain strong. Preventing such latency amplification requires predictive decision-making rather than reactive server switching.

The BLS offers a promising solution for this challenge. Its fast training speed, incremental learning capability, and strong generalization performance enable real-time adap-

tation to rapidly evolving network states without high computational burden. Although BLS has shown success in domains such as time-series prediction and industrial IoT analytics, its potential for latency-aware optimization in MSWNs remains largely unexplored.

Motivated by this research gap, this dissertation investigates BLS-based server allocation for latency reduction in MSWNs. By leveraging BLS's rapid learning and incremental adaptability, the proposed eLOW scheme is designed to predict latency-related metrics and proactively prevent problem formation before delay buildup occurs. This chapter therefore establishes a foundation for predictive, AI-driven latency control in the emerging B5G and 6G network architectures.

4.2 System Model

This section formulates the system model used for analyzing and evaluating the proposed eLOW scheme. It defines the network topology, channel model, and interference assumptions that characterize the MSWN. Furthermore, key performance metrics such as network latency and network capacity are derived to provide the analytical foundation for the subsequent optimization and simulation processes.

4.2.1 Network Model

The network model of MSWN is shown in Fig 4.1. The MSWN employs MEC servers, which can process data and service requests at the network's edge to enhance network performance metrics such as latency, processing speed, and overall efficiency. MEC servers are capable of making decisions more efficiently by reducing network resource requirements and mitigating traffic congestion during packet transmission [58]. The set of MEC servers is represented by $j = \{1, 2, ..., J\} \in \mathcal{J}$, whereas the set of devices is denoted by $i = \{1, 2, ..., I\} \in \mathcal{I}$. It is assumed that MEC servers are wirelessly interconnected, allowing for the sharing of gathered information from all the devices. Each device periodically sends one packet to an MEC server. All the devices in the network operate on the same frequency, have identical bandwidth, and share the same antenna gain. The network coverage area is segmented from the center into several sector-shaped areas, corresponding to the number of MEC servers, with each area hosting one MEC server. Devices are

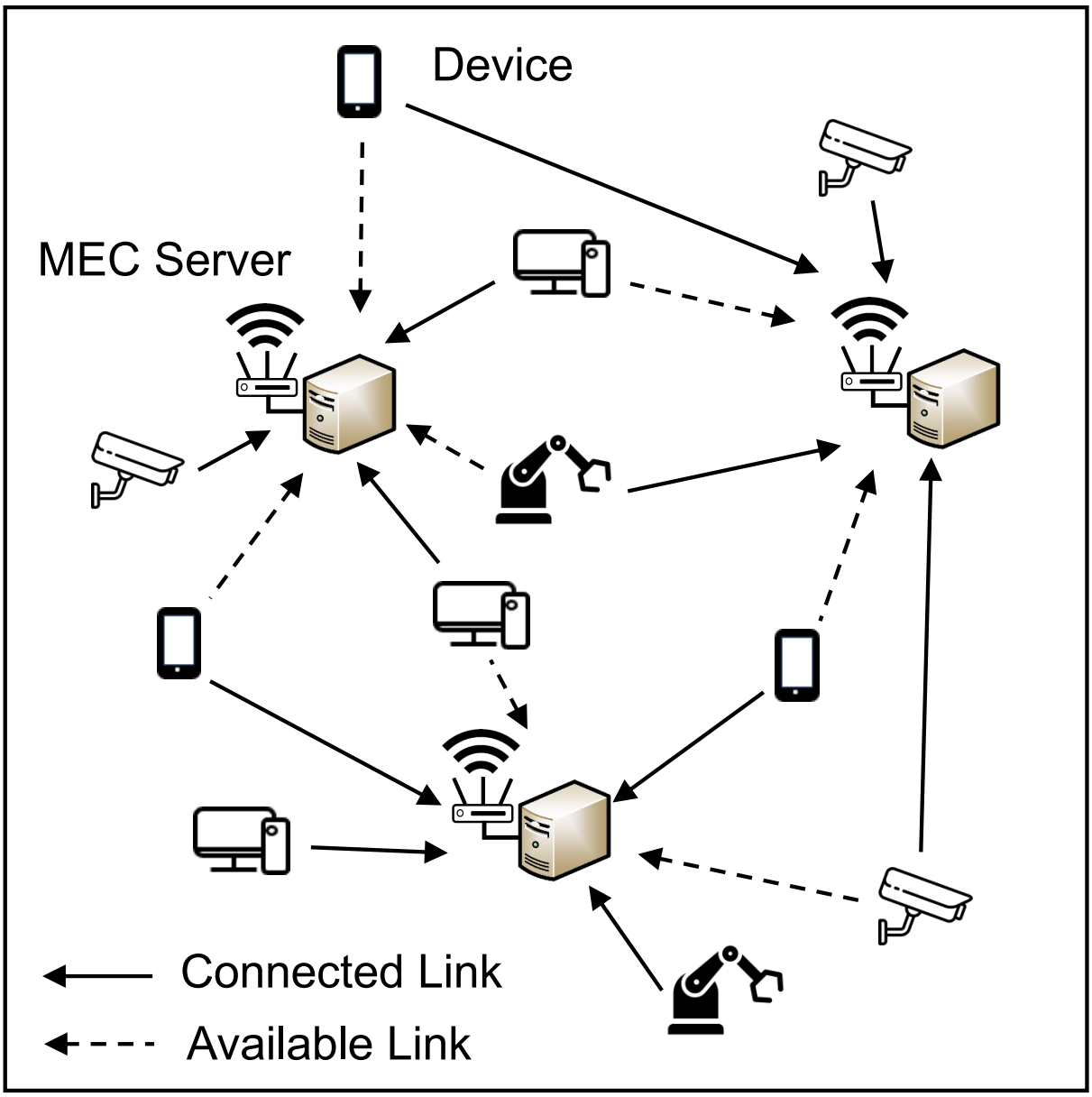


Figure 4.1: Illustration of multi-MEC server wireless networks.

distributed randomly across the entire network area. In this MSWN model, there exist many possible alternative links between each device and several servers. These alternative links, when chosen to transmit the data packet, become a connected link, indicating the establishment of a direct communication path for data transmission from a device to its corresponding server.

Suppose the set of interfering devices is represented by $k = \{1, 2, ..., K\} \in \mathcal{K}$. Thus, the signal-to-interference-plus-noise ratio ($SINR$) at a server j from a device i is given by

$$SINR_{ij} = \frac{G_{ij} \cdot P_i}{\eta_j \cdot B + \sum_{k \in \mathcal{K}} G_{kj} \cdot P_k} \tag{4.1}$$

where P_i is the transmit power of device i, η_j is noise level of server j, and B is the channel bandwidth. G_{ij} is the power ratio between device i and server j calculated by the log-distance pathloss model [17].

4.2.2 Calculation of Network Latency and Network Capacity

We can obtain the transmission rate, R_{ij} from a device i to MEC server j as $R_{ij} = B \cdot \log_2(1 + SINR_{ij})$. In our study, we set the actual size of each packet will be randomly determined in the range of 100 bytes below the maximum packet size. For example, we set the maximum packet size is 1500 bytes, the actual packet size will be around 1400 to 1500 bytes.

In the MSWN model, the time division multiple access (TDMA) protocol is assumed, in which each device that sends to the corresponding MEC server must use the allocated time slot for its data packet. In this way, the latency is the total number of transmission time slots for all the devices for a jth MEC server. Hence, we define that the network latency, Θ is the maximum latency of all the MEC servers in a communication network, can be written as

$$\Theta = \max_{j \in \mathcal{J}} \left\{ \sum_{i=1}^{I_j} \frac{L_i}{R_{ij}} \right\} \tag{4.2}$$

where L_i is the packet size for device i, I_j is the number of devices sending data packets to the MEC server j.

For the sake of simplicity in the evaluation part, we assume that each device has only one packet to be sent to the MEC server. Therefore, the network capacity, $\mathcal{C}$, is the sum of the sizes of all packets sent from devices divided by the network latency, which can be written as

$$\mathcal{C} = \frac{\sum_{i=1}^{I} L_i}{\Theta} \tag{4.3}$$

4.2.3 Network Optimization Targets

The primary optimization variable in the eLOW scheme is the server association of each device in MSWN. Each device may have several candidate MEC servers within its coverage, and selecting the most suitable server directly affects the network latency.

In contrast to Chapter 3, where the optimization problem focused on selecting next-hop relays to construct high-capacity multihop paths, the optimization problem in this chapter is defined as a server-selection problem. Each device must determine which MEC server it should offload its data to in order to minimize the overall service latency.

The eLOW scheme models this decision-making problem as a supervised learning task using the BLS. BLS takes device-level network observations as input features and outputs the predicted server index that yields the lowest network latency. Thus, the optimization performed in this chapter corresponds to mapping network conditions to an appropriate device–server association.

4.3 Extreme Low Latency Transmission Scheme

This section presents an eLOW scheme that leverages the BLS to enable devices to select suitable MEC servers under the assumptions and constraints. The goal is to determine an optimal server allocation strategy that minimizes network latency and enhances overall capacity performance.

4.3.1 Broad Learning System Overview

The Broad Learning System (BLS) is an incremental supervised learning algorithm designed to adapt to dynamic environments without relying on deep network architectures. Unlike traditional deep neural networks, BLS employs a flat network structure composed of feature nodes and enhancement nodes, which enables fast training, efficient updating, and robust generalization. An illustration of the BLS structure is shown in Figure 4.2.

In the context of the proposed eLOW scheme, the BLS serves as the core decision-making module for server association in MSWNs. Specifically, the optimization variable in this chapter is the server index selected by each device. Thus, BLS is used as a supervised learning model that maps device-level network observations to a discrete server-selection

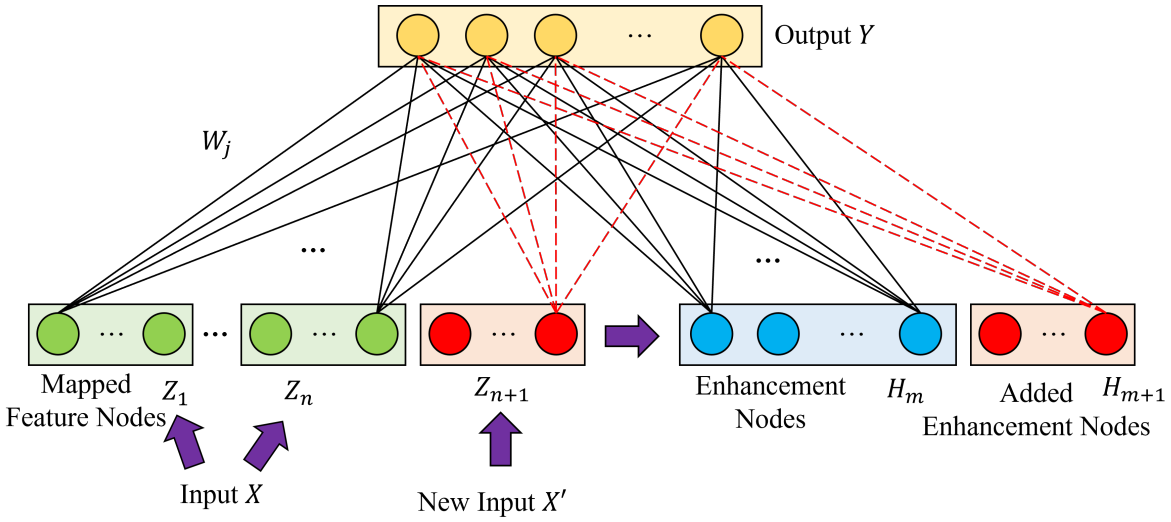


Figure 4.2: Illustration of the BLS.

output. The input features provided to the BLS include device locations, distances to all MEC servers, link-quality indicators, task sizes, and the computational capabilities of each MEC server. These features collectively characterize the latency-related conditions experienced by each device. The output of the BLS is a server label indicating which MEC server the device should associate with to minimize its network latency.

BLS predicts the optimal server for each device based on the learned relationship between network conditions and latency-minimizing server assignments. This formulation enables fast and scalable server allocation in dynamic MSWN environments.

As a supervised learning model, BLS requires a labeled dataset consisting of input features and corresponding output labels. In this study, the input features include the locations of all devices and MEC servers, the task size of each device, and the computational capacity of each MEC server. To generate the training labels, we employ the Gurobi optimizer [59] under diverse network conditions. The optimal server assignment obtained from Gurobi serves as the target label in the training dataset. Once the dataset is prepared, it is used to train and evaluate the BLS model. After training is complete, the weight matrix that links the mapped feature node Z and output Y will be used as the training result of BLS for online server selection.

Data Preprocessing and Feature Extraction

To ensure numerical stability and equal feature contribution, all input data are normalized using min–max normalization:

$$z' = \frac{z - z_{min}}{z_{max} - z_{min}}, \tag{4.4}$$

where z represents a raw feature value, z' the normalized feature, and z_{min} and z_{max} denote the minimum and maximum values of each feature across the dataset.

Next, feature extraction is performed using the Random Forest algorithm, which evaluates feature importance based on the Gini impurity:

$$\Omega(SB_n|X) = \sum_{c=1}^{z} \frac{|SB_{n,c}|}{|SB_n|} \cdot \Omega(SB_{n,c}), \tag{4.5}$$

$$\Omega(SB_{n,c}) = \sum_{c=1}^{z} p(x_c)(1 - p(x_c)), \tag{4.6}$$

where $SB_{n,c}$ denotes a sub-branch, and $p(x_c)$ is the proportion of samples belonging to category c. Features with high importance are retained, effectively reducing dimensionality and mitigating overfitting while improving predictive accuracy.

Feature and Enhancement Node Construction

After preprocessing, the normalized data are projected into feature nodes through randomly generated weights and biases:

$$Z_i = \phi(XW_{ei} + \beta_{ei}), \quad i = 1, \ldots, n, \tag{4.7}$$

where X is the input matrix, W_{ei} and β_{ei} are the random weights and biases, and $\phi(\cdot)$ is an activation function.

These feature nodes are further expanded into enhancement nodes:

$$H_j = \xi(Z_n W_{hj} + \beta_{hj}), \quad j = 1, \ldots, m, \tag{4.8}$$

where $\xi(\cdot)$ denotes another nonlinear activation function. This "broad" structure increases representational capacity without adding network depth, preserving computational efficiency.

Weight Optimization and Incremental Learning

Once the feature and enhancement nodes are constructed, the optimal output weights are determined via ridge regression:

$$W = (\lambda I + A^T A)^{-1} A^T Y, \tag{4.9}$$

where A is the concatenation of feature and enhancement nodes, Y is the labeled dataset, and λ is a regularization parameter that controls overfitting.

A major advantage of BLS is its incremental learning ability. When new devices join the network or mobility patterns change, BLS can update its parameters by adding new nodes without retraining from scratch, greatly reducing computational overhead and ensuring adaptability in dynamic wireless environments.

4.3.2 BLS-Based Optimization Strategies for MSWN

To effectively manage service latency while maintaining sufficient network capacity in multi-server wireless networks, we extend the original BLS optimization (BLSO) framework [57] and tailor it to the MSWN context. In [57], they apply BLSO to minimize the total computing time, which encompasses local computing time as well as remote computing time, with the latter subdivided into uploading time, remote processing time, and waiting time. In this paper, the performance of BLSO thoroughly examines how to optimize network performance metrics, i.e., network capacity and network latency.

However, the BLSO results poor overall network performance due to the parameter of remote computing time did not consider any channel model and interference model. As a result, we refine the BLSO to apply two different network allocation strategies to accommodate the dynamic changes of MMWN environment. Since the network latency depends on the data rate, which relies on the transmit power of a device and total interference level at the MEC server, we need to formalize the optimization problem in terms of the nearest MEC server or maximum SINR. We propose two novel network allocation strategies: BLS with the nearest MEC server (BLS/NS) and BLS with maximum SINR (BLS/MS).

BLS with Nearest MEC Server (BLS/NS)

BLS/NS, assumes that MEC servers possess sufficient computational capacity to handle tasks from multiple devices. Each device selects the geographically nearest MEC server to minimize transmission distance and propagation delay:

$$\mathbf{P}_{BLS/NS} : \min_{\mathcal{D}_i^j} \quad d_{ij} \tag{4.10}$$

where d_{ij} represents the Euclidean distance between device i and server j.

BLS with Maximum SINR (BLS/MS)

BLS/MS, focuses on maximizing the signal-to-interference-plus-noise ratio (SINR) of the communication link:

$$\mathbf{P}_{BLS/MS} : \max_{\mathcal{D}_i^j} \quad SINR_{ij} \tag{4.11}$$

where $SINR_{ij}$ is the SINR between device i and server j. This strategy aims to enhance link quality and mitigate interference effects, thereby improving throughput and stability.

In the BLS/NS strategy, the selected server is the one whose index is predicted by the BLS model to minimize the distance-based cost. In the BLS/MS strategy, the BLS model outputs the server label associated with the maximum SINR among available server candidates.

Common Constraints

All two strategies are subject to the same network constraints:

$$\mathbf{C1} : d_{ij} \leq \frac{d_{max}}{2} \qquad \mathbf{C2} : SINR_{ij} > 0 \tag{4.12}$$

where J and I denote the number of servers and devices, respectively.

The integration of BLS with these optimization objectives forms a eLOW scheme for MSWNs. By leveraging BLS's fast training and incremental adaptability, the scheme can dynamically learn and update optimal server associations under varying network conditions. The proposed strategies jointly enable the network to flexibly balance latency, connectivity, and throughput, ultimately achieving eLOW in Beyond 5G environments.

4.4 Numerical Simulations

This section presents the numerical simulation results to evaluate the performance of the proposed BLS-based eLOW scheme in MSWNs. The experiments are conducted to validate the efficiency, scalability, and adaptability of the two strategies under different network configurations and traffic conditions.

4.4.1 Simulation Scenarios and Settings

The simulations are implemented using Python 3.7 on a Windows 10 system equipped with an 8-core Intel Core i5-1145G7 @ 2.60 GHz CPU and 8.0 GB DDR4 RAM. We deploy multiple MEC servers and devices randomly in a 500 m $\times$ 500 m area. The channel noise level is set to -174 dBm/Hz. Datasets for BLS training include the packet size of each device, device number, device and MEC server locations, and channel gain between devices and all MEC servers.

The proposed strategies are compared with BLSO and evaluated based on the optimization problems, where each employs a distinct objective function. The Gurobi 9.5.2 commercial solver is used to obtain optimal labels for supervised training. Each simulation is executed 100,000 times with random device–server distributions, and average values are reported as final results.

The simulation parameters follow the IEEE 802.11ax specification [45], as summarized in Table 4.1.

4.4.2 Evaluation Metrics

The performance of BLS is analyzed with respect to two primary aspects:

(1) **BLS Learning Performance** — evaluated in terms of *training time* and *accuracy* for different percentages of training data.

(2) **Network Performance Metrics** — evaluated using:

- **Network capacity (Mbps)**: overall throughput of all devices in the network.
- **Network latency (s)**: the average time required for successful data transmission between devices and MEC servers.

Table 4.1: Simulation parameters and settings of eLOW.

Parameter	**Value**
Channel bandwidth	80 MHz
Attenuation constant	3.5
Transmit power	500 mW
Shadowing attenuation	4 dB
Decorrelation distance	1 m
Wall attenuation	0 dB
No. of devices	$50 \sim 150$
No. of MEC servers	$3 \sim 10$
No. of feature nodes for BLS	160
No. of enhancement nodes for BLS	1000
No. of adding enhancement nodes for BLS	50
Regularization coefficient for BLS	2×10^{-10}
Shrink coefficient for BLS	0.9
No. of incremental steps for BLS	5

The detailed BLS learning performance is shown in Table 4.2, which demonstrates that the accuracy improves with a larger proportion of training data. When 90 % of the data are used for training, BLS achieves 94.43 % accuracy within 5.37 seconds, showing an optimal trade-off between training time and accuracy.

Table 4.2: Training time and accuracy of BLS evaluation.

Percentage of training data	**40%**	**50%**	**60%**	**70%**	**80%**	**90%**
Training time (s)	2.30	2.84	3.80	4.48	4.57	5.37
Accuracy (%)	87.27	88.16	90.11	91.75	91.88	94.43
Increased accuracy per second	–	1.64%	2.03%	2.41%	1.44%	3.18%

4.4.3 Simulation Results and Discussion

Impact of Device Density

Figure 4.3 illustrates how the average network capacity and latency vary with the number of devices. As the number of devices increases, the network experiences higher congestion and interference, leading to reduced capacity and increased latency.

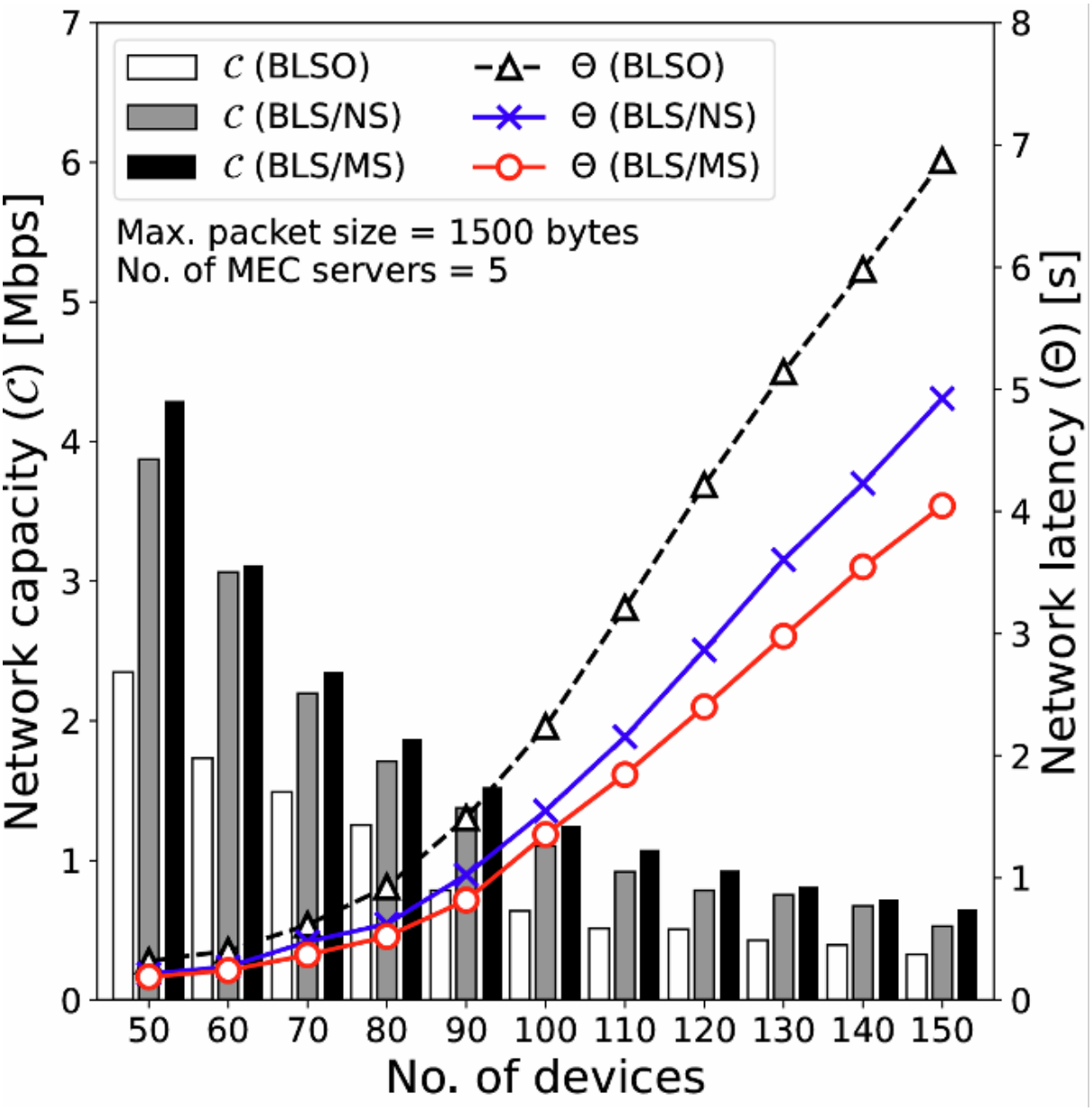


Figure 4.3: Average network capacity and latency versus number of devices.

For 50 devices, BLS/MS achieves a capacity of 4.29 Mbps, which is 1.83 times and 1.11 times higher than BLSO and BLS/NS, respectively. When the device count increases

to 150, BLS/MS still maintains superior performance with 0.64 Mbps capacity and 3.55 s latency, corresponding to reductions of 0.51 times and 0.72 times in latency compared with BLSO and BLS/NS.

Impact of MEC Server Density

Figure 4.4 shows that as the number of MEC servers increases, average network capacity improves while latency decreases. More MEC servers provide devices with greater selection flexibility, mitigating congestion and balancing workloads.

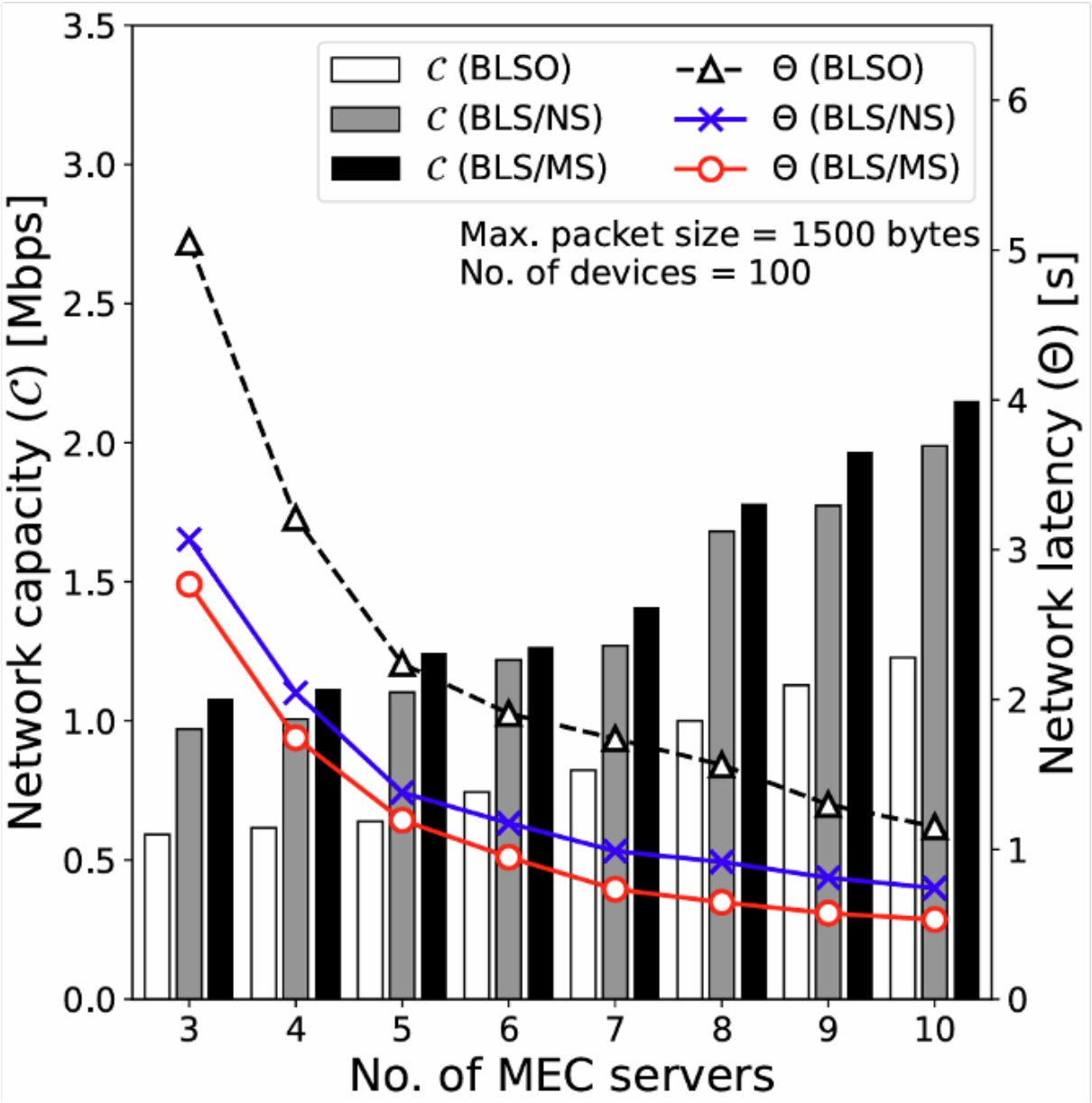


Figure 4.4: Average network capacity and latency versus number of MEC servers.

For 3 servers, BLS/MS achieves 1.08 Mbps capacity, 1.81 times and 1.04 times higher than BLSO and BLS/NS, respectively. At 10 servers, it achieves 2.15 Mbps capacity and 0.53 s latency, significantly outperforming other schemes.

Impact of Packet Size

Figure 4.5 illustrates that increasing packet size leads to longer transmission times and hence higher latency. However, the effect on network capacity remains marginal (within 8 %) since both total packet size and total transmission time scale proportionally.

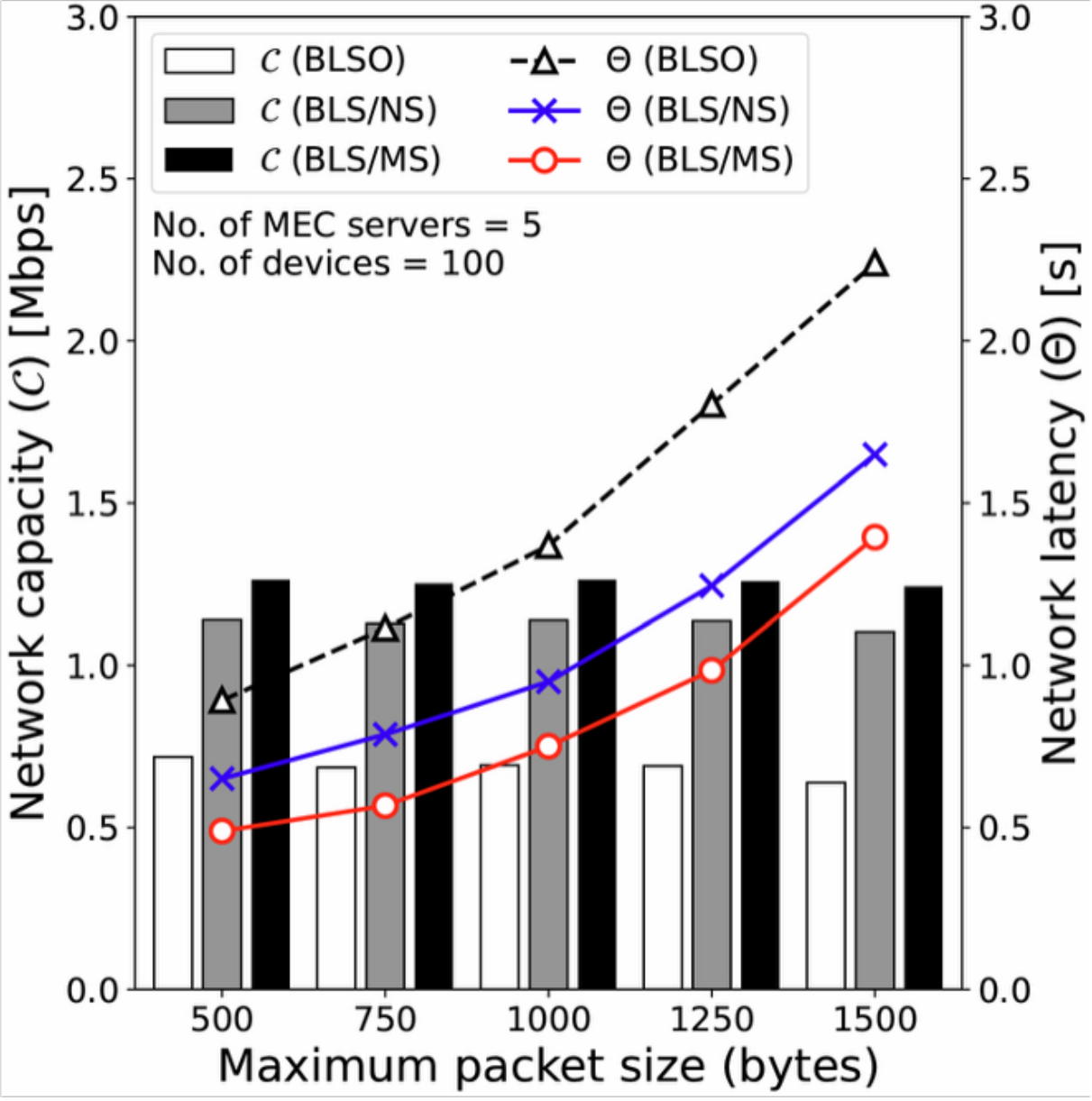


Figure 4.5: Average network capacity and latency versus packet size.

Discussion

These results clearly show that BLS/MS outperforms BLSO and BLS/NS in both network capacity and latency. While BLSO focuses solely on minimizing computational delay and BLS/NS prioritizes spatial proximity, BLS/MS effectively balances link quality, interference, and device–server pairing. Consequently, it achieves more stable and efficient network operations, particularly in dense deployment scenarios.

4.5 Summary

This chapter investigated the eLOW scheme, which serves as the latency-oriented module of the proposed CORE framework in MSWNs. By focusing on device–server association in MEC-enabled MSWNs, eLOW aims to reduce end-to-end service latency while preserving high network capacity under dense and interference-prone conditions.

Building on the advantages of the Broad Learning System, eLOW formulates server selection as a supervised learning problem and adopts BLS as a fast and incrementally updatable decision engine. BLS-based optimization strategies were introduced: BLSO, which minimizes total service time; BLS/NS, which prioritizes proximity to reduce propagation delay; and BLS/MS, which emphasizes link quality and interference mitigation to sustain high data rates.

A comprehensive system model for MSWNs was established, including channel and interference characterization, and performance metrics for network latency and capacity were derived. Numerical simulations under varying device densities, server densities, and packet sizes demonstrated that BLS/MS consistently outperforms BLSO and BLS/NS in terms of both network latency and capacity. The results further confirmed that BLS provides high decision accuracy with modest training time, and its incremental learning capability allows the server-association policy to adapt efficiently to changing network conditions.

Overall, the eLOW scheme offers an effective solution to the latency bottleneck in multi-server wireless networks by combining BLS-based server selection with latency- and capacity-aware optimization. It complements the capacity-oriented eCAP scheme developed in Chapter 3 and provides a low-latency foundation for the unified power and

interference management scheme (uPOW) presented in Chapter 5.

Chapter 5

Unified Power Management Scheme

Following the eCAP scheme and the eLOW scheme, this chapter presents the third and final component of the proposed CORE framework: the Unified Power Management (uPOW) scheme. Unlike the previous chapters, which focus primarily on transmission efficiency and server allocation, this chapter addresses a critical challenge arising in edge computing–enabled WMNs, namely the efficient execution of distributed computation tasks under interference and energy constraints.

In edge computing scenarios, user devices offload computation tasks to edge servers through multihop wireless transmissions. As network density increases and multiple tasks are processed concurrently, task completion time becomes highly sensitive not only to path selection and server selection, but also to mutual interference and transmit power consumption. Even when transmission paths are optimized and server loads are well balanced, uncoordinated power usage among nodes can lead to severe interference coupling, excessive energy expenditure, and prolonged task execution delays. These effects limit the performance of WMNs.

The uPOW scheme is designed to address this challenge by enabling unified and intelligent coordination of communication and computation-related decisions. Within the CORE framework, uPOW jointly considers server selection, multihop path selection, and transmit power control, rather than optimizing them independently. Specifically, in the deciding module, a cooperative learning engine integrates BLS and Q-learning to select the server and the multihop path. In the sharing module, these decisions are synchronized across nodes via edge servers to avoid conflicting transmission behaviors in dense

deployments. Finally, in the switching module, a CTPC mechanism dynamically regulates transmit power in response to interference levels, mobility, and link stability, forming a closed-loop control process.

The novelty of the uPOW scheme lies in its unified three-stage optimization structure. By simultaneously considering task execution efficiency, interference mitigation, and energy conservation, uPOW enables robust and scalable operation of WMNs. This chapter completes the CORE framework by providing a power- and interference-aware optimization layer that supports autonomous, scalable, and sustainable task execution in B5G wireless systems.

The remainder of this chapter is organized as follows. Section 5.1 reviews related work on power control and interference management in WMNs. Section 5.3 presents the system model for uPOW. Section 5.4 details the three-stage uPOW. Section 5.5 evaluates the proposed scheme through numerical simulations. Finally, Section 5.6 concludes the chapter.

5.1 Research Background

With the evolution toward B5G and future 6G networks, emerging applications demand unprecedented levels of connectivity, ultra-low latency, and high network capacity. Moreover, as 6G aims to support AI-native services, future wireless systems must be capable of delivering real-time AI inference and decision support in addition to communication services [60,61]. An increasing number of smart devices, such as autonomous drones, wearable sensors, and intelligent vehicles, will generate computation-intensive tasks requiring immediate AI processing. However, most of these devices are restricted by processing power and battery capacity, making local execution of complex AI models impractical. These ambitious requirements pose significant challenges to the design of MEC-enabled wireless networks, especially in scenarios with high device density and dynamic user mobility.

MEC brings computing and storage capabilities closer to the network edge, enabling real-time data processing and low-latency services. Task allocation and offloading in MEC has emerged as a pivotal strategy to alleviate the computational burden on resource-constrained devices, enabling efficient processing and reduced latency in various applica-

tions [62, 63]. However, conventional MEC systems rely on direct device-to-server communication, which severely limits the scalability and performance of wireless networks in high-density environments [47, 64]. Moreover, future wireless communications are anticipated to operate at higher frequencies (e.g., mmWave and THz) to support massive data transmission, significantly escalating infrastructure deployment costs over extensive geographic areas [65, 66]. In this context, multihop wireless networking provides a cost-effective and scalable alternative by leveraging device relaying and localized communication. MWMNs have emerged as a promising architecture, where devices can communicate through multi-hop wireless links to offload their tasks to ESs distributed within the system. This resolution effectively supports massive connectivity and achieve ubiquitous computing. Multihop networking plays a crucial role in extending the communication range and enhancing the reliability of wireless networks, particularly in scenarios lacking fixed infrastructure [67, 68], such as the rapid deployment of temporary communication networks in disaster-stricken zones, multihop relay support for rescue teams connecting to remote command centers, and flexible wireless network setups in large-scale events or stadiums using UAVs and mobile ESs to support surveillance, sensing, and emergency communications [69, 70]. This design enables better load balancing, reduced latency, and improved network coverage. Nevertheless, the design of highly efficient MWMNs remains challenging, as it necessitates comprehensive consideration of application-specific QoS requirements, efficient data forwarding strategies, robust wireless communications, as well as resource restrictions. This includes optimal server selection, dynamic route planning to accommodate user mobility, and interference-aware power control, all critical to ensuring robust performance in complex and time-sensitive environments.

Recently, AI-driven approaches have been widely adopted to optimize resource allocation and routing decisions in wireless networks. In the community of MEC, extensive efforts have been devoted to task offloading, server selection, and resource allocation by leveraging AI techniques, especially deep RL. Meanwhile, RL-based algorithms [38] are widely developed to optimize multihop routing by considering wireless link quality metrics such as SNR and SINR. However, deep RL methods often suffer from sample inefficiency, training instability, and high computational cost, especially when deployed in large-scale and dynamic multihop MEC networks [71]. These limitations hinder their real-

time applicability in edge computing environments with stringent latency and scalability requirements.

To address these challenges, the broad learning system (BLS) approach [54] has demonstrated its effectiveness in handling large-scale data and adapting to dynamic environments, making it a promising candidate for resource allocation and server selection in dynamic MEC networks. BLS offers several advantages, such as fast training speed without requiring deep architectures, excellent generalization ability with limited training samples, and incremental learning capability to adapt to changing environments. Recent studies have applied BLS to classification and resource allocation tasks in edge computing environments, demonstrating better scalability and lower training latency compared to deep RL-based methods [56, 57, 72]. These characteristics make BLS particularly suitable for large-scale and dynamic MEC scenarios. However, existing works often treat server selection and multihop routing as isolated problems, ignoring their mutual coupling in practical MEC scenarios. Additionally, few studies have addressed the challenge of adaptive transmit power control in dense wireless networks to mitigate interference while maintaining throughput performance.

To bridge these research gaps, this dissertation investigates the joint optimization of server selection, multihop routing, and transmit power control in MWMNs for supporting the reliable communication and computation of users, as well as optimizing network capacity. We proposed a three-stage uPOW scheme. We summarize our major contributions as follows:

- We design a comprehensive scheme for MEC-enabled MWMNs, spanning from the application layer down to the physical layer. The scheme jointly considers user-server association, dynamic routing, and interference-aware power control under realistic device mobility, communication constraints, and QoS requirements. The proposed scheme addresses the challenges of optimizing resource allocation and multihop communication under stringent latency and throughput constraints, which are critical to ensuring reliable network operation in dynamic and densely deployed MEC scenarios.

- We transformed the joint network optimization problem into a three-stage decision process and proposed the uPOW scheme to jointly solve the problem. In the first

stage, a BLS is employed to efficiently determine the optimal server allocation. In the second stage, a SINR-based Q-learning algorithm is proposed to construct multihop routing paths that adapt to varying network topology and link conditions. Moreover, we incorporate a transmission power control algorithm to further reduce interference and improve link efficiency.

- Extensive simulation results validate the superiority of the proposed methods over existing algorithms, demonstrating significant improvements in network capacity, task completion time, interference power, and QoS. This research contributes to the development of intelligent, adaptive, and scalable MEC-enabled wireless networks for future 6G systems.

5.1.1 Related Works

This section reviews existing studies related to the proposed uPOW scheme. The discussion is divided into three parts: edge-enabled IoT systems, wireless multihop networks, and BLS-based approaches.

5.1.2 Edge-Enabled IoT Systems

MEC has emerged as a promising solution to overcome the computation limitations of mobile devices by offloading tasks to edge servers. Extensive research has been conducted on task allocation and resource optimization strategies in MEC and IoT environments. Wang et al. [73] investigated a joint task, spectrum, and power allocation problem for MEC-based networks with heterogeneous task requirements, and developed a multi-stack RL algorithm to accelerate convergence and improve performance. Chen et al. [74] studied dynamic task allocation and service migration in edge-cloud IoT systems under highly dynamic user demands and mobility, proposing a deep deterministic policy gradient-based algorithm to minimize cloud server load while satisfying latency and migration constraints. Chen et al. [75] further explored distributed joint task and computing resource allocation in heterogeneous edge networks using multi-agent deep reinforcement learning (DRL) and sigmoidal programming. Lin et al. [76] addressed AI service placement in multi-user MEC systems, optimizing CPU frequency scaling and bandwidth allocation. Moreover,

UAV-assisted edge computing has gained attention for providing computation services in dynamic environments. Goudarzi et al. [77] proposed an optimization framework for UAV-assisted vehicular edge computing to minimize age of information, energy consumption, and rental costs using a soft actor-critic-based RL algorithm. Tran et al. [78] tackled UAV relay-assisted IoT communication, optimizing resource allocation and UAV trajectory to serve more devices under latency and storage constraints.

5.1.3 Wireless Multihop Networks

In wireless multihop networks, extensive research has focused on optimizing task offloading, routing, and resource allocation. Ahmed et al. [79] proposed a proximal policy optimization-based RL algorithm to minimize task execution delay in multihop vehicular task offloading. Nguyen et al. [80] presented a UAV-assisted multihop edge computing architecture using deep Q-learning for task partitioning and offloading. Zhao et al. [81] proposed a two-layer DRL framework for RSU-to-Everything networks, using LSTM-based models to predict neighbor behavior for offloading decisions. In federated learning, Chen et al. [82] proposed a framework over wireless mesh networks with in-network model aggregation and joint optimization of routing and spectrum allocation. Ji et al. [83] introduced a GNN-assisted DRL for V2X communications, modeling V2V interference as graph edges for distributed resource allocation. Akyildiz et al. [84] proposed a mobility-driven multihop task offloading and resource optimization protocol for connected vehicular networks.

5.1.4 Broad Learning System

Machine learning (ML) techniques have been widely used for task allocation and offloading in wireless networks, including deep learning-based methods [85, 86]. Recently, BLS has gained interest, thanks to its non-deep learning framework, which offers fast learning and low computational complexity. Xu et al. [55] introduced recurrent BLS for time series prediction, enhancing its ability to handle sequential data. Chi et al. [57] proposed a BLS-based task offloading scheme named BLSO in IIoT networks, demonstrating superior training efficiency and adaptability to dynamic networks. These studies highlight the increasing role of DRL and BLS in managing the complexity and dynamics of modern

MEC and wireless multihop environments.

5.2 Motivation

Although extensive studies have been conducted on server allocation, resource management, and multihop routing in MEC-enabled wireless networks, most existing works treat server selection and multihop path planning as isolated optimization problems. In future MWMN scenarios, server selection and routing decisions are inherently coupled due to the convergence of communication and computation, jointly impacting application QoS and network capacity. However, making these decisions remains challenging due to the dynamic network topology, interference, and device characteristics.

Furthermore, few existing studies have addressed adaptive transmit power control in dense multihop wireless networks to balance interference mitigation and throughput enhancement. As device density increases, interference among nodes becomes a dominant factor limiting network scalability and reliability. Therefore, designing a unified framework that simultaneously handles server association, multihop routing, and transmission power adjustment is crucial for achieving high-performance, interference-aware, and energy-efficient MEC-enabled multihop systems.

To address these challenges, this dissertation proposes the uPOW scheme. The uPOW scheme jointly integrates three modules: BLS-based server allocation, Q-learning-based path selection, and CTPC, to form a unified cross-layer optimization scheme. This integrated approach enables efficient decision-making across the computation, communication, and physical layers, providing improved scalability, latency reduction, and interference management for dynamic and mission-critical network scenarios.

5.3 System Model

This section presents the system model of the proposed uPOW framework. We first describe the overall network architecture and its operating mechanisms, followed by the formulation of optimization objectives and constraints. The system model consists of three layers: user devices, edge servers, and an orchestration system, which together form a multi-server wireless multihop network (MWMN) enabling efficient computation

offloading and communication management.

5.3.1 Network Model

This chapter considers a three-tier MWMN, as presented in Figure 5.1, consisting of user devices (UDs) at the bottom layer, ESs at the middle layer, and an orchestration system (OS) at the top layer. UDs ($\mathcal{U} = \{1, 2, \cdots, U\}$) are randomly distributed within the network. A set of ESs ($\mathcal{E} = \{1, 2, \cdots, E\}$) is deployed in a grid-based pattern to support task requests by ensuring resources and services. UDs can connect to ESs through wireless multihop links established among devices, enabling real-time data collection and task processing to support various edge-aware applications.

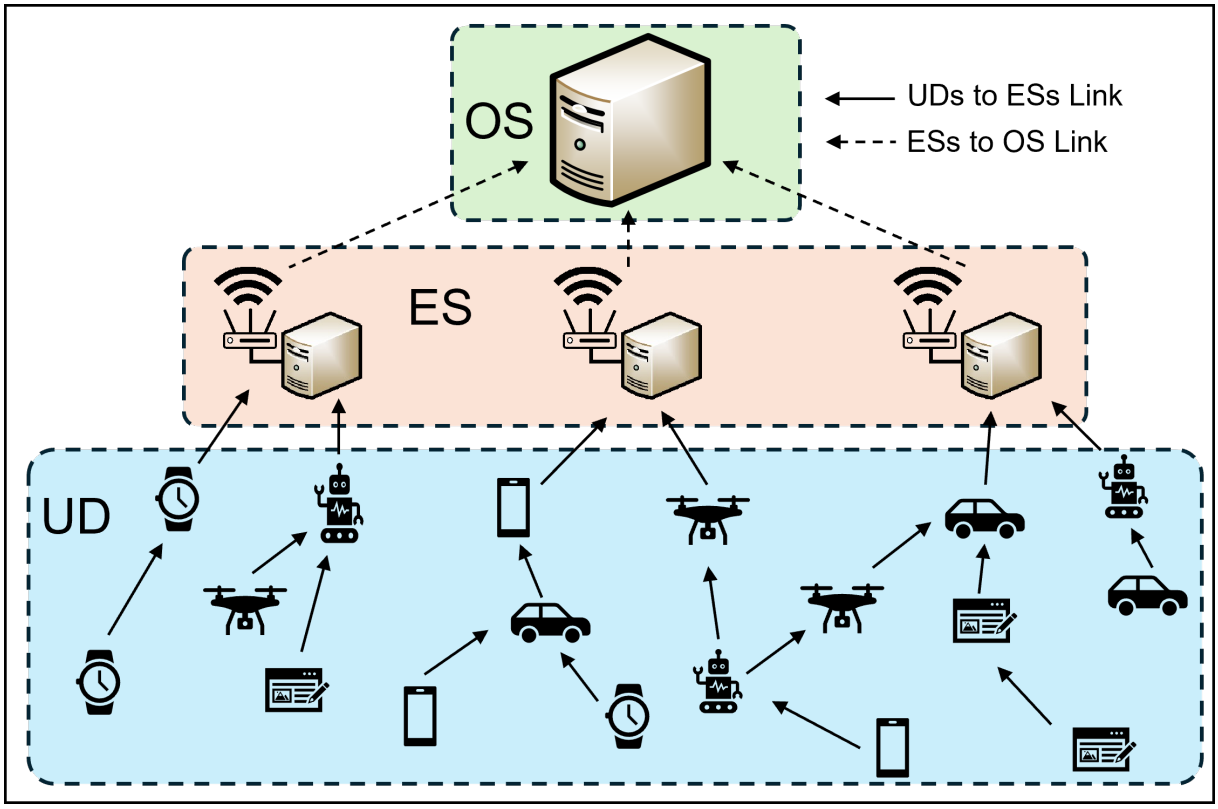


Figure 5.1: Illustration of the system model.

The ESs are connected to the OS, allowing for the sharing of gathered information from all UDs within the network. UDs within the transmission range of others can act as relay devices, receiving and forwarding the data until the data reaches the target ES. ESs serve as computational centers located closer to UDs. Each ES maintains a real-time database reflecting its computational resources. All devices in the network operate

follows a unified communication protocol, which enables them to operate on the same frequency band and share the same maximum access bandwidth. This assumption simplifies system modeling and aligns with typical configurations in standardized wireless communication environments. As for OS, it is located in the network area and oversees global network optimization and management. It aggregates network-wide information, executes sophisticated machine learning algorithms for optimized ES allocation and multihop path selection, and maintains databases containing historical mobility patterns, server capabilities, and computational demands.

The UDs are categorized into two types based on their mobility characteristics and role within the network:

- Predefined-Path High-speed Devices (PHDs): These devices move at high speed along known paths, which have been pre-recorded and analyzed by the OS. Their predictable mobility allows preemptive adjustments to routing and server assignments. Examples include high-speed delivery UAVs, autonomous vehicles, and mobile robots deployed for emergency response or automated industrial inspection.
- Random-Path Low-speed Devices (RLDs): These devices exhibit relatively slow movement with unpredictable paths, typically due to human-centric usage patterns. Their unpredictable mobility requires periodic location updates and dynamic network adjustments. Examples include smartphones carried by pedestrians/people, wearable health monitoring devices used by medical patients, and handheld terminals for workers in industrial environments.

5.3.2 Network Operation

The operation of the MWMN is structured into three primary phases, i.e., server allocation, multihop path selection, and network topology update.

Server Allocation Phase

Server allocation refers to allocating each task to an appropriate ES that can ensure its QoS requirements, ultimately achieving system efficiency. Due to limited resources, each UD offloads its task to an ES. At the beginning of making a decision, the real-time state of all UDs is obtained, including their locations, task requests, and the service

delay requirements. Then, the OS decides for every UD by comprehensively considering tasks' QoS, ESs' resource capacity, and the quality of wireless links established within the system. We use a row vector of binary indicators $b_{u,e}(t) = \{0, 1\}, u \in \mathcal{U}, e \in \mathcal{E}$ to represent the decision of server allocation at time t. $b_{u,e} = 1$ indicates that the task of UD u is allocated to be processed by ES e, and $b_{u,e} = 0$ otherwise. Since each UD's task can only be offloaded to a single ES, the following constraint is applied:

$$\mathbf{C1} : \sum_{e \in \mathcal{E}} b_{u,e} = 1, \quad \forall u \in \mathcal{U}. \tag{5.1}$$

After obtaining a server allocation decision, the task estimation time $t_{u,e}^{est}$ of a task offloaded from u to an ES e can be obtained, which includes the task uploading time, task processing time, and the resultant downloading time [57], i.e.,

$$t_{u,e}^{est} = \hat{t}_{u,e}^{up} + \hat{t}_{u,e}^{down} + t_{u,e}^{proc}, \tag{5.2}$$

where $t_{u,e}^{proc}$ is the task processing time and is determined by its task size (L_u) and processing speed of allocated ES e (σ_e):

$$t_{u,e}^{proc} = \frac{L_u}{\sigma_e}. \tag{5.3}$$

Here, $\hat{t}_{u,e}^{up}$ and $\hat{t}_{u,e}^{down}$ represent the estimated task upload time and download time, respectively. Let L_u^{res} denote the estimated size of the processed task results for UD u. Then, the estimated upload and download times can be calculated as

$$\hat{t}_{u,e}^{up} = \frac{L_u}{R_{u,e}}, \quad \hat{t}_{u,e}^{down} = \frac{L_u^{res}}{R_{u,e}}. \tag{5.4}$$

Here, $R_{u,e}$ is the transmission rate between the UD and the ES, i.e.,

$$R_{u,e} = B \cdot \log_2 \left(1 + SINR_{u,e}\right), \tag{5.5}$$

where B is the channel bandwidth, $SINR_{u,e}$ is the SINR between u and e calculated by

$$SINR_{u,e} = \frac{G_{u,e} \cdot P_u}{\eta \cdot B + \sum_{k \in \mathcal{K}} G_{k,e} \cdot P_k}, \tag{5.6}$$

where P is the transmit power and η is the noise level. $\mathcal{K}$ is the set of interference UDs. We consider the worst-case scenario, where all UDs in the network contribute to interference during transmission. As shown in Figure 5.2, the interference model considers

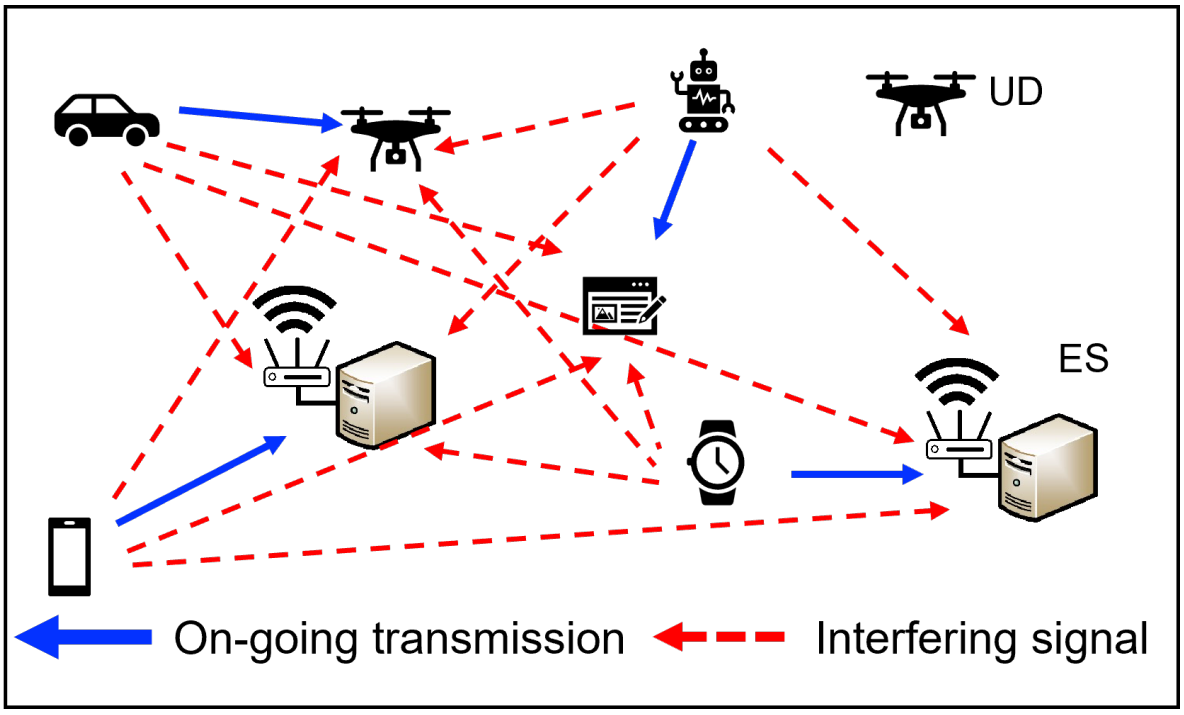


Figure 5.2: Illustration of the interference model.

all transmitting links during an ongoing link between UDs and ESs. $G_{u,e}$ is the power ratio between u and e, as determined by the log-distance path loss model [17], i.e.,

$$G_{u,e} = \frac{1}{10^{\left(\frac{PL_{u,e}}{10}\right)}}, \tag{5.7}$$

where

$$PL_{u,e} = 20 \cdot \log_{10}(d_0) + 10 \cdot \zeta \cdot \log_{10}(\frac{d_{u,e}}{d_0}) - \omega_{ij} + \psi. \tag{5.8}$$

Here, d_0 is the decorrelation distance and is set to 10 m in this research. ζ is an attenuation constant and ω is the wall attenuation. ψ is shadowing attenuation and $d_{u,e}$ is the distance between UD u and ES e.

To meet QoS requirements, $t_{u,e}^{est}$ of each task must not exceed the task tolerable time (τ_u) defined by each UD:

$$\mathbf{C2}: \quad t_{u,e}^{est} \leq \tau_u, \quad \forall u \in \mathcal{U}. \tag{5.9}$$

Multihop Path Selection Phase

After the ES allocation, the OS needs to select wireless multihop paths for efficient data transfer between each UD and its assigned ES. Specifically, for each UD, a multihop path consisting of intermediate UDs is established to connect it to the assigned ES.

To represent the multihop path clearly, we employ an adjacency matrix defined as

$$\boldsymbol{y}_u = [y_{i,j}]_{(|\mathcal{U}|)\times(|\mathcal{U}|+|\mathcal{E}|)}, \tag{5.10}$$

where $i \in \mathcal{U}$ is the transmitting node and $j \in \mathcal{U} \cup \mathcal{E}$ is the receiving node. Specifically, the source node is $u \in \mathcal{U}$, and the destination node is $e \in \mathcal{E}$. $y_{i,j} = \{0, 1\}$. $y_{i,j} = 1$ means the link between i and j is included in the path. To guarantee the connectivity and continuity of the multihop path from an UD to its designated ES, the following constraints are applied:

$$\mathbf{C3}: \begin{cases} \sum_{j\in\mathcal{U}\cup\mathcal{E}} y_{u,j} = 1, \sum_{j\in\mathcal{U}\cup\mathcal{E}} y_{j,u} = 0, \\ \sum_{j\in\mathcal{U}\cup\mathcal{E}} y_{j,e} = 1, \sum_{j\in\mathcal{U}\cup\mathcal{E}} y_{e,j} = 0, \\ \sum_{j\in\mathcal{U}\cup\mathcal{E}} y_{i,j} = \sum_{j\in\mathcal{U}\cup\mathcal{E}} y_{j,i} \leq 1, \end{cases} \tag{5.11}$$

where u is the source node (the originating UD), which only sends information outward, and e is the destination ES, which only receives information. All intermediate nodes must have equal in-degree and out-degree (at most 1) to ensure the continuity and uniqueness of the selected path [87].

To ensure high-quality data transmission in MWMNs, the path selection strategy should prioritize not only connectivity but also the efficiency of communication. In this context, average E2E throughput serves as a critical performance metric that reflects the overall transmission capability of the selected path [88]. By maximizing the average throughput, the system encourages the use of high-quality wireless links with better channel conditions and lower interference, thereby improving data rate and reducing delay across multiple hops. The average E2E throughput is defined as the mean of the transmission rates across all links in the selected multihop path. The average throughput for a UD, $R_{u,e}^{avg}$, is

$$R_{u,e}^{ave} = \frac{\sum_{i\in\mathcal{U}\cup\mathcal{E}} \sum_{j\in\mathcal{U}\cup\mathcal{E}} y_{i,j} \cdot R_{i,j}}{\sum_{i\in\mathcal{U}\cup\mathcal{E}} \sum_{j\in\mathcal{U}\cup\mathcal{E}} y_{i,j}}. \tag{5.12}$$

For each selected multihop path, the task service time (t_u^{ser}) must not exceed the task estimation time (t_u^{est}). To ensure the timeliness and reliability of task execution in MWMNs, it is essential to guarantee that the task service time for each task remains within its estimated deadline. This constraint is particularly important for latency-sensitive applications such as real-time monitoring, autonomous control, and emergency

communication, where task completion beyond the task tolerable time may lead to system failure or degraded QoS. Therefore, we impose a delay-bound constraint to ensure that all selected transmission paths and computing assignments comply with each user's time requirement, i.e.,

$$\mathbf{C4}: \quad t_{u,e}^{ser} \leq t_{u,e}^{est}, \quad \forall u \in \mathcal{U}, \tag{5.13}$$

where $t_{u,e}^{ser}$ consists of the task upload time ($t_{u,e}^{up}$), the task processing time ($t_{u,e}^{proc}$), and the result download time ($t_{u,e}^{down}$):

$$t_{u,e}^{ser} = t_{u,e}^{up} + t_{u,e}^{down} + t_{u,e}^{proc}. \tag{5.14}$$

$t_{u,e}^{up}$ and $t_{u,e}^{down}$ depend on the actual size of the transmitted data and transmission rates, which can be calculated by

$$t_{u,e}^{up} = \frac{L_u}{R_{u,e}^{ave}}, \quad t_{u,e}^{down} = \frac{L_u^{res}}{R_{u,e}^{down}}, \tag{5.15}$$

where R_u^{ave} is the average transmission rate across the multi-hop path established between UD u and the target ES for supporting the edge computing service procedure of its task requests. $R_{u,e}^{down}$ is the download transmission rate.

In the upload phase, each UD transmits its task to the ES via a multihop wireless path. Therefore, the average data rate across the multihop path, $R_{u,e}^{ave}$, is used for calculating the upload time. In the download phase, ESs transmit the result directly to UDs via broadcast. Since all ESs may transmit simultaneously, they introduce mutual interference, and thus, the individual download transmission rate is used to account for this effect. In the download phase, the SINR experienced by UD u when receiving results from ES e is given by:

$$SINR_{u,e}^{down} = \frac{G_{u,e} \cdot P_e}{\eta \cdot B + \sum_{e' \in \mathcal{E}, e' \neq e} G_{u,e'} \cdot P_{e'}}, \tag{5.16}$$

where P_e denotes the transmit power of ES e, and the interference term $\sum_{e' \in \mathcal{E}, e' \neq e} G_{u,e'} \cdot P_{e'}$ captures the total interference from all other simultaneously broadcasting ESs. Then, the download transmission rate is calculated as:

$$R_{u,e}^{down} = B \cdot \log_2 \left(1 + SINR_{u,e}^{down}\right). \tag{5.17}$$

It is worth noting that multihop transmission is adopted primarily due to the practical limitations of wireless coverage and resource availability in large-scale MEC scenarios. In

many real-world environments (e.g., disaster-affected regions, large-scale events, or areas lacking direct ES coverage), single-hop connectivity is typically unavailable or severely constrained. Therefore, multi-hop offloading becomes not only beneficial but often necessary. However, for cases where single-hop transmission can directly achieve lower latency and better QoS, the proposed scheme naturally prefers single-hop routing, as reflected by the path selection stage. Hence, constraint **C4** essentially ensures that only feasible multi-hop or single-hop paths meeting the latency requirements are selected.

If the selected multihop path fails to satisfy constraint **C4**, the OS re-executes the path selection algorithm to determine a feasible path. Moreover, in certain scenarios (e.g., when a UD is located close to an ES), the single-hop path may offer lower latency compared to multihop alternatives. In cases where the OS repeatedly fails to find a feasible multihop path satisfying constraint **C4** after multiple attempts, it will default to the single-hop link provided that the single-hop service time can meet $t_{u,e}^{est}$. To ensure convergence and bounded computational cost, the OS explores a finite set of candidate paths (limited by hop count and network topology). If no feasible path satisfying **C4** is found within a predefined number of attempts (e.g., 200), the system will fall back to single-hop transmission if it can satisfy the delay constraint. Finally, the allocation and routing results are disseminated from the OS to all ESs, which then broadcast this information to respective UDs. UDs adopt their ES assignments and paths to initiate task transmission.

Constraints **C2** and **C4** form a hierarchical structure that governs task latency control in our proposed scheme. To clearly describe the relationship among the three timing parameters—task tolerable time, task estimation time, and task service time—we outline the dependency as follows:

- τ_u is defined by each UD to represent its maximum tolerable latency, serving as the initial QoS requirement.
- $t_{u,e}^{est}$ is computed by the OS in the server allocation stage based on expected transmission and computation delays. It is constrained by **C2** to satisfy $t_{u,e}^{est} \leq \tau_u$.
- $t_{u,e}^{ser}$ is measured after multihop routing and power control are finalized. It must remain within the previously estimated value, satisfying constraint **C4** as $t_{u,e}^{ser} \leq t_{u,e}^{est}$.

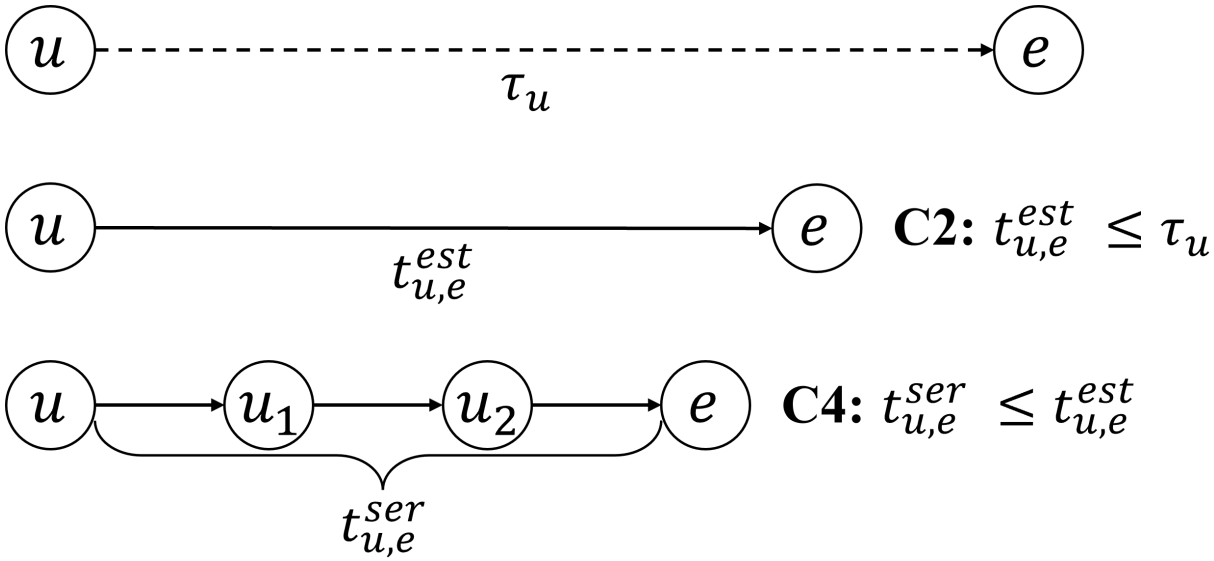


Figure 5.3: Illustration of the relationship between **C2** and **C4**.

This layered design ensures that each task execution path not only adheres to the user's QoS requirement but also maintains feasibility throughout network dynamics. The relationship between **C2** and **C4** is shown in Figure 5.3.

Network Topology Update Phase

The third phase is the network topology update phase. This phase implements periodic network updates to adapt to system dynamics, especially device mobility, using two strategies for different types of UDs.

The system works continuously in an infinite time horizon with discrete slots. At the beginning of each time slot, ESs collect the information from all their associated UDs. Then, it will upload this gathered information in addition to their own state information to the OS. To tackle the mobility of UDs and the dynamics of task requests, as well as the resources of ESs, the formulated decisions should be dynamically updated based on the real-time system state.

Due to their predictable movement, ES assignments and multihop path selections for PHDs were determined and adjusted by the OS in the previous phase. During transit, PHDs autonomously execute these path and server adjustments, incurring manageable migration overheads.

For RLDs, at each time slot, the OS recalculates the optimal multihop paths and ES

assignments based on the current location information uploaded by these devices. If a recalculated route differs from the previously assigned route, the OS proactively notifies the RLD with the updated path information. Otherwise, the RLD implicitly continues using the previously assigned route, thus minimizing redundant signaling overhead and ensuring resource efficiency.

5.3.3 Problem Formulation

In MWMNs, efficiently handling large-scale task offloading while maintaining reliable communication quality is critical to ensuring overall network performance. Motivated by these requirements, we formulate a joint optimization problem aimed at minimizing task service time, maximizing link quality, and enhancing E2E throughput. OS performs optimization for two primary objectives: the first objective is to minimize the task estimation time and maximize SINR to ensure that each UD's tasks are processed efficiently while maintaining reliable wireless communication links. The second objective is to maximize the average E2E throughput for transmissions between UDs and ESs to guarantee efficient data delivery, minimize transmission delays, and improve the overall network performance of MWMN. The optimization problem in the OS is formulated as follows:

$$\begin{aligned} \mathbf{P1}: & \min_{b_{u,e}} \sum_{u \in \mathcal{U}} \left[\delta \cdot t_{u,e}^{est} - (1-\delta) \cdot SINR_{u,e} \right] \\ \mathbf{P2}: & \max_{y_{i,j}, t_{u,e}^{ser}} \quad \sum_{u \in \mathcal{U}} R_{u,e}^{ave} \\ & s.t. \quad \mathbf{C1}, \mathbf{C2}, \mathbf{C3}, \mathbf{C4}, \end{aligned} \tag{5.18}$$

where $\delta \in [0, 1]$ is a weighting coefficient that balances the trade-off between task estimation time and link quality. The OS performs optimization for two primary objectives sequentially, where the relationship between **P1** and **P2** is clearly structured to ensure optimal network performance. In the first stage **P1**, the optimization prioritizes assigning each UD to an appropriate ES by minimizing task estimation time and maximizing SINR. This ensures that τ_u of each UD is initially met. In the second stage **P2**, given the ES assignments determined by **P1**, the wireless multihop paths from UDs to their corresponding ESs are optimized to maximize the sum of average end-to-end throughput. Specifically, maximizing the average E2E throughput is mathematically equivalent to minimizing the transmission-related portion of the service time. For a given task size L_u, the

upload delay is expressed in Equation (5.15), which is a monotonically decreasing function of $R_{u,e}^{ave}$. Hence, by choosing the multihop path that yields the highest $R_{u,e}^{ave}$, the algorithm simultaneously drives $t_{u,e}^{ser}$ towards its minimum feasible value, thereby satisfying not only **C4** but also beyond. At this stage, a stricter constraint is applied, ensuring that $t_{u,e}^{ser}$ is not only within $t_{u,e}^{est}$ but also seeks to maximize throughput. Thus, the optimization follows a hierarchical approach: first satisfying latency and connectivity requirements through server assignment, then further refining network performance through path optimization.

In practical MWMNs, the formulated joint optimization problem presents significant challenges. First, future 6G networks will involve extremely dense deployments, resulting in high-dimensional optimization problems that complicate global solutions and demand efficient and scalable algorithms. Second, the problem involves strong coupling among server assignment, link quality (e.g., SINR), and multihop routing, as these decisions directly influence each other. Strict QoS constraints also increase complexity, as they depend on shared resources and multihop contention.

Computationally, the problem combines discrete and continuous variables within nonlinear objectives and constraints. Even simplified versions, such as optimal multihop routing under delay and SINR constraints, are known to be NP-hard. Dynamic environments with rapidly evolving network topologies and task arrivals exacerbate these issues, necessitating adaptive learning-based or heuristic methods capable of approximating near-optimal solutions with manageable computational overhead.

5.4 Unified Power Management Scheme

In wireless multihop networks, transmit power control plays a critical role in balancing throughput, energy efficiency, and interference mitigation. A fixed transmit power setting may simplify implementation but often results in excessive interference and energy waste, particularly when multiple devices transmit concurrently within overlapping ranges. To address these challenges, this section proposes a unified power management (uPOW) scheme that coordinates both individual and network-level power adjustments.

As shown in Fig. 5.4, the proposed uPOW scheme consists of three stages. After receiving information of all UDs from ESs, the BLS is executed at the OS to assign each UD to an optimal ES in stage one. This decision accounts for multiple factors and aims to

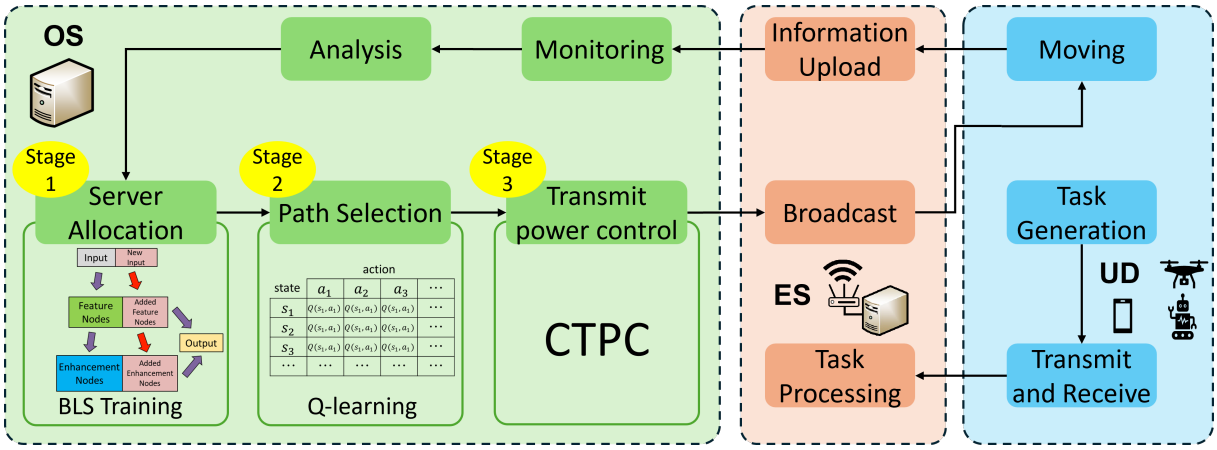


Figure 5.4: Overall framework of the proposed uPOW scheme.

minimize task estimation time and maximize SINR. In stage two, after server allocation, a Q-learning algorithm is applied to discover efficient multihop wireless routes from each UD to its selected ES. The Q-agent is trained using the SNR and SINR as a reward signal, while respecting routing constraints to ensure low interference and high throughput. In stage three, to further reduce the interference power, a CTPC algorithm is employed. This stage adaptively tunes each UD's transmission power based on its link performance deviation from the average throughput, effectively achieving energy-aware communication and network-wide SINR balancing. For mobile UDs, the uPOW scheme also incorporates distinct strategies tailored to different mobility types, ensuring route robustness and task delivery reliability under dynamic topology conditions. The flowchart of uPOW is shown in Fig. 5.5

5.4.1 Broad Learning System-based Server Allocation

In this subsection, the BLS is applied to the server allocation problem within the proposed MWMN. Rather than reintroducing the full theoretical derivation, this section focuses on how BLS processes the input features derived from the system model and produces the optimal ES selection for each UD.

For each decision epoch, the OS constructs the input feature matrix by combining device and server locations, packet sizes, the computational capacities of ESs, and large-scale channel gains. All features are normalized using the min–max normalization defined in Equation (4.4), and insignificant features are removed according to their importance

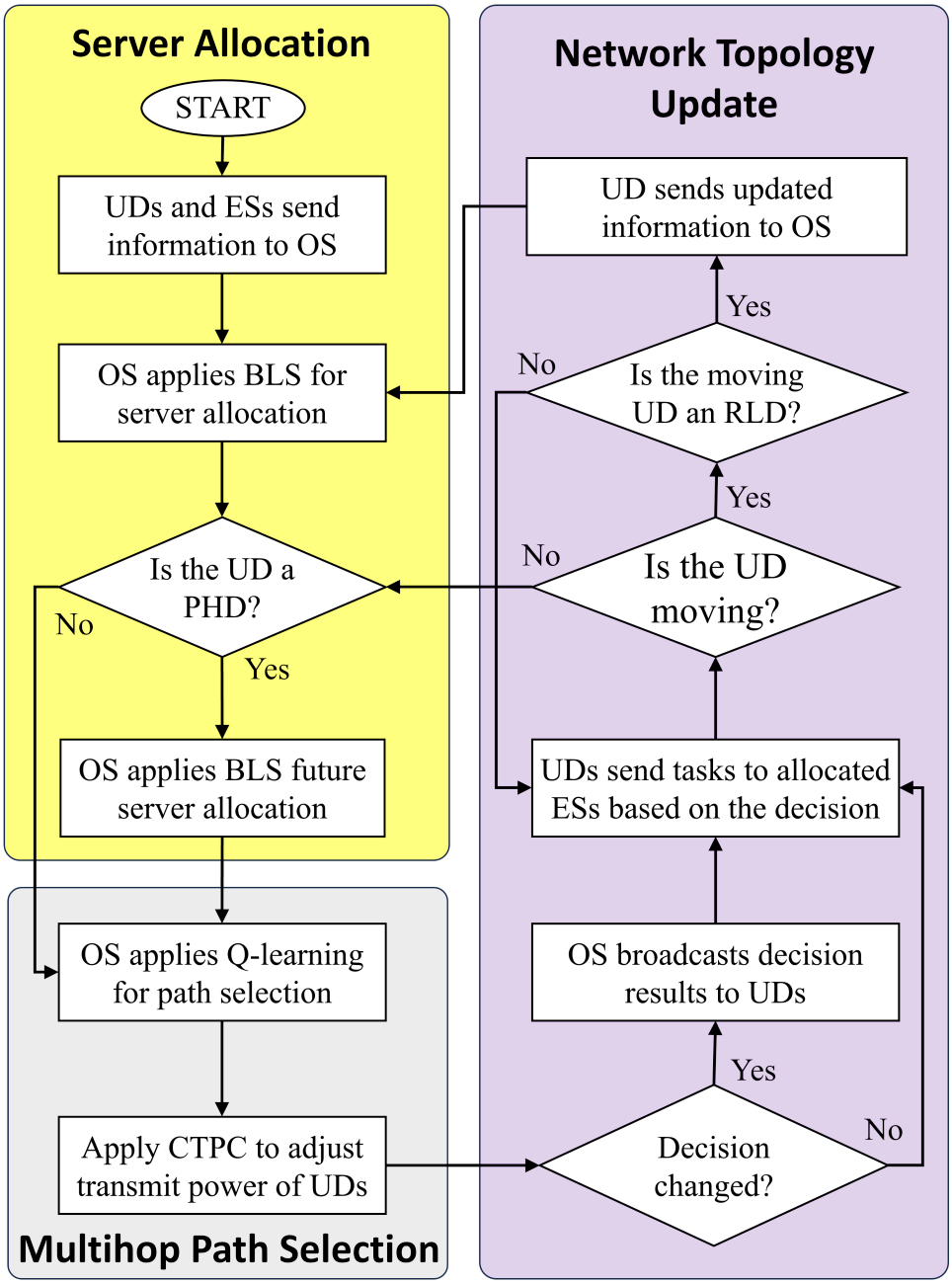


Figure 5.5: uPOW Flowchart.

ranking determined by the Random Forest algorithm based on the Gini impurity metric. This preprocessing ensures numerical stability and prevents overfitting.

To obtain supervised labels for BLS training, an oracle optimization problem is solved using the Gurobi optimizer [59] under various network conditions. The optimization aims to minimize overall delay or maximize communication quality while satisfying the network constraints defined in Section 4. The optimal ES assignment obtained from this process

serves as the label for each UD in the training dataset. After generating sufficient training samples, the dataset is used to train the BLS model following the standard procedure of constructing feature nodes and enhancement nodes, as defined in Equations (4.7)–(4.9). Once trained, the BLS model can infer the most suitable ES for each UD in real time.

One of the advantages of BLS is its incremental learning capability, which allows it to efficiently adapt to dynamic environments. When network conditions change, such as new devices joining or existing devices moving, the BLS model can be updated by adding a limited number of new features or enhancement nodes without retraining the entire model, maintaining both efficiency and accuracy.

5.4.2 Q-learning-based Path Selection

While BLS determines the optimal ES for each device, it does not specify the route for data transmission in multihop wireless networks. Since path selection is a sequential decision process affected by channel variations, interference, and mobility, Q-learning is introduced to dynamically determine the optimal multihop paths from UDs to their assigned ESs.

The multihop routing process is modeled as a Markov Decision Process (MDP), where the state represents the current transmitting node, the action corresponds to selecting a next-hop node, and the reward reflects the link quality between them, defined by the instantaneous SINR calculated from Equation (3.3). Each episode begins from a source UD and terminates once the packet reaches the ES selected by BLS. Through repeated interactions with the environment, the Q-learning agent learns to maximize the cumulative rewards, which correspond to high-quality, low-interference transmission paths.

To improve convergence and reduce exploration instability, the learning procedure is divided into two phases. In the first phase, SNR is used as the reward metric to construct a noise-resilient baseline topology. In the second phase, the SINR metric is employed to account for interference, enabling more precise optimization of multihop routes in dense networks. During training, the Q-table is iteratively updated as

$$Q^{new}(s_m, a_m) = (1-\alpha)Q(s_m, a_m) + \alpha\left(\tilde{r}_m + \gamma Q^{\max}(m+1)\right), \tag{5.19}$$

where α is the learning rate, γ is the discount factor, and $\tilde{r}_m = \max\{r_m, r_{m+1}\}$ represents the enhanced reward that considers both the current and next-hop link qualities. This one-

step look-ahead mechanism enables the learning process to favor decisions that maintain strong overall path performance rather than only immediate gains.

Unlike previous layer-based routing designs, the proposed Q-learning formulation removes the layer constraint during next-hop selection. That is, a node can flexibly select any neighboring relay within its transmission range, regardless of its hierarchical level or hop order. This relaxation allows the algorithm to explore a much broader action space and to discover unconventional yet efficient transmission routes. As a result, the learned routing policy exhibits greater adaptability to heterogeneous topologies and varying node densities.

After convergence, each device refers to its learned Q-table to select relay nodes that maximize the Q-values within its transmission range. If the obtained path satisfies the latency constraint defined in Equation (4.2), it is adopted as the optimal route. Otherwise, the Q-table continues to update until the constraint is met. In cases where multihop routes fail to meet the delay requirement, a direct single-hop link to the ES is used as a fallback.

To cope with device mobility, the Q-tables are dynamically updated based on recent network states. Devices with predictable movement patterns update their routing tables locally using precomputed Q-values, while devices with random mobility rely on the OS to periodically recalculate their routes. This adaptive mechanism ensures that the routing remains robust and efficient even in highly dynamic network environments.

In summary, BLS provides efficient server allocation based on global network features, while Q-learning enables adaptive routing through real-time reinforcement learning. Together, they achieve low latency and high capacity in the proposed multi-server wireless network.

5.4.3 Consensus Transmit Power Control

In the final stage of uPOW, CTPC operates on the transmit power variables $\{P_i\}$ of all UDs while keeping the server associations $b_{u,e}(t)$ and the path-selection indicators $y_{i,j}(t)$ fixed. The objective of CTPC is to jointly reduce the total interference power and the overall energy consumption, while maintaining the E2E throughput achieved by the previous two stages within an acceptable deviation range. To this end, CTPC iteratively scales each P_i by a consensus coefficient μ that is chosen to minimize a composite cost function

combining power-reduction gain and throughput preservation. As a result, CTPC explicitly optimizes the physical-layer power levels in a coordinated manner, complementing the upper-layer decisions on server allocation and route selection.

Transmit power control (TPC) strategies focus on improving individual link performance by adjusting each node's power based on local channel conditions or link requirements. For instance, nodes may increase power to overcome channel fading or decrease it when close to the receiver to conserve energy. However, such independent adaptations tend to produce uncoordinated power levels, which can unintentionally escalate interference across the network, especially in dense multihop environments. Moreover, purely local optimization cannot guarantee overall network stability or fairness, as high-power nodes may dominate the medium, degrading the quality of neighboring links.

To overcome these limitations, power control must be treated as a network-wide coordination problem rather than a per-link adjustment. Therefore, we introduce a consensus-based framework that aligns local power control decisions across all nodes to achieve global balance between throughput enhancement and interference reduction.

To mitigate interference and improve overall network efficiency, the proposed scheme employs a CTPC algorithm [89]. CTPC dynamically optimizes transmit powers through iterative consensus updates, ensuring that local throughput improvements do not cause disproportionate interference increases in the network. While improving individual throughput often requires increasing transmit power, such local gains can unintentionally cause global interference escalation, degrading the performance of nearby links. The core idea of the CTPC algorithm is to coordinate the transmit power decisions of all nodes via consensus-based optimization, ensuring that local throughput improvements do not lead to disproportionate increases in global interference. The operation of the CTPC algorithm can be divided into three main stages.

Initial Network Simulation

Firstly, after determining multihop paths using the enhanced Q-learning, an initial simulation is conducted with uniform default transmit powers at all nodes. This step yields baseline network metrics, including the average throughput and the average transmit power, which serve as benchmarks for subsequent optimization.

Weight Factors and Normalized Cost Function

To simultaneously consider throughput gain (TG) and transmit power reduction gain (PG), TG is defined as the ratio of the average throughput after applying CTPC (R_{CTPC}^{avg}) to the average throughput:

$$TG = \frac{R_{CTPC}^{avg}}{R^{avg}}. \tag{5.20}$$

PG is defined as the ratio of the average transmit power (P^{avg}) to the average transmit power after applying CTPC (P_{CTPC}^{avg}):

$$PG = \frac{P^{avg}}{P_{CTPC}^{avg}}. \tag{5.21}$$

These gains are normalized using two weight factors:

$$WT = \frac{PG}{TG + PG}, \quad WP = 1 - WT. \tag{5.22}$$

Subsequently, a normalized cost function is constructed to balance both metrics:

$$Cost = WT \cdot TG + WP \cdot PG. \tag{5.23}$$

This cost function facilitates finding an optimal balance between throughput enhancement and power saving.

Optimal Consensus Coefficient via Binary Search

To determine the best power control strategy, CTPC employs a binary search for the optimal consensus coefficient, denoted as μ. A search interval for μ is first established. For each candidate value of μ, the node transmit power is adjusted by

$$P_u^{new} = P_u \cdot \mu, \tag{5.24}$$

where P_u is the initial power of u, and P_u^{new} is the updated power after applying the consensus scaling. This approach allows nodes with higher-than-average throughput to reduce their power more significantly, thus minimizing their interference with other nodes. Conversely, nodes with below-average throughput retain or slightly increase their transmit power to ensure reliable communication and throughput.

For each candidate μ, a normalized cost function, which balances throughput gain and power saving, is evaluated. The binary search process iteratively explores the search space

and records the cost for each candidate value. This optimization is achieved through an iterative binary search process. In each iteration, two candidate consensus coefficients are selected within the current search interval. For each candidate, a complete network simulation is performed using the corresponding adjusted transmit powers, where each node's power is scaled by the current consensus coefficient.

The resulting network throughput and power consumption are then used to evaluate the normalized cost function. Based on the comparison of the two cost values, the search interval is updated by discarding the suboptimal half, and the next pair of candidates is selected. This process continues iteratively until the difference between the upper and lower bounds of the interval falls below a predefined tolerance threshold. In this way, the algorithm effectively searches for the optimal consensus coefficient μ^* that maximizes the cost function.

As a result, CTPC dynamically assigns node-specific transmit power levels in accordance with each node's relative performance, thereby achieving network-wide interference mitigation and performance enhancement.

5.5 Numerical Simulations

In this section, the performance of the proposed uPOW scheme is evaluated through extensive numerical simulations. We first describe the simulation scenarios, parameter settings, and performance metrics used in the evaluation. Then, we compare uPOW with other schemes under different network conditions, and analyze the results in terms of capacity, latency, interference, energy efficiency, and QoS.

5.5.1 Simulation Scenarios and Settings

To evaluate network performance with more precision, we redefine task completion time (θ) and network capacity (C). Specifically, θ is defined as the average of $t_{u,e}^{ser}$ among all UDs, which can be expressed as:

$$\theta = \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} t_{u,e}^{ser}. \tag{5.25}$$

Correspondingly, the network capacity C is defined as the ratio of the total amount of transmitted data to the task completion time. This is mathematically represented by:

$$C = \frac{\sum_{u \in \mathcal{U}} (L_u + L_u^{res})}{\theta}. \tag{5.26}$$

To evaluate the interference mitigation capability of the proposed scheme, we introduce the total interference power metric. The total interference power (P^{int}) is defined as the cumulative interference power experienced by all receiving nodes in the network during data transmissions, i.e.,

$$P^{int} = \sum_{u \in \mathcal{U}} \sum_{k \in \mathcal{U}, k \neq u} G_{k,u} \cdot P_k, \tag{5.27}$$

where $G_{k,u}$ denotes the channel power gain from the interfering device k to the receiving device u, and P_k is the transmission power of the interfering device k. This metric effectively quantifies the level of interference present within the network, thus clearly reflecting the interference mitigation capability of our proposed method.

To further assess energy efficiency, we define the total energy consumption metric in terms of Joules per bit (J/bit), which reflects the amount of energy consumed per successfully transmitted bit. For each wireless link during multihop communication, the energy consumption is computed by first calculating the transmission time as the ratio of the transmitted packet length to the corresponding link transmission rate. This transmission time is then multiplied by the transmit power used by the sending device to obtain the consumed energy (in Joules). The energy is finally normalized by dividing it by the transmitted packet length to yield the energy consumption per bit. Formally, the total energy consumption O is computed as

$$O = \frac{\sum_{i \in \mathcal{U} \cup \mathcal{E}} P_i}{\sum_{i \in \mathcal{U} \cup \mathcal{E}} \sum_{j \in \mathcal{U} \cup \mathcal{E}} R_{i,j}} \quad \text{(J/bit)}, \tag{5.28}$$

where P_i is the transmission power of node i, $L_{i,j}$ is the packet length, and $R_{i,j}$ is the transmission rate over the link between i and j. The total energy consumption is then calculated as the sum of energy per bit across all successfully transmitted links. This metric allows a fair comparison of energy efficiency between different algorithms and network settings.

In order to comprehensively evaluate the service quality of the proposed algorithms, we introduce a QoS metric (q), which measures the proportion of UDs simultaneously satisfying both latency constraints defined in the ES assignment and path selection phases (constraints **C2** and **C4**), respectively. Specifically, QoS represents the ratio of UDs

meeting the two conditions simultaneously. Thus, the QoS is defined as

$$q = \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} \mathbb{I}(u \text{ satisfies } \mathbf{C2} \text{ and } \mathbf{C4}), \tag{5.29}$$

where $\mathbb{I}(\cdot)$ is an indicator function that equals 1 when the condition in parentheses is satisfied and 0 otherwise. This metric intuitively captures the algorithm's ability to meet latency requirements simultaneously in both ES assignment and path selection phases, thereby offering a comprehensive measure of network service quality.

To evaluate the performance of uPOW, we carried out extensive simulations on an Apple Mac mini (2018) with Intel Core i7 3.2 GHz and 64GB DDR4 RAM. The emulated network covers a 500 m $\times$ 500 m area where 100–140 UDs move randomly with a maximum displacement of 0.5 m/s. Between 5 and 9 ESs are uniformly deployed; each UD provides a maximum transmission range of 80 m. Tasks arrive at UDs with sizes uniformly distributed between 10 MB and 30 MB, and the corresponding result size is assumed to be 5% of the original task. These simulation settings are designed to closely resemble real-world MEC scenarios involving heterogeneous device deployments, realistic movement patterns, and dynamic task generation. All simulation parameters are summarized in Table 5.1.

Currently, our model assumes a uniform unified communication protocol, simplifying analysis and simulation complexity. However, practical wireless networks are inherently heterogeneous. Devices typically operate on different frequencies or bandwidths due to hardware constraints or licensing. Our proposed uPOW scheme can be naturally extended to handle such heterogeneity by incorporating these hardware attributes into the BLS input features. By retraining the BLS model with heterogeneous parameters in the training dataset (generated through offline optimization using Gurobi), the server selection decisions can directly consider these non-uniform resource constraints. Additionally, Q-learning and CTPC support heterogeneous networks, as they already rely on node-specific SINR measurements and transmit power adjustments. Therefore, our scheme maintains significant flexibility and scalability for practical deployment in resource-heterogeneous MEC scenarios.

5.5.2 Simulation Results and Performance Analysis

In this section, we evaluate the performance of the proposed scheme through extensive simulations. To demonstrate the effectiveness of our approach, we compare our proposed

Table 5.1: Simulation parameters and settings of uPOW.

Parameter	**Value**
Network coverage area	500 m × 500 m
Number of UDs	50 ∼ 250
Number of ESs	5 ∼ 9
Ratio of PHD to RLD	70% : 30%
Channel bandwidth	40 MHz
Transmission power	200 mW
Attention constant	3.5
Shadowing attenuation	4 dB
Decorrelation distance	1 m
Noise level	-174 dBm/Hz
Task size	10 MB ∼30 MB
Result data size	5% of original task size
Maximum transmission range	80 m
Maximum UD movement per second	0.5 m
Weighting factor	0.9
No. of feature nodes for BLS	160
No. of enhancement nodes for BLS	1000
No. of adding enhancement nodes for BLS	50
Regularization coefficient for BLS	2×10^{-10}
Shrink coefficient for BLS	0.9
No. of incremental steps for BLS	5
Percentage of training data for BLS	90 %
Learning rate for Q-learning	0.5
Discount factor for Q-learning	0.9
Maximum iterations for Q-learning	200
Threshold for Q-learning	1 kbps
Transmission power adjustment Range	0.5 ∼ 1.0
Search tolerance	0.01
Baseline transmission power	200 mW

uPOW scheme with the Greedy Search (GS) and the existing reinforcement learning-based multihop relaying algorithm, referred to as QBMR [38]. The GS algorithm employs a greedy strategy for server allocation by selecting the server with the minimum $t_{u,e}^{est}$. In the selection phase, GS adopts an SNR-based greedy algorithm for multihop path selection, where each hop greedily selects the neighbor node that maximizes the immediate reward. In contrast, our uPOW approach integrates both intelligent server allocation and optimized routing strategies to enhance network performance comprehensively.

Simulation results are analyzed based on several key performance metrics, including network capacity, average throughput, QoS, task completion time, total energy consumption, and total interference power. To provide a comprehensive evaluation of the proposed algorithms, we consider three scenarios:

- The number of ESs is fixed, while the number of UDs varies, in order to evaluate the scalability of the proposed scheme with respect to user density.
- The number of UDs is fixed, while the number of ESs varies, to investigate the impact of ES deployment density on network performance.
- The UDs are mobile, and the impact of periodic location updates is evaluated to assess the robustness of the proposed algorithms in dynamic network environments.

Each scenario is simulated 500 times, and the results are presented as the average of all simulation runs to achieve statistical reliability.

We first evaluate the impact of varying the number of UDs on key network performance metrics while fixing the number of ESs to 5.

Figure 5.6 shows that all three algorithms exhibit a clear increasing trend in network capacity as the number of UDs grows from 50 to 250, which can be attributed to the more effective utilization of available server and relay resources under denser network conditions. Among the three schemes, the proposed uPOW consistently and significantly outperforms both GS and QBMR across all network scales. Specifically, as the number of user devices increases from 50 to 250, the network capacity achieved by uPOW rises from approximately 6.20 Gbps to 16.55 Gbps. In comparison, the capacity of the QBMR algorithm increases from 4.25 Gbps to 13.80 Gbps, while GS shows a more limited improvement, ranging only from 2.95 Gbps to 9.90 Gbps. On average, uPOW achieves about a

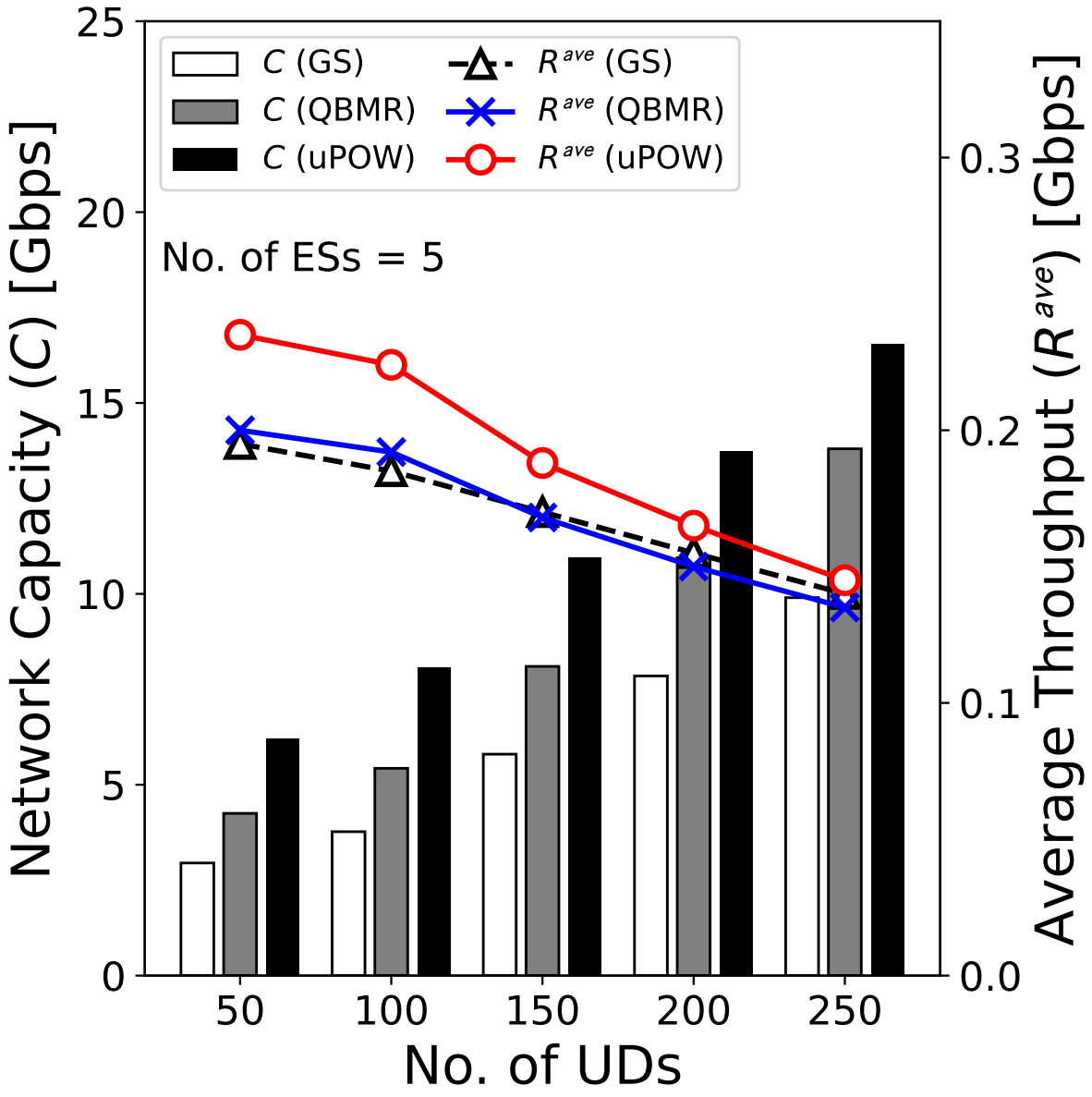


Figure 5.6: Network capacity and average throughput with varying no. of UDs.

25%–30% higher network capacity than QBMR and approximately 65%–75% higher than GS. These substantial performance gains by uPOW can be attributed to its integrated approach combining intelligent server allocation with Q-learning-based routing optimization and transmission power control, which effectively reduces network interference and enhances resource utilization.

As for the average throughput, it can be observed that all three methods exhibit a decreasing trend as the number of UDs increases from 50 to 250. This behavior is expected, since a larger number of devices intensifies resource contention and congestion when shar-

ing a fixed amount of server, thereby reducing the throughput per device. Nevertheless, even under such challenging conditions, the proposed uPOW scheme consistently maintains a higher average throughput than both GS and QBMR across all network scales. Specifically, as the number of UDs increases from 50 to 250, the average throughput of uPOW decreases moderately from approximately 235 Mbps to 145 Mbps. In comparison, the throughput achieved by QBMR drops from 200 Mbps to 135 Mbps, while GS shows a similar but more pronounced degradation, decreasing from 195 Mbps to 140 Mbps.

Overall, uPOW achieves an average throughput approximately 8%–18% higher than QBMR and 4%–21% higher than GS across the evaluated scenarios. This improved performance clearly highlights the effectiveness of the uPOW scheme in alleviating network congestion through optimized resource allocation and efficient multihop path selection.

As shown in Figure 5.7, we observe a general upward trend in total interference power for all three algorithms with increasing UDs. This trend is expected, as higher device density typically intensifies wireless interference. The proposed uPOW algorithm consistently achieves the lowest total interference power across all scenarios. Specifically, as the number of user devices increases from 50 to 250, the total interference power of uPOW rises moderately from approximately -5.23 dBm to 5.42 dBm. In comparison, the interference power observed under QBMR increases from -4.45 dBm to 6.30 dBm, while GS exhibits the most severe interference growth, ranging from -4.10 dBm to 7.50 dBm. Overall, uPOW consistently maintains a lower interference level across all network scales, achieving an interference reduction of approximately 1–2 dBm compared to both GS and QBMR. These outcomes highlight uPOW's effectiveness in dynamically controlling transmission power and selecting optimal routing paths, leading to substantial interference mitigation, even under dense network conditions.

As for the total energy consumption, all three algorithms show a clear increase in energy consumption as the number of user devices grows. This increase is a direct result of the additional transmission and computational resources required for handling more tasks. Nonetheless, our proposed uPOW algorithm consistently exhibits the lowest energy consumption among the three approaches. Specifically, as the number of UDs increases from 50 to 250, the total energy consumption of uPOW rises from approximately 0.12 nJ/bit to 0.70 nJ/bit. In comparison, the energy consumption under QBMR in-

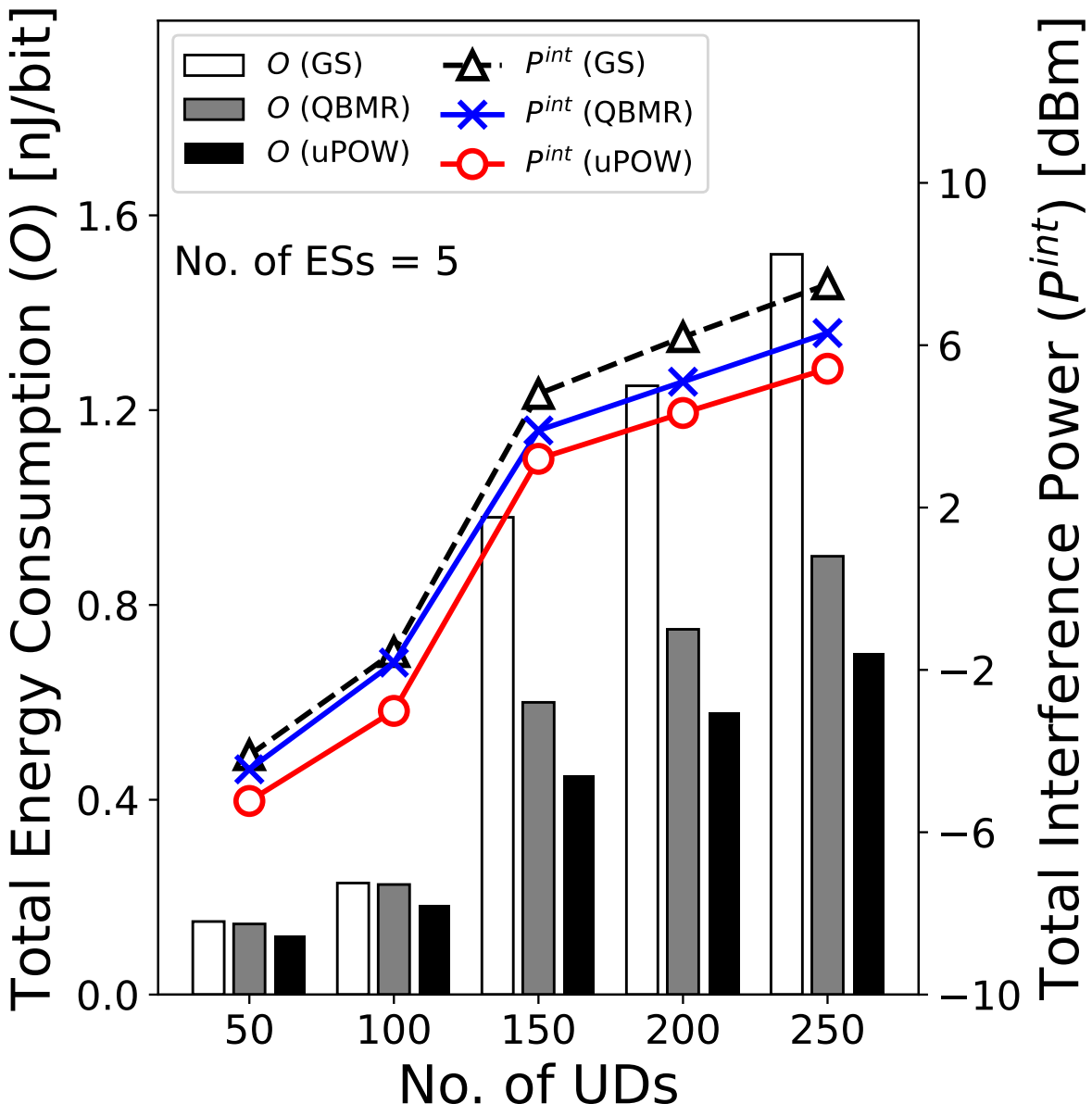


Figure 5.7: Interference power and energy consumption with varying no. of UDs.

creases from 0.15 nJ/bit to 0.90 nJ/bit, while GS exhibits the steepest growth, ranging from 0.15 nJ/bit to 1.52 nJ/bit. Overall, uPOW achieves approximately 20%–25% lower energy consumption compared to QBMR and around 40%–55% lower than GS across the evaluated scenarios. The superior energy efficiency of uPOW can be attributed to its integrated optimization scheme that incorporates transmission power control and efficient path selection, thereby substantially reducing unnecessary energy expenditure.

Figure 5.8 illustrates that the task completion time increases gradually for all algorithms when the number of UDs grows. This is expected since additional UDs introduce

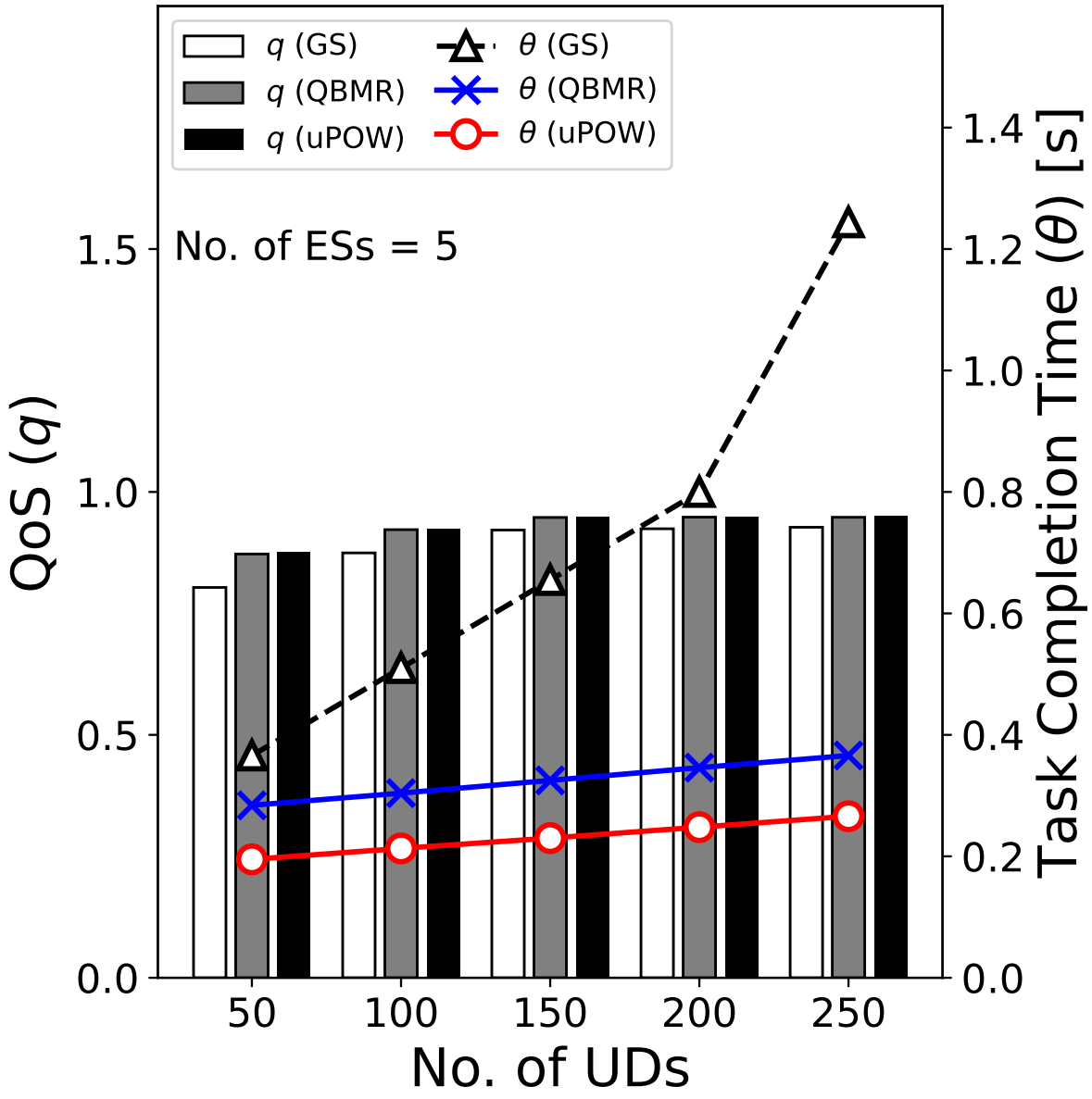


Figure 5.8: Task completion time and QoS performance with varying no. of UDs.

higher competition for computing and communication resources, consequently increasing delays. However, uPOW consistently demonstrates significantly lower task completion times compared to both GS and QBMR. Specifically, as the number of user devices increases from 50 to 250, the task completion time of uPOW increases slightly from approximately 0.20 s to 0.27 s, while maintaining a consistently low latency level across different network scales. In comparison, the task completion time of the QBMR algorithm increases from 0.28 s to 0.37 s, whereas GS exhibits the most severe delay growth, sharply rising from 0.37 s to 1.24 s as the network becomes denser. Overall, uPOW achieves approxi-

mately 27%–31% shorter task completion time than QBMR and around 47%–79% lower than GS across the evaluated scenarios. These improvements indicate that uPOW effectively integrates optimized server allocation and efficient multihop routing, significantly reducing delays in task processing and communication.

According to the results, QoS increases slightly for all methods with an increasing number of UDs. Notably, uPOW consistently achieves the highest QoS among the three schemes. As the number of UDs increases from 50 to 250, the QoS of uPOW improves steadily from approximately 87.56% to 94.94%. In comparison, QBMR increases from 87.23% to 94.79%, while GS shows a more pronounced improvement, rising from 80.34% to 92.73%. Although the performance gap between uPOW and QBMR remains relatively small (consistently below 1%), uPOW maintains a slight yet stable advantage across all network scales. Compared to GS, uPOW demonstrates a clear performance gain, achieving approximately 2%–7% higher QoS. These results reflect that uPOW's enhanced resource allocation and routing mechanisms effectively support higher QoS guarantees in dense network scenarios.

To further investigate the impact of ES deployment density on network performance, we evaluate the key performance metrics by varying the number of ESs from 5 to 9 while keeping the number of UDs fixed at 100.

As shown in Figure 5.9, we first evaluate how the network capacity and average throughput performance vary as the number of ESs increases. From the simulation results, it becomes clear that the network capacity consistently increases for all three algorithms. This is because, as more ESs provide additional computational resources, UDs can find closer or better ESs, thus enhancing overall network capacity. Specifically, uPOW's network capacity increases from approximately 7.64 Gbps (5 ESs) to 8.39 Gbps (9 ESs). In comparison, QBMR improves moderately from 5.47 Gbps to 6.11 Gbps, whereas GS shows the lowest performance increase, from 3.70 Gbps to 4.95 Gbps. A similar trend is observed in terms of average throughput. All three algorithms experience improved average throughput as the number of ESs increases. uPOW again demonstrates a significant advantage, rising from approximately 222.40 Mbps (5 ESs) to 269.87 Mbps (9 ESs). QBMR and GS increase from 192.76 Mbps to 237.43 Mbps and from 177.12 Mbps to 216.07 Mbps, respectively. Overall, uPOW achieves roughly 10%-20% higher throughput

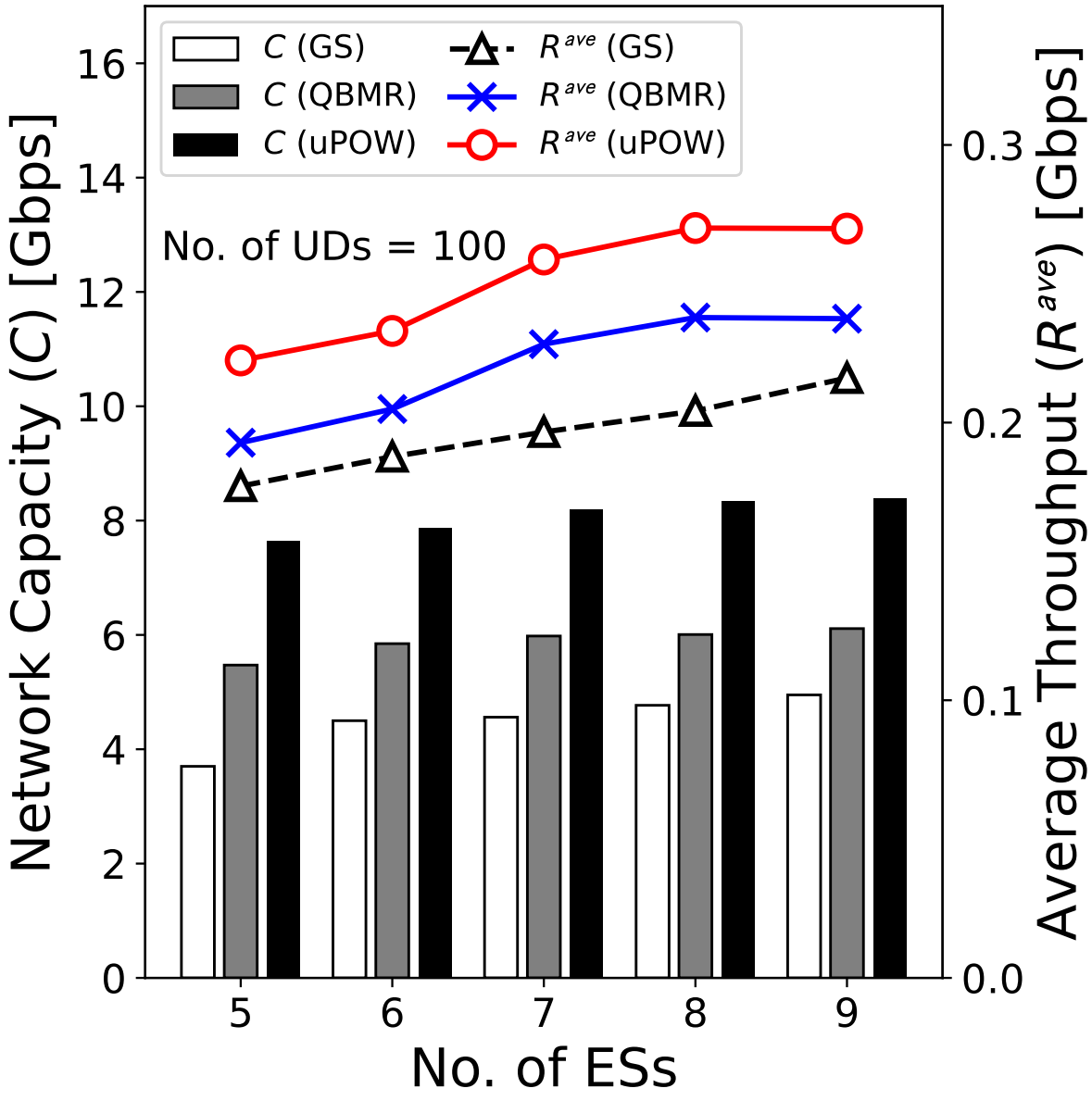


Figure 5.9: Network capacity and average throughput with varying no. of ESs.

than QBMR and around 20%-30 higher throughput compared to GS. These results underline that uPOW effectively leverages the additional ESs to maximize resource utilization and improve overall network performance.

As shown in Figure 5.10, for total interference power, an evident decreasing trend is observed for all three algorithms with an increasing number of ESs, indicating that deploying additional ESs effectively alleviates network interference. uPOW consistently achieves the lowest interference power, decreasing sharply from -3.21 dBm (5 ESs) to -5.16 dBm (9 ESs), reflecting substantial interference mitigation through optimized routing

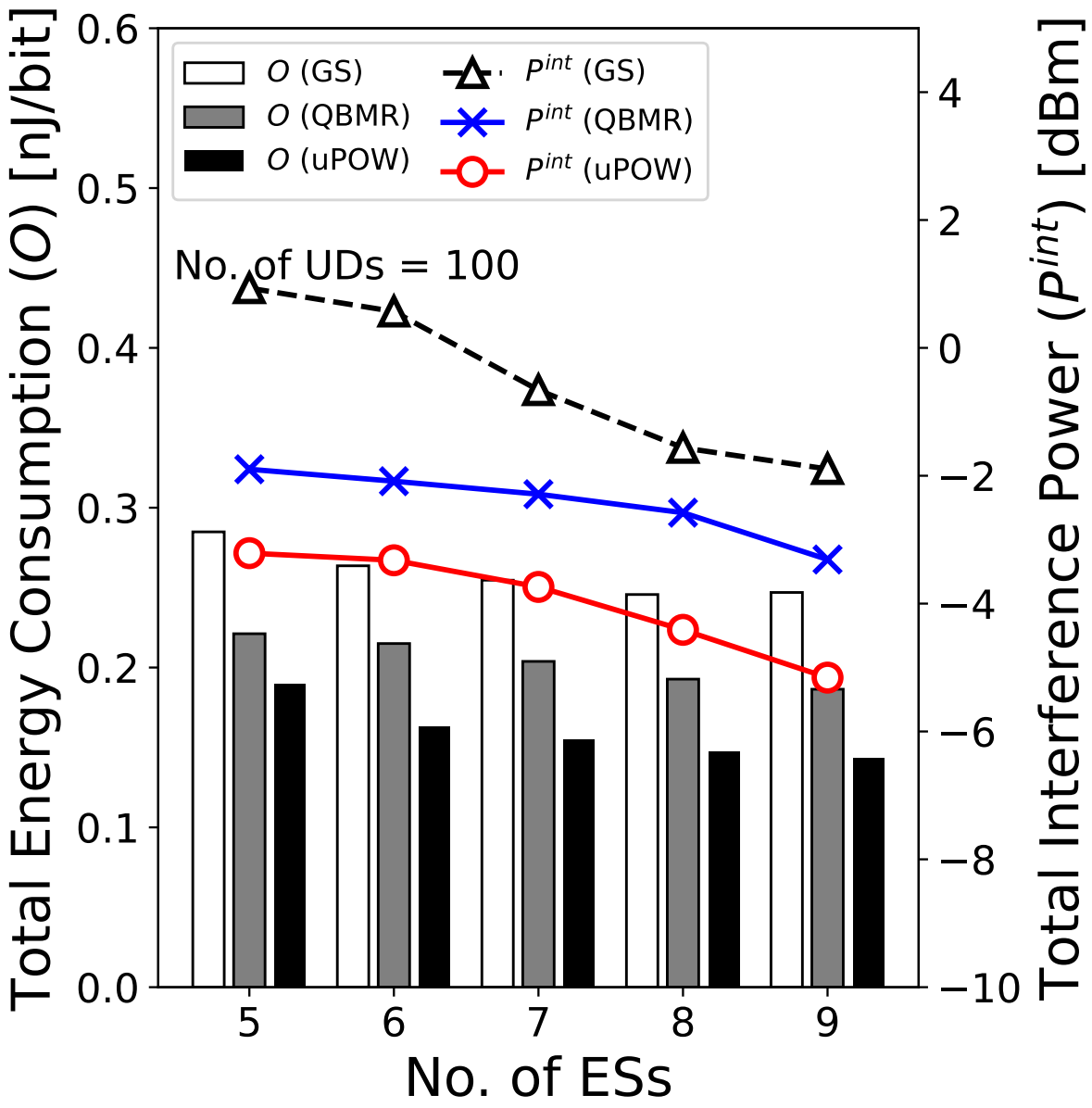


Figure 5.10: Interference power and energy consumption with varying no. of ESs.

and transmission power control. QBMR also achieves interference reduction, improving from -1.90 dBm to -3.31 dBm, while GS starts at positive interference (0.93 dBm) and only reduces to -1.89 dBm, highlighting its limited interference management capability. Similarly, energy consumption consistently decreases for all three algorithms as more edge servers are deployed, due to reduced transmission distances and improved resource allocation efficiency. uPOW demonstrates the lowest energy consumption, declining from 0.190 nJ/bit (5 ESs) to 0.143 nJ/bit (9 ESs). QBMR exhibits moderate improvement from 0.221 nJ/bit to 0.186 nJ/bit, whereas GS achieves the smallest reduction from 0.285

nJ/bit to approximately 0.247 nJ/bit. Overall, uPOW consumes around 20%-30% less energy compared to QBMR and roughly 40%-50% less compared to GS, emphasizing its superior energy efficiency.

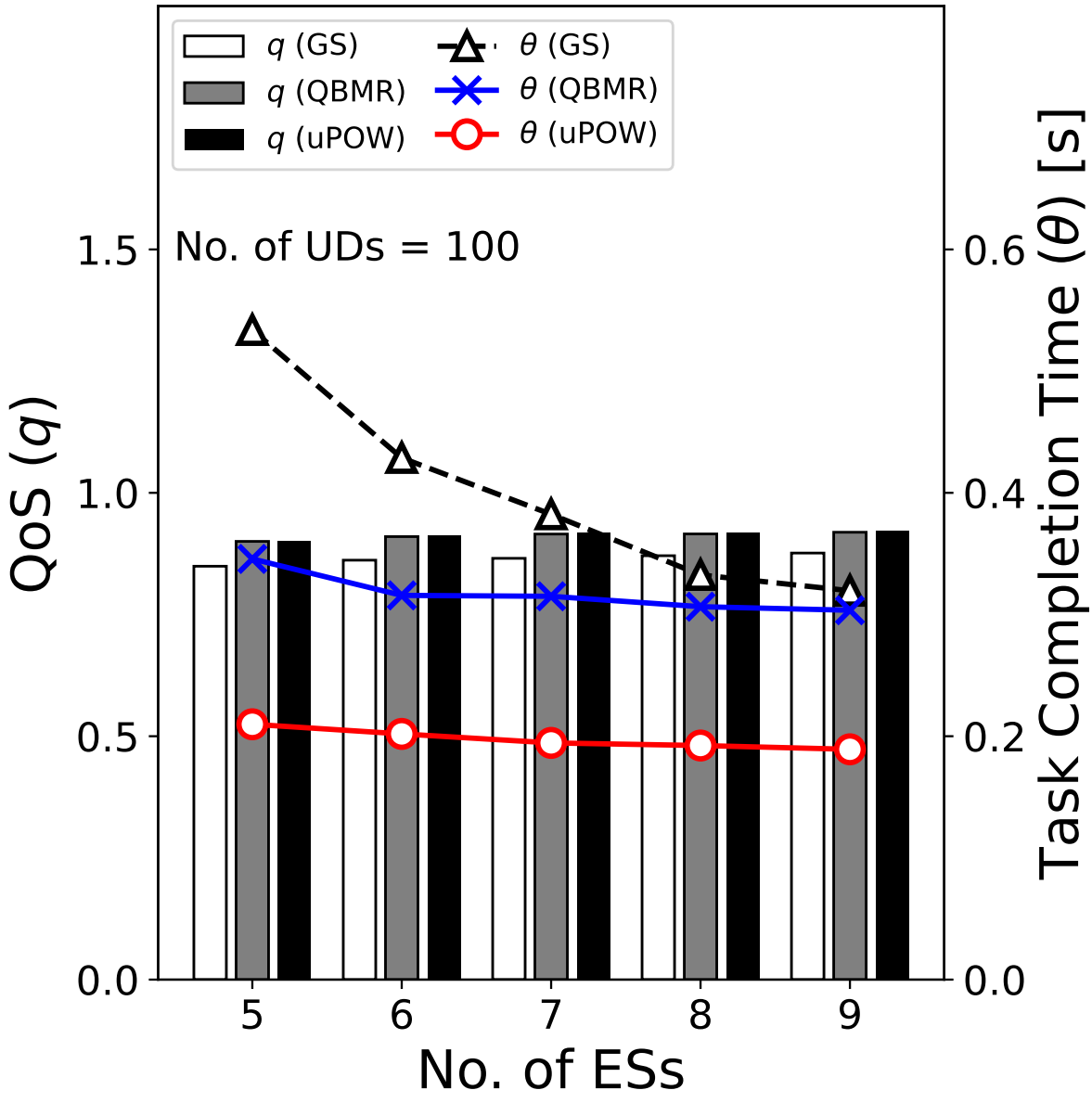


Figure 5.11: Task completion time and QoS performance with varying no. of ESs.

Figure 5.11 presents the QoS and task completion time performance. The QoS for all algorithms gradually improves, demonstrating the benefit of added computational resources in reducing latency and improving task completion reliability. Among the evaluated algorithms, uPOW consistently delivers the highest QoS. Specifically, uPOW achieves QoS improvements from approximately 90.07% (5 ESs) to 92.08% (9 ESs), while QBMR

slightly lags behind, ranging from 90.02% to 91.90%. GS exhibits the lowest QoS, improving from 84.92% to 87.63%. These results clearly demonstrate that uPOW provides superior quality of service guarantees compared to the baseline methods. Regarding task completion time, all three algorithms exhibit a clear reduction. uPOW consistently demonstrates the lowest task completion time across all scenarios, decreasing notably from approximately 0.210 seconds (5 ESs) to 0.189 seconds (9 ESs). QBMR's completion time decreases moderately from 0.346 seconds to 0.304 seconds, whereas GS's time declines sharply from 0.534 seconds to 0.320 seconds but remains the highest among all methods. Thus, uPOW reduces task completion time significantly—by approximately 40% compared to QBMR and by over 50% compared to GS—demonstrating its robust efficiency in task execution.

In the third scenario, we focus on the impact of UD mobility on the network performance. We set the number of ESs to 5 and the number of UDs to 100. The time slots for updating location information vary from 0 to 25 seconds to investigate their influence on the performance metrics.

Figure 5.12 illustrates the impact of periodic location updates on network capacity and average throughput when UDs are mobile. As the time slot increases, the network capacity remains relatively stable across all algorithms with slight fluctuations due to mobility-induced dynamics. Specifically, the proposed uPOW maintains the highest capacity, ranging from approximately 7.64 to 7.72 Gbps, demonstrating robustness against varying mobility. QBMR achieves intermediate capacity, fluctuating between 5.43 and 5.50 Gbps, while GS shows the lowest capacity, consistently around 3.77 Gbps. These results suggest that the proposed uPOW effectively adapts to mobility-induced network changes, maintaining superior resource utilization and performance stability. Similarly, the average throughput exhibits minor fluctuations due to mobility, but the relative ranking among algorithms remains unchanged. uPOW consistently delivers the highest throughput, maintaining around 223–226 Mbps throughout the varying update intervals. QBMR achieves moderate throughput performance, staying around 193–197 Mbps, while GS remains the lowest, varying between 179–192 Mbps. The stable and superior throughput performance of uPOW highlights its effective management of mobility-related variations through intelligent server allocation and optimized path selection.

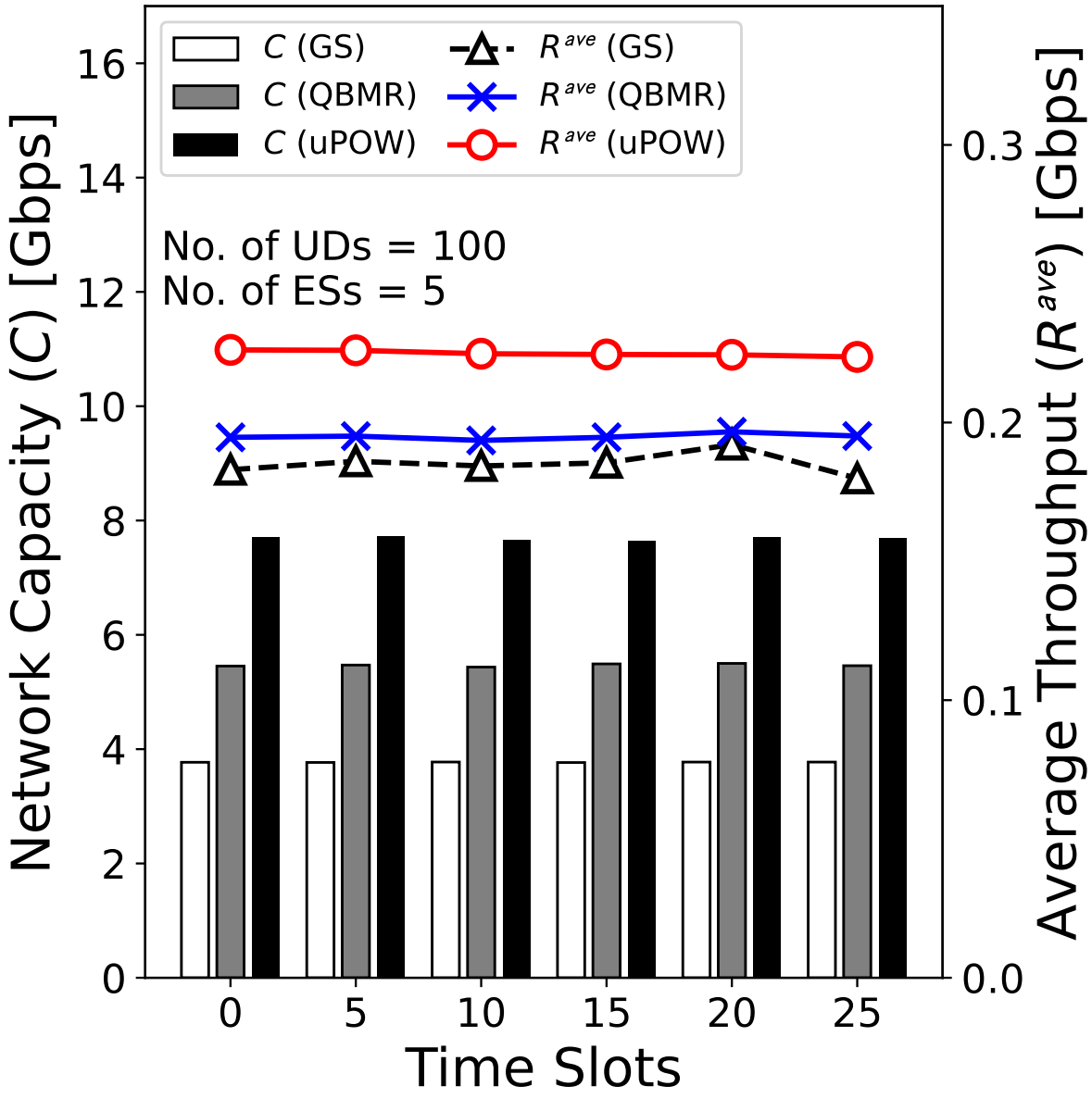


Figure 5.12: Network capacity and average throughput with periodic location updates.

Figure 5.13(b) shows the results of total interference power and total energy consumption. The total interference power fluctuates modestly for all three algorithms due to the dynamic wireless environment. However, uPOW achieves the lowest interference levels, ranging between approximately -1.99 dBm and -2.72 dBm. QBMR experiences higher interference fluctuations from about -0.81 dBm to -1.65 dBm, whereas GS displays consistently higher interference power, ranging between 0.47 dBm and 0.61 dBm. These results confirm uPOW's capability to effectively manage and reduce interference through adaptive power control and optimized path selection strategies, even under mobility-induced

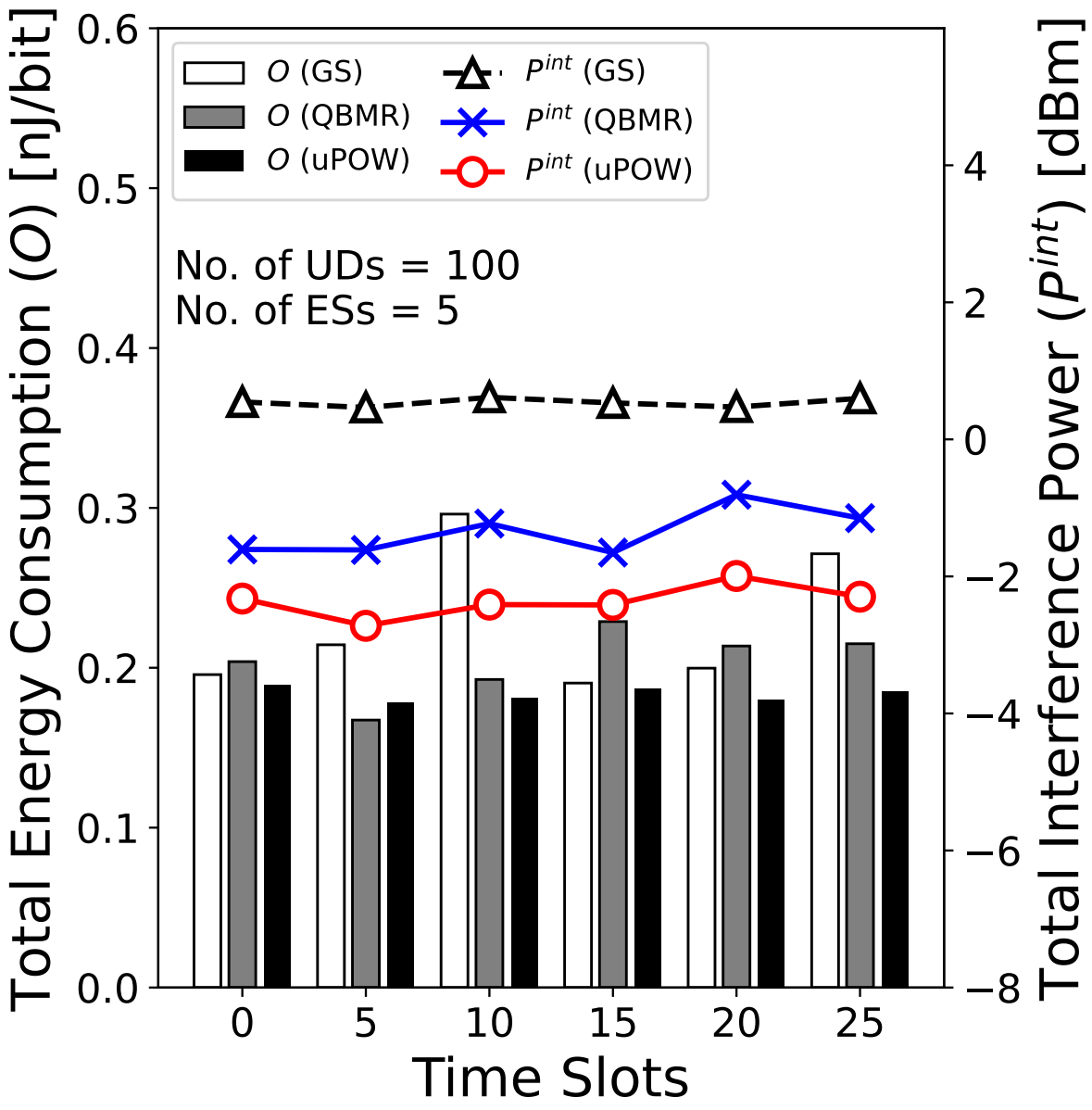


Figure 5.13: Interference Power and Energy Consumption with Periodic Location Updates

network variations. Regarding total energy consumption, measured in nJ/bit, the uPOW maintains stable and the lowest stable energy consumption across all scenarios, ranging between approximately 177.98 and 189.21 nJ/bit. QBMR and GS exhibit higher variability and consumption levels, with QBMR fluctuating between 167.30 to 228.90 nJ/bit and GS varying widely between 190.44 to 296.12 nJ/bit. The clear energy efficiency advantage of uPOW highlights its adaptive energy-aware optimization, effectively handling the extra energy demands typically associated with device mobility.

As shown in Figure 5.14, for the QoS, uPOW shows remarkable stability across differ-

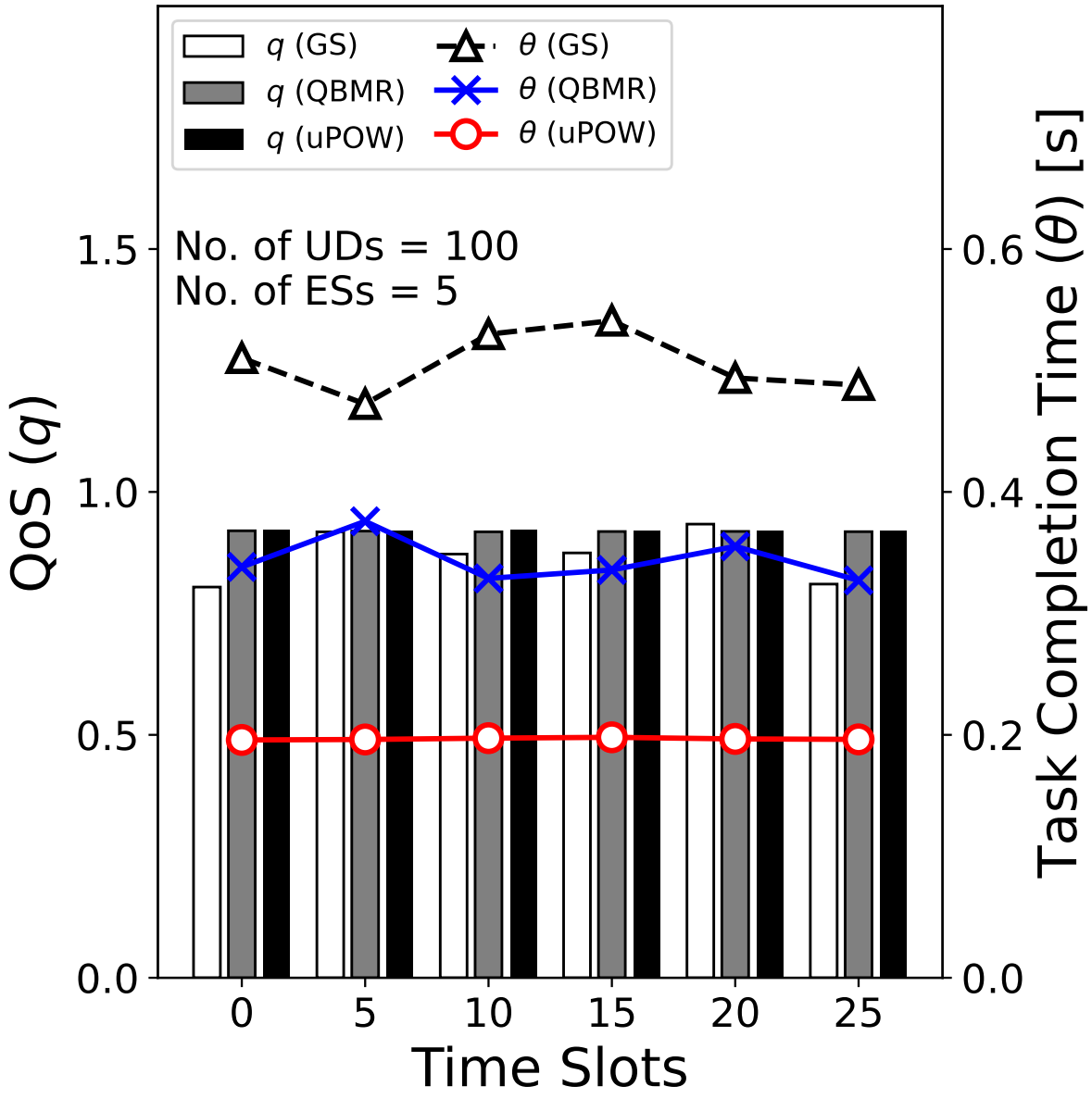


Figure 5.14: Task completion time and QoS with periodic location updates.

ent location update intervals, both consistently achieving around 92%. Specifically, uPOW remains stable within a narrow range (approximately 91.99–92.19%), closely matched by QBMR (approximately 91.79–92.00%). Conversely, GS exhibits significant variability, with QoS ranging from as low as 80.42% at 0 seconds up to 93.38% at 20 seconds. These findings reveal that uPOW effectively ensures reliable QoS in dynamic scenarios, demonstrating notable resilience to mobility-induced uncertainties. In terms of task completion time, uPOW is consistently better than other algorithms, maintaining a low and stable completion time around 0.196 seconds across all intervals. QBMR shows moderate vari-

ations between approximately 0.327 to 0.376 seconds, while GS fluctuates considerably between 0.472 to 0.541 seconds. The significantly lower and more stable task completion times achieved by uPOW underscore its robustness in handling device mobility, effectively mitigating potential performance degradation through optimized resource allocation and routing strategies.

Across different scenarios, including varying user density, ES deployment density, and user mobility, the proposed scheme consistently achieves higher network capacity, lower task completion time, better throughput performance, and higher QoS, and that is in comparison to GS and QBMR. These findings confirm that the proposed design is well-suited for large-scale dynamic networks and can efficiently support reliable and low-latency communications in future wireless systems.

5.6 Summary

This section proposes a novel three-stage optimization scheme named uPOW, designed to address the key challenges of scalability, interference, and energy efficiency in MWMNs. The proposed uPOW scheme integrates three intelligent components: a BLS for optimal server allocation, a SINR-based Q-learning algorithm for multihop path selection, and a CTPC algorithm for adaptive power adjustment. Extensive simulation results demonstrate that uPOW consistently outperforms existing benchmark algorithms, including GS and QBMR, in terms of various key performance indicators. The proposed method effectively enhances network capacity, reduces latency, improves QoS, and achieves more stable and higher throughput in dense and dynamic environments. The proposed uPOW scheme presents a scalable and flexible solution for next-generation wireless networks.

Chapter 6

Conclusion

This dissertation set out to address a central question for B5G WMNs: How can capacity, latency, and energy efficiency be jointly optimized in a dynamic wireless multihop environment? Approaching this question required not only the development of individual algorithms, but also the design of a coherent problem–solution trajectory in which each step addressed a specific challenge that emerged from the previous one.

6.1 Overall Discussion

The starting point of this dissertation was the observation that performance degradation in WMNs is not driven by a single factor but by a chain of interdependent performance limitations. Long multihop relaying paths cause latency accumulation; uneven task offloading across edge servers leads to capacity imbalance; and dense simultaneous transmissions amplify interference and energy consumption. Therefore, the goal of this dissertation was not to eliminate one problem in isolation, but to progressively identify and resolve the dominant limitation at each stage of network scaling.

The first stage of the research focused on the most critical performance limitation: insufficient network capacity caused by interference coupling along multihop routes. To address this issue, the eCAP scheme was introduced. By integrating FG-assisted topology formation, two-phase Q-learning routing, and CoF cooperative forwarding, eCAP successfully constructed high-throughput multihop transmission paths and significantly enhanced network capacity in WMNs. However, eCAP operates under a single-server assumption,

meaning that devices always relay toward the same root server. Thus, the scheme does not handle scenarios where many devices need to simultaneously access multiple edge servers, which becomes a dominant performance limitation as the network scales.

This insight motivated the second stage of the research and the development of the eLOW scheme, which aimed to reduce end-to-end service latency in MSWNs. The eLOW scheme formulated server selection as a learning-driven optimization problem and adopted the BLS as a fast and incrementally updatable decision engine for latency-sensitive resource allocation. While eLOW effectively balanced server workloads and suppressed queue buildup, it is designed under a single-hop assumption in which devices directly connect to edge servers without relying on multihop forwarding.

To overcome these limitations, the third stage of the research introduced the uPOW scheme. uPOW operates in MWMNs and jointly considers server allocation, path selection, and transmit power control. By integrating BLS-based server selection, SINR-driven Q-learning routing, and CTPC, uPOW forms a unified optimization cycle that supports cooperative and cross-dependent decision making. Simulation results demonstrated that uPOW consistently outperforms benchmark algorithms such as GS and QBMR in terms of network capacity, interference mitigation, energy efficiency, and QoS satisfaction—confirming that jointly optimizing server allocation, routing, and power control is necessary for scalable multihop communication.

Through this three-stage progression, the dissertation reveals a key conceptual insight: the dominant performance limitation of WMNs evolves as optimization progresses. This finding motivated the development of the CORE framework, which unifies eCAP, eLOW, and uPOW under a single adaptive architecture. Rather than applying a fixed optimization strategy, CORE coordinates the most appropriate scheme according to network scale, service demand, and traffic characteristics. This represents a transition from static algorithm execution toward adaptive and cooperative AI-native networking in future wireless systems.

6.2 Contributions

The major contributions of this dissertation are summarized below, following the chronological evolution of the research and the progressively shifting performance limitations

identified throughout analysis and simulation:

- **Identification of progressively shifting performance limitations in WMNs.** This dissertation reveals a three-phase performance degradation pattern in WMNs as system scale increases: (i) limited multihop throughput due to interference-coupled routing, (ii) excessive E2E latency caused by server load imbalance in multi-server environments, and (iii) interference and energy inefficiency under dense simultaneous transmissions. This finding transforms the optimization objective from isolated metrics to coordinated performance improvement.

- **Design of the eCAP scheme for throughput enhancement.** To overcome the challenge of low network capacity in WMNs, this dissertation proposes the eCAP scheme, which jointly exploits FG approach for root node selection, two-phase Q-learning for path selection, and CoF strategy for cooperative transmissions. eCAP significantly increases network capacity by forming interference-resilient multihop paths, confirming the potential of learning-driven cooperative forwarding in WMNs.

- **Development of the eLOW scheme for latency reduction in MSWNs.** When server access becomes the new primary constraint under multi-server deployment, the proposed eLOW scheme leverages the BLS for fast and adaptive server selection. The BLS/NS and BLS/MS variants reduce service latency by balancing load, enhancing server connectivity, and improving link quality, demonstrating that BLS is capable of accurate and scalable decision-making in dynamic MEC systems.

- **Formulation of the uPOW scheme for unified interference and energy management.** To address interference-driven capacity loss and power inefficiency in large-scale MWMNs, this work develops the uPOW scheme, which integrates BLS-based server allocation, SINR-driven Q-learning path selection, and CTPC-based transmit power coordination. This is the first unified learning-and-consensus-driven solution that simultaneously improves network capacity, energy efficiency, and QoS satisfaction under dense network conditions.

- **Integration of the three schemes into the CORE framework.** The eCAP, eLOW, and uPOW schemes are unified into the CORE framework, which follows

a progressive optimization pipeline capable of adapting to network operating conditions. CORE enables future AI-native wireless networks to cooperatively optimize throughput, delay, and power sustainability without modifying existing protocol structures, establishing a scalable foundation for B5G/6G autonomous network management.

6.3 Social Impact and Broader Implications

The research presented in this dissertation has potential social impact in several important domains related to future wireless communication systems.

First, the proposed cooperative AI-driven framework contributes to the reliability and scalability of WMNs, which are critical in disaster recovery and emergency communication scenarios. In such environments, communication infrastructure is often damaged or unavailable, and rapidly deployable multihop networks supported by edge computing can enable coordination among rescue teams, sensors, and command centers.

Second, by jointly optimizing routing, server selection, and transmit power, the proposed framework improves energy efficiency and reduces unnecessary interference. This contributes to more sustainable operation of large-scale wireless systems, which is increasingly important as the number of connected devices continues to grow in smart cities, industrial IoT, and environmental monitoring applications.

Third, the integration of edge computing and distributed intelligence supports low-latency and adaptive services for time-sensitive applications, such as industrial automation and public safety systems. By enabling efficient task execution under dynamic network conditions, the proposed framework helps bridge the gap between advanced communication technologies and practical societal needs.

Overall, this dissertation advances the design of intelligent and sustainable wireless communication frameworks, supporting the long-term development of robust digital infrastructure in the B5G and future 6G era.

6.4 Future Works

Although the outcomes of this dissertation are promising, several simplifying assumptions were adopted to maintain analytical feasibility. These include perfect CSI, synchronized time reference, unified medium access (TDMA), and the absence of signaling overhead for model updates. Moreover, hardware-level impairments and queuing delay were not considered. Addressing these factors will enable more realistic deployment conditions in future studies.

While the proposed CORE framework has demonstrated substantial potential, several research directions emerge naturally from the current findings:

- **Federated and distributed learning.** Incorporating federated learning and multi-agent reinforcement learning can enable privacy-preserving and communication-efficient distributed optimization, reducing reliance on centralized model aggregation.

- **Heterogeneous and cross-domain network environments.** Extending the CORE framework to heterogeneous deployments with mixed radio access technologies, dual-band links, diverse mobility levels, and satellite-terrestrial integration will greatly increase realism and applicability.

- **Joint orchestration of spectrum-computation-caching.** Future enhancements may incorporate adaptive spectrum sensing, dynamic bandwidth slicing, and caching-aware task placement to achieve E2E coordination across multiple network resource dimensions.

- **Real-world implementation and prototyping.** Evaluating the proposed schemes on SDR platforms or edge computing testbeds will provide practical insights into latency under synchronization errors, power-consumption behavior, and real-time learning convergence.

- **Green intelligence and sustainable communication.** Integrating energy-harvesting models, sleep scheduling for idle nodes, and carbon-aware learning objectives will further align CORE with carbon-neutral and environmentally sustainable 6G development goals.

In conclusion, this dissertation demonstrates that artificial intelligence does not replace communication theory; rather, it strengthens it. By unifying sensing, learning, and coordinated decision-making into a cooperative and modular architecture, the CORE framework moves wireless multihop networks closer to the long-term vision of fully autonomous, self-optimizing, and self-sustaining communication systems in the B5G era.

Bibliography

[1] Cisco Systems. Cisco annual internet report (2018–2023) white paper, 2020. Accessed: November 2025.

[2] Ericsson. Ericsson mobility report, november 2024, 2024. Accessed: November 2025.

[3] International Telecommunication Union (ITU-R). Recommendation ITU-R M.2160-0: Framework and overall objectives of the future development of IMT for 2030 and beyond. `https://www.itu.int/rec/R-REC-M.2160-0-202311-I/en`, 2023. Accessed: November 2025.

[4] Walid Saad, Mehdi Bennis, and Mingzhe Chen. A vision of 6g wireless systems: Applications, trends, technologies, and open research problems. *IEEE Network*, 34(3):134–142, 2020.

[5] Zhengquan Zhang, Yue Xiao, Zheng Ma, Ming Xiao, Zhiguo Ding, Xianfu Lei, George K. Karagiannidis, and Pingzhi Fan. 6g wireless networks: Vision, requirements, architecture, and key technologies. *IEEE Vehicular Technology Magazine*, 14(3):28–41, 2019.

[6] I.F. Akyildiz and Xudong Wang. A survey on wireless mesh networks. *IEEE Communications Magazine*, 43(9):S23–S30, 2005.

[7] Petar Popovski, Jimmy J. Nielsen, Cedomir Stefanovic, Elisabeth de Carvalho, Erik Strom, Kasper F. Trillingsgaard, Alexandru-Sabin Bana, Dong Min Kim, Radoslaw Kotaba, Jihong Park, and Rene B. Sorensen. Wireless access for ultra-reliable low-latency communication: Principles and building blocks. *IEEE Network*, 32(2):16–23, 2018.

[8] Jayavardhana Gubbi, Rajkumar Buyya, Slaven Marusic, and Marimuthu Palaniswami. Internet of things (iot): A vision, architectural elements, and future directions. *Future Generation Computer Systems*, 29(7):1645–1660, 2013. Including Special sections: Cyber-enabled Distributed Computing for Ubiquitous Cloud and Network Services and Cloud Computing and Scientific Applications — Big Data, Scalable Analytics, and Beyond.

[9] Tarik Taleb, Konstantinos Samdanis, Badr Mada, Hannu Flinck, Sunny Dutta, and Dario Sabella. On multi-access edge computing: A survey of the emerging 5g network edge cloud architecture and orchestration. *IEEE Communications Surveys and Tutorials*, 19(3):1657–1681, 2017.

[10] Zirui Chen, Zhaoyang Zhang, and Zhaohui Yang. Big ai models for 6g wireless networks: Opportunities, challenges, and research directions. *IEEE Wireless Communications*, 31(5):164–172, 2024.

[11] Abdulmalik Alwarafy, Mohamed Abdallah, Bekir Sait Ciftler, Ala Al-Fuqaha, and Mounir Hamdi. Deep reinforcement learning for radio resource allocation and management in next generation heterogeneous wireless networks: A survey. *IEEE Communications Surveys & Tutorials*, 2021. arXiv:2106.00574.

[12] Ahmed A. Ismail, Nour Eldeen Khalifa, and Reda A. El-Khoribi. A survey on resource scheduling approaches in multi-access edge computing environment: A deep reinforcement learning study. *Cluster Computing*, 28:184, 2025.

[13] Haoxiang Luo, Yu Yan, Yanhui Bian, Wenjiao Feng, Ruichen Zhang, Yinqiu Liu, Jiacheng Wang, Gang Sun, Dusit Niyato, Hongfang Yu, et al. Ai reasoning for wireless communications and networking: A survey and perspectives. *arXiv preprint arXiv:2509.09193*, 2025.

[14] J. Yuan, Z. Li, W. Yu, and B. Li. A cross-layer optimization framework for multihop multicast in wireless mesh networks. *IEEE Journal on Selected Areas in Communications*, 24(11):2092–2103, 2006.

[15] Long Le and Ekram Hossain. Cross-layer optimization frameworks for multihop wireless networks using cooperative diversity. *IEEE Transactions on Wireless Communications*, 7(7):2592–2602, 2008.

[16] Amit Kr. Jain, Rupesh Acharya, Saroj Jakhar, and Tarun Mishra. Fifth generation (5g) wireless technology "revolution in telecommunication". In *2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT)*, pages 1867–1872, 2018.

[17] Zhihan Cui, Aung Thura Phyo Khun, Yuto Lim, and Yasuo Tan. Factor graph-based deep reinforcement learning for path selection scheme in full-duplex wireless multihop networks. In *Int. Wireless Commun. and Mobile Comput. (IWCMC)*, pages 1220–1225, 2023.

[18] Mohammed H. Alsharif, Anabi Hilary Kelechi, Mahmoud A. Albreem, Shehzad Ashraf Chaudhry, M. Sultan Zia, and Sunghwan Kim. Sixth generation (6G) wireless networks: Vision, research activities, challenges and potential solutions. *Symmetry*, 12(4):1–5, 2020.

[19] Gautam Trivedi and Bijan Jabbari. On wireless link connectivity for resilient multihop networks. In *IEEE Mil. Commun. Conf. (MILCOM)*, pages 285–290, 2021.

[20] Chikara Fujimura, Kosuke Sanada, and Kazuo Mori. Throughput analysis for string-topology full-duplex multi-hop network. In *Int. Symp. on Wireless Pers. Multimedia Commun. (WPMC)*, pages 535–541, 2017.

[21] Shahbaz Rezaei, Mohammed Gharib, and Ali Movaghar. Throughput analysis of IEEE 802.11 multi-hop wireless networks with routing consideration: A general framework. *IEEE Trans. on Commun.*, 66(11):5430–5443, 2018.

[22] Won Jae Lee, Jin Ki Kim, Kyeong Rok Kim, and Jae Hyun Kim. Second receiver selection algorithm for fairness in full duplex communications. In *IEEE VTS Asia Pacific Wireless Commun. Symp. (APWCS)*, pages 1–5, 2019.

[23] Jinsong Gui and Kai Zhou. Flexible adjustments between energy and capacity for topology control in heterogeneous wireless multi-hop networks. *J. Netw. Syst. Manage.*, 24(4):789–812, 2016.

[24] Deniz Türsel Eliiyi, Hilal Arslan, Vahid Khalilpour Akram, and Onur Uğurlu. Parallel identification of central nodes in wireless multi-hop networks. In *Signal Process. and Commun. Appl. Conf. (SIU)*, pages 1–4, 2020.

[25] Rupei Xu, Yuming Jiang, and Jason P. Jue. Mist: An efficient approach for software-defined multicast in wireless mesh networks. In *2024 IEEE 35th International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, pages 1–6, 2024.

[26] Yongyi Mao, Frank Kschischang, Baochun Li, and Subbarayan Pasupathy. A factor graph approach to link loss monitoring in wireless sensor networks. *IEEE J. on Sel. Areas in Commun.*, 23(4):820–829, 2005.

[27] Wei Li, Zhen Yang, and Haifeng Hu. Distributed multi-sensor tracking in wireless networks using nonparametric variant of sum-product algorithm. In *Asia-Pacific Conf.on Commun. (APCC)*, pages 132–137, 2013.

[28] Changhui Jiang, Yuwei Chen, Chen Chen, Jianxin Jia, Haibin Sun, Tinghuai Wang, and Juha Hyyppä. Smartphone PDR/GNSS integration via factor graph optimization for pedestrian navigation. *IEEE Trans. on Instrum. and Meas.*, 71:1–12, 2022.

[29] Xiande Bu, Chuan Liu, Qiang Yu, Lei Yin, and Feng Tian. Optimization on cooperative communications based on network coding in multi-hop wireless networks. In *Int. Wireless Commun. and Mobile Comput. (IWCMC)*, pages 384–387, 2020.

[30] Jiajie Xue and Brian Michael Kurkoski. Lower bound on the error rate of genie-aided lattice decoding. In *IEEE Int. Symp. on Inf. Theory (ISIT)*, pages 3232–3237, 2022.

[31] Julia Rosenberger, Michael Urlaub, Felix Rauterberg, Tina Lutz, Andreas Selig, Michael Bühren, and Dieter Schramm. Deep reinforcement learning multi-agent system for resource allocation in industrial internet of things. *Sensors*, 22(11):4099, 2022.

[32] Y. Wang, F. Shang, and J. Lei. Reliability optimization for channel resource allocation in multihop wireless network: A multigranularity deep reinforcement learning approach. *IEEE Internet of Things J.*, 9(20):19971–19987, 2022.

[33] T. Ho and K. Nguyen. Joint server selection, cooperative offloading and handover in multi-access edge computing wireless network: A deep reinforcement learning approach. *IEEE Trans. on Mobile Comput.*, 21(7):2421–2435, 2022.

[34] T Mazhar, I Haq, A Ditta, S. A. H. Mohsan, F Rehman, I Zafar, J. A. Gansau, and L. P. W. Goh. The role of machine learning and deep learning approaches for the detection of skin cancer. *Healthcare*, 11(3):415, 2023.

[35] Y Lee, X Ma, SID. A Lang, F. E Valderrama-Araya, and L. A Chapuis. Deep learning based modeling of wireless communication channel with fading. *arXiv preprint arXiv:2312.06849*, 2023.

[36] Dmitrii A. Dugaev, Ivan G. Matveev, Eduard Siemens, and Viatcheslav P. Shuvalov. Adaptive reinforcement learning-based routing protocol for wireless multihop networks. In *Int. Scientific-Tech. Conf. on Actual Problems of Electron. Instrum. Eng. (APEIE)*, pages 209–218, 2018.

[37] Tanutsorn Wongphatcharatham, Watid Phakphisut, Thongchai Wijitpornchai, Poonlarp Areeprayoonkij, Tanun Jaruvitayakovit, and Pimkhuan Hannanta-anan. Multi-agent Q-learning for power allocation in interference channel. In *Int. Tech. Conf. on Circuits/Syst., Comput. and Commun. (ITC-CSCC)*, pages 876–879, 2022.

[38] Xiaowei Wang and Xin Wang. Reinforcement learning-based multihop relaying: A decentralized Q-learning approach. *Entropy (Basel)*, 23(10), 2021.

[39] Y. Su, Xi. Lu, Y. Zhao, L. Huang, and X. Du. Cooperative communications with relay selection based on deep reinforcement learning in wireless sensor networks. *IEEE Sensors J.*, 19(20):9561–9569, 2019.

[40] Aung Thura Phyo Khun, Yuto Lim, and Yasuo Tan. Optimal achievable transmission capacity scheme with transmit power control in full-duplex wireless multihop networks. In *Int. Conf. on Mobile Comput. and Ubiquitous Netw. (ICMU)*, pages 1–6, 2021.

[41] ANSI/IEEE Standard 802.11. Part II: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications mesh networking. *IEEE Standard Specification 802.11s Mesh WLAN*, pages 1–372. , 2011.

[42] Francisco S Melo. Convergence of Q-learning: A simple proof. *Inst. of Syst. and Robot., Tech. Rep*, pages 1–4, 2001.

[43] Bobak Nazer and Michael Gastpar. Compute-and-forward: Harnessing interference through structured codes. *IEEE Trans. on Inf. Theory (ISIT)*, 57(10):6463–6486, 2011.

[44] A. O. Lim, X. Wang, Y. Kado, and B. Zhang. Cooperative communications with relay selection based on deep reinforcement learning in wireless sensor networks. *Mobile Netw. and Appl.*, 19(20):9561–9569, 2019.

[45] ANSI/IEEE Standard 802.11. Part II: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications enhancements for high-efficiency WLAN. *IEEE Standard Specification 802.11ax*, pages 1–767. , 2021.

[46] Kun Guo and Tony Q. S. Quek. Dynamic computation offloading in multi-server mec systems: An online learning approach. In *IEEE Global Commun. Conf.*, pages 1–6, 2020.

[47] Qi Gan, Guoxin Li, Wenhui He, Yinling Zhao, Yuchao Song, and Chenglong Xu. Delay-minimization offloading scheme in multi-server mec networks. *IEEE Wireless Commun. Lett.*, 12(6):1071–1075, 2023.

[48] Ducsun Lim and Inwhee Joe. A DRL-based task offloading scheme for server decision-making in multi-access edge computing. *Electronics*, 2023.

[49] O. Ajayi, Xianghui Cao, Hangguan Shan, and Yu Cheng. Self-renewal machine learning approach for fast wireless network optimization. *IEEE 20th Int. Conf. on Mobile Ad Hoc and Smart Syst. (MASS)*, pages 134–142, 2023.

[50] Ping Hu, Yi Zhong, and Yuchen Lai. Capacity prediction for wireless networks based on convolutional neural network. In *Int. Conf. on Inf. and Commun. Technol. for Disaster Manage. (ICT-DM)*, pages 1–8, 2021.

[51] Oluwaseun T. Ajayi, Shuai Zhang, and Yu Cheng. Machine learning assisted capacity optimization for b5g/6g integrated access and backhaul networks. In *IEEE INFO-*

COM 2023 - IEEE Conf. on Comput. Commun. Workshops (INFOCOM WKSHPS), pages 1–6, 2023.

[52] Roman Zhohov, Alexandros Palaios, Henrik Rydén, Reza Moosavi, and Joel Berglund. Reducing latency: Improving handover procedure using machine learning. In *IEEE 93rd Veh. Technol. Conf. (VTC2021-Spring)*, pages 1–5, 2021.

[53] Hee-Jun Lee and Sang-Hwa Chung. A scheduling method for reducing latency in wi-sun fan networks. In *14th Int. Conf. on Ubiquitous and Future Netw. (ICUFN)*, pages 439–444, 2023.

[54] C. L. Philip Chen and Zhulin Liu. Broad learning system: An effective and efficient incremental learning system without the need for deep architecture. *IEEE Trans. on Neural Netw. and Learn. Syst.*, 29(1):10–24, 2018.

[55] Meiling Xu, Min Han, C. L. Philip Chen, and Tie Qiu. Recurrent broad learning systems for time series prediction. *IEEE Trans. on Cybern.*, 50(4):1405–1417, 2020.

[56] Xinrong Gong, Tong Zhang, C. L. Philip Chen, and Zhulin Liu. Research review for broad learning system: Algorithms, theory, and applications. *IEEE Trans. on Cybern.*, 52(9):8922–8950, 2022.

[57] Jiancheng Chi, Zhihan Cui, and Yuto Lim. BLSO: Broad learning system-based scheme for adaptive task offloading in industrial iot. In *Int. Conf. on Mobile Comput. and Ubiquitous Net. (ICMU)*, pages 1–6, 2023.

[58] Weisong Shi, Jie Cao, Quan Zhang, Youhuizi Li, and Lanyu Xu. Edge computing: Vision and challenges. *IEEE Internet of Things J.*, 3(5):637–646, 2016.

[59] LLC Gurobi Optimization. Gurobi optimizer. `https://www.gurobi.com/`, 2025.

[60] Yousef Sanjalawe, Salam Fraihat, Salam Al-E'Mari, Mosleh Abualhaj, Sharif Makhadmeh, and Emran Alzubi. A review of 6G and AI convergence: Enhancing communication networks with artificial intelligence. *IEEE Open Journal of the Communications Society*, 6:2308–2355, Mar 2025.

[61] Tianjiao Chen, Juan Deng, Qinqin Tang, and Guangyi Liu. Optimization of quality of AI service in 6G native AI wireless networks. *Electronics*, 12(15), Aug 2023.

[62] Shi Dong, Junxiao Tang, Khushnood Abbas, Ruizhe Hou, Joarder Kamruzzaman, Leszek Rutkowski, and Rajkumar Buyya. Task offloading strategies for mobile edge computing: A survey. *Computer Networks*, 254:110791, Sep 2024.

[63] Peng Peng, Weiwei Lin, Wentai Wu, Haotong Zhang, Shaoliang Peng, Qingbo Wu, and Keqin Li. A survey on computation offloading in edge systems: From the perspective of deep reinforcement learning approaches. *Computer Science Review*, 53:100656, Jun 2024.

[64] Saumyaranjan Dash, Asif Uddin Khan, Santosh Kumar Swain, and Binayak Kar. Clustering based efficient MEC server placement and association in 5G networks. In *Proceedings of the 2021 19th OITS International Conference on Information Technology (OCIT)*, pages 167–172, Bhubaneswar, India, Dec 2021.

[65] Theodore S. Rappaport, Yunchou Xing, Ojas Kanhere, Shihao Ju, Arjuna Madanayake, Soumyajit Mandal, Ahmed Alkhateeb, and Georgios C. Trichopoulos. Wireless communications and applications above 100 GHz: Opportunities and challenges for 6G and beyond. *IEEE Access*, 7:78729–78757, Jun 2019.

[66] Chong Han, Yiqin Wang, Yuanbo Li, Yi Chen, Naveed A. Abbasi, Thomas Kürner, and Andreas F. Molisch. Terahertz wireless channels: A holistic survey on measurement, modeling, and analysis. *IEEE Communications Surveys and Tutorials*, 24(3):1670–1707, Jun 2022.

[67] Farrukh Salim Shaikh and Roland Wismüller. Routing in multi-hop cellular device-to-device networks: A survey. *IEEE Communications Surveys and Tutorials*, 20(4):2622–2657, Jun 2018.

[68] Nicole Todtenberg and Rolf Kraemer. A survey on bluetooth multi-hop networks. *Ad Hoc Networks*, 93:101922, Jun 2019.

[69] Zhuohui Yao, Wenchi Cheng, Wei Zhang, Tao Zhang, and Hailin Zhang. The rise of UAV fleet technologies for emergency wireless communications in harsh environments. *IEEE Network*, 36(4):28–37, Oct 2022.

[70] Indu Chandran and Kizheppatt Vipin. Multi-UAV networks for disaster monitoring: challenges and opportunities from a network perspective. *Drone Systems and Applications*, 12:1–28, Jul 2024.

[71] Xuehua Li, Jiuchuan Zhang, and Chunyu Pan. Federated deep reinforcement learning for energy-efficient edge computing offloading and resource allocation in industrial internet. *Applied Sciences*, 13(11), 2023.

[72] Yunfei Zheng, Badong Chen, Shiyuan Wang, and Weiqun Wang. Broad learning system based on maximum correntropy criterion. *IEEE Transactions on Neural Networks and Learning Systems*, 32(7):3083–3097, Jul 2021.

[73] Sihua Wang, Mingzhe Chen, Xuanlin Liu, Changchuan Yin, Shuguang Cui, and H. Vincent Poor. A machine learning approach for task and resource allocation in mobile-edge computing-based networks. *IEEE Internet of Things Journal*, 8(3):1358–1372, Feb 2021.

[74] Yan Chen, Yanjing Sun, Chenyang Wang, and Tarik Taleb. Dynamic task allocation and service migration in edge-cloud IoT system based on deep reinforcement learning. *IEEE Internet of Things Journal*, 9(18):16742–16757, Sep 2022.

[75] Yan Chen, Yanjing Sun, Hao Yu, and Tarik Taleb. Joint task and computing resource allocation in distributed edge computing systems via multi-agent deep reinforcement learning. *IEEE Transactions on Network Science and Engineering*, 11(4):3479–3494, Jul 2024.

[76] Zehong Lin, Suzhi Bi, and Ying-Jun Angela Zhang. Optimizing AI service placement and resource allocation in mobile edge intelligence systems. *IEEE Transactions on Wireless Communications*, 20(11):7257–7271, Nov 2021.

[77] Shidrokh Goudarzi, Seyed Ahmad Soleymani, Mohammad Hossein Anisi, Anish Jindal, and Pei Xiao. Optimizing UAV-assisted vehicular edge computing with age of information: An sac-based solution. *IEEE Internet of Things Journal*, 12(5):4555–4569, Mar 2025.

[78] Dinh-Hieu Tran, Van-Dinh Nguyen, Symeon Chatzinotas, Thang X. Vu, and Björn Ottersten. UAV relay-assisted emergency communications in IoT networks: Resource allocation and trajectory optimization. *IEEE Transactions on Wireless Communications*, 21(3):1621–1637, Mar 2022.

[79] Manzoor Ahmed, Salman Raza, Haseeb Ahmad, Wali Ullah Khan, Fang Xu, and Khaled Rabie. Deep reinforcement learning approach for multi-hop task offloading in vehicular edge computing. *Engineering Science and Technology, an International Journal*, 59:101854, Oct 2024.

[80] Nguyen Tien Hoa, Do Van Dai, Le Hoang Lan, Nguyen Cong Luong, Duc Van Le, and Dusit Niyato. Deep reinforcement learning for multi-hop offloading in UAV-assisted edge computing. *IEEE Transactions on Vehicular Technology*, 72(12):16917–16922, Dec 2023.

[81] Wei Zhao, Tangjie Weng, Yu Cheng, Zhi Liu, and Nei Kato. Deep reinforcement learning for optimizing multi-hop distributed collaborative task offloading in R2X. *IEEE Transactions on Vehicular Technology*, pages 1–15, Jan 2025.

[82] Xianhao Chen, Guangyu Zhu, Yiqin Deng, and Yuguang Fang. Federated learning over multihop wireless networks with in-network aggregation. *IEEE Transactions on Wireless Communications*, 21(6):4622–4634, Jun 2022.

[83] Maoxin Ji, Qiong Wu, Pingyi Fan, Nan Cheng, Wen Chen, Jiangzhou Wang, and Khaled B. Letaief. Graph neural networks and deep reinforcement learning-based resource allocation for V2X communications. *IEEE Internet of Things Journal*, 12(4):3613–3628, Feb 2025.

[84] Oğuzhan Akyıldız, Feyza Yıldırım Okay, İbrahim Kök, and Suat Özdemir. Mx-TORU: Location-aware multi-hop task offloading and resource optimization protocol for connected vehicle networks. *Computer Networks*, 259:111094, Feb 2025.

[85] Bo Fan, Zhengbing He, Yuan Wu, Jia He, Yanyan Chen, and Li Jiang. Deep learning empowered traffic offloading in intelligent software defined cellular V2X networks. *IEEE Transactions on Vehicular Technology*, 69(11):13328–13340, Sep 2020.

[86] He Li, Kaoru Ota, and Mianxiong Dong. Learning IoT in edge: Deep learning for the internet of things with edge computing. *IEEE Network*, 32(1):96–101, Jan 2018.

[87] Robin J. Wilson. *Introduction to Graph Theory*. Prentice Hall, New York, 5 edition, May 2010.

[88] Shahbaz Rezaei, Mohammed Gharib, and Ali Movaghar. Throughput analysis of IEEE 802.11 multi-hop wireless networks with routing consideration: A general framework. *IEEE Transactions on Communications*, 66(11):5430–5443, Nov 2018.

[89] Aung Thura Phyo Khun, Hao Wang, Yuto Lim, and Yasuo Tan. Consensus transmit power control with optimal search technique for full-duplex wireless multihop networks. In *Proceedings of the 2021 26th IEEE Asia-Pacific Conference on Communications (APCC)*, pages 62–67, Kuala Lumpur, Malaysia, Nov 2021.

List of Publications

Journals

[1] **Zhihan Cui**, Yan Chen, Yuto Lim, and Tarik Taleb, "Joint Server Allocation and Path Selection in Wireless Multihop Networks with Edge Computing," *IEEE Internet of Things Journal*, vol. 12, no. 23, pp. 49768-49783, 2025, doi: 10.1109/JIOT.2025.3605580.

[2] **Zhihan Cui**, Yuto Lim, and Yasuo Tan, "Factor Graph-based Deep Reinforcement Learning for Path Selection Scheme in Full-duplex Wireless Multihop Networks," *Ad Hoc Networks*, vol. 161, pp. 103542, 2024, doi: 10.1016/j.adhoc.2024.103542.

International Conference

[3] **Zhihan Cui**, Jianwen Chi, Yuto Lim, and Yasuo Tan, " BLSQ: AI-Enhanced Performance Framework for Wireless Multihop Network," *IEEE Region 10 Conference (TENCON)*, Sabah, Malaysia, 2025.

[4] **Zhihan Cui**, Jianwen Chi, Yuto Lim, and Yasuo Tan, "Broad Learning System Scheme for Multi-server MEC Wireless Networks," *International Conference on Advanced Information Networking and Applications (AINA)*, Barcelona, Spain, 2025.

[5] Jiancheng Chi, **Zhihan Cui**, and Yuto Lim, "BLSO: Broad Learning System-Based Scheme for Adaptive Task Offloading in Industrial IoT," *International Conference on Mobile Computing and Ubiquitous Network (ICMU)*, Osaka, Japan, 2023.

[6] **Zhihan Cui**, Aung Thura Phyo Khun, Yuto Lim, and Yasuo Tan, "Factor Graph-based Deep Reinforcement Learning for Path Selection Scheme in Full-duplex Wire-

less Multihop Networks," *International Wireless Communications and Mobile Computing (IWCMC)*, Marrakech, Morocco, 2023.

Domestic Conference

[7] **Zhihan Cui**, Yan Chen, Yuto Lim, and Tarik Taleb, "BLSQ for Multi-server Wireless Multihop Networks," *IEICE Technical Committee on Sensor Network and Mobile Intelligence (SeMI)*, vol. 125, no. 135, SeMI2025-27, pp. 53–58, 2025.

[8] **Zhihan Cui**, Yuto Lim, and Yasuo Tan, "Performance Analysis for Wireless Multihop Networks with Broad Learning System and Q-Learning," *IEICE Technical Committee on Information Networks (IN)*, vol. 124, no. 420, IN2024-104, pp. 152–157, 2025.

[9] **Zhihan Cui**, Jiancheng Chi, Yuto Lim, and Yasuo Tan, "Broad Learning System Scheme for Multi-server Wireless Networks," *IEICE Technical Committee on Sensor Network and Mobile Intelligence (SeMI)*, vol. 124, no. 109, SeMI2024-13, pp. 7–12, 2024.

[10] Jianwen Sun, **Zhihan Cui**, Yuto Lim, and Yasuo Tan, "Design and Throughput Analysis of Concurrent PNC Scheme in Full-Duplex Wireless Networks," *IEICE Technical Committee on Sensor Network and Mobile Intelligence (SeMI)*, vol. 124, no. 109, SeMI2024-14, pp. 13–18, 2024.

[11] **Zhihan Cui**, Aung Thura Phyo Khun, Yuto Lim, and Yasuo Tan, "Study of Deep Reinforcement Learning for Wireless Multihop Networks," *IEICE Technical Committee on Sensor Network and Mobile Intelligence (SeMI)*, vol. 122, no. 309, SeMI2022-113, pp. 37–42, 2023.

[12] **Zhihan Cui**, Aung Thura Phyo Khun, Yuto Lim, and Yasuo Tan, "Study of Network Capacity in Wireless Multihop Networks with Nested Lattice and Factor Graph," *IEICE Technical Committee on Information Networks (IN)*, vol. 122, no. 233, IN2022-41, pp. 53–58, 2022.

[13] **Zhihan Cui**, Aung Thura Phyo Khun, Yuto Lim, and Yasuo Tan, "Study of Network Capacity in Wireless Multihop Networks with Factor Graph," *Joint Conference of Hokuriku Chapters of Electrical and Information Societies (JHES)*, 2022.