Joint Server Allocation and Path Selection in Wireless Multihop Networks with Edge Computing

Zhihan Cui, Yan Chen, *Member, IEEE* Yuto Lim, and Tarik Taleb, *Senior Member, IEEE*

***Abstract*—With the rapid evolution towards Beyond 5G and future 6G networks, multi-access edge computing (MEC)-enabled wireless networks are expected to support massive device connectivity, ultra-low latency, and high network capacity. However, meeting these stringent requirements in multi-server wireless multihop networks essentially requires the joint orchestration of server selection, multihop routing, and interference management. This paper develops a novel three-stage optimization scheme named broad learning system with Q-learning (BLSQ), consisting of a broad learning system-based server allocation stage, a signal-to-interference-plus-noise ratio-driven Q-learning-based multihop path selection stage, and a consensus transmit power control stage for adaptive interference mitigation. Furthermore, a consensus transmit power control mechanism is incorporated to adaptively adjust the transmit power of user devices, aiming to balance interference mitigation and throughput enhancement. The proposed scheme is particularly suitable for various mission-critical and dynamic scenarios, such as emergency communication in disaster-stricken areas, multihop data exchange between rescue teams and command centers, and flexible network deployment in large-scale events using unmanned aerial vehicles. Extensive simulation results demonstrate that the proposed BLSQ schemes outperforms existing related approaches in terms of network capacity, task completion time, interference management, and quality of servers, validating the superiority and robustness of our design for future MEC-enabled wireless networks.**

***Index Terms*—Wireless multihop networks, server allocation, path selection algorithm, broad learning system, Q-learning, and transmit power control.**

I. Introduction

With the evolution toward Beyond 5G (B5G) and future 6G networks, emerging applications demand unprecedented levels of connectivity, ultra-low latency, and high network capacity. According to the IMT-2030 framework proposed by ITU-R [1], future mobile networks are expected to support connection densities ranging from 10^6 to 10^8 devices per square kilometer, with latency requirements as low as 0.1 to 1.0 milliseconds. 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 [2], [3]. 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 multi-access edge computing (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 applications [4], [5]. 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 [6], [7]. 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 [8], [9]. In this context, multihop wireless networking provides a cost-effective and scalable alternative by leveraging device relaying and localized communication. Multi-server wireless multihop networks (MWMNs) have emerged as a promising architecture, where devices can communicate through multi-hop wireless links to offload their tasks to edge servers (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 [10], [11], 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 unmanned aerial vehicles (UAVs) and mobile ESs to support surveillance, sensing, and emergency communications [12], [13]. 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 quality of service (QoS) requirements, efficient data forwarding strategies, robust wireless communications, as well as resource restrictions. This includes optimal server selection,

This work is partly conducted at ICTFICIAL Oy, Finland. It is partly supported by a Grant-in-Aid for Scientific Research (B) from the Japan Society for the Promotion of Science (JSPS), Japan. Grant number is 23K28071; in part by the European Union's HE research and innovation program HORIZON-JUSNS-2023 through the 6G-Path project under Grant No. 101139172, in part by the European Union's Horizon 2020 Research and Innovation Program through the aerOS project under Grant No. 101069732. The paper reflects only the authors' views, and the European Commission bears no responsibility for any utilization of the information contained herein. (*Corresponding author: Yan Chen*)

Zhihan Cui and Yuto Lim are with the School of Information Science, Japan Advanced Institute of Science and Technology, Ishikawa 9231292, Japan (e-mail: s2320024@jaist.ac.jp, ylim@jaist.ac.jp).

Yan Chen and Tarik Taleb are with the Faculty of Electrical Engineering and Information Technology, Ruhr University Bochum, 44801 Bochum, Germany (e-mail: yanchen@ieee.org, tarik.taleb@rub.de).

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 reinforcement learning (RL). Meanwhile, RL-based algorithms [14] are widely developed to optimize multihop routing by considering wireless link quality metrics such as signal-to-noise ratio (SNR) and signal-to-interference-plus-noise ratio (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 [15]. 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 [16] 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 [17]–[19]. 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 paper 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 broad learning system with Q-learning (BLSQ) 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 BLSQ 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.

The rest of this paper is organized as follows. Section II reviews the related works. Section III presents the system model and problem formulation of MWMNs. Section IV details the proposed BLSQ schemes. Section V evaluates the performance of the proposed schemes through extensive simulations under different network scenarios. Finally, Section VI concludes this paper and outlines potential future research directions.

II. Related Works

A. 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 Internet of Things (IoT) environments. Wang et al. [20] 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. [21] 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. [22] 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. [23] 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. [24] 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. [25] tackled UAV relay-assisted IoT communication, optimizing resource allocation and UAV trajectory to serve more devices under latency and storage constraints.

B. Wireless Multihop Networks

In wireless multihop networks, extensive research has focused on optimizing task offloading, routing, and resource

allocation. Ahmed et al. [26] proposed a proximal policy optimization-based RL algorithm to minimize task execution delay in multihop vehicular task offloading. Nguyen et al. [27] presented a UAV-assisted multihop edge computing architecture using deep Q-learning for task partitioning and offloading. Zhao et al. [28] 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. [29] proposed a framework over wireless mesh networks with in-network model aggregation and joint optimization of routing and spectrum allocation. Ji et al. [30] introduced a GNN-assisted DRL for V2X communications, modeling V2V interference as graph edges for distributed resource allocation. Akyildiz et al. [31] proposed a mobility-driven multihop task offloading and resource optimization protocol for connected vehicular networks.

C. Broad Learning System

Machine learning (ML) techniques have been widely used for task allocation and offloading in wireless networks, including deep learning-based methods [32], [33]. Recently, BLS has gained interest, thanks to its non-deep learning framework, which offers fast learning and low computational complexity. Xu et al. [34] introduced recurrent BLS for time series prediction, enhancing its ability to handle sequential data. Chi et al. [17] proposed a BLS-based task offloading scheme (BLSO) in Industrial IoT 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.

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 converge of communication and computation, jointly impacting the application QoS and network capacity. However, making these decisions remains challenging due to the dynamic network topology, interference, and device characteristics. In addition, few existing studies have addressed the adaptive transmit power control in dense multihop wireless networks to balance interference mitigation and throughput enhancement.

To address these research gaps, this paper proposes a novel optimization scheme that jointly considers server selection and multihop routing in MWMNs. Specifically, a BLS-based strategy is utilized to perform efficient server selection, while a SINR-driven Q-learning algorithm is designed for multihop path planning. Furthermore, a consensus transmit power control (CTPC) algorithm is incorporated to dynamically adjust the transmit power of user devices, aiming to improve network capacity and reduce interference. These contributions enable the proposed scheme to achieve superior performance in terms of network scalability, latency reduction, and interference management, particularly in dynamic and mission-critical network scenarios.

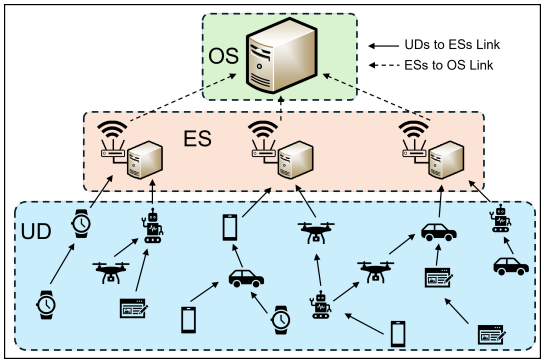


Fig. 1: Illustration of the Envisiones System Model.

III. System Model

A. Network Model

This paper considers a three-tier MWMN, as presented in Fig. 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 ($\mathcal{E} = \{1, 2, \cdots, E\}$) are 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.

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 reach 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.

B. Network Operation

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

1) 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 makes the decision 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 $x_u^e(t) = \{0,1\}, u \in \mathcal{U}, e \in \mathcal{E}$ to represent the decision of server allocation at time t. $x_u^e = 1$ indicates that the task of UD u is allocated to be processed by ES e, and $x_u^e = 0$ otherwise. Since each UD's task can only be offloaded to a single ES, the following constraint is applied:

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

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 [17], i.e.,

$$t_{u,e}^{est} = \hat{t}_{u,e}^{up} + \hat{t}_{u,e}^{down} + t_{u,e}^{proc}, \tag{2}$$

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 (F_e):

$$t_{u,e}^{proc} = \frac{L_u}{F_e}. \tag{3}$$

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

$$\hat{t}_{u,e}^{up} = \frac{L_u}{R_{u,e}}, \quad \hat{t}_{u,e}^{down} = \frac{L_u^{res}}{R_{u,e}}. \tag{4}$$

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

$$R_{u,e} = B \cdot \log_2\left(1 + \gamma_{u,e}\right), \tag{5}$$

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

$$\gamma_{u,e} = \frac{G_{u,e} \cdot P_u}{\eta \cdot B + \sum_{k \in \mathcal{K}} G_{k,e} \cdot P_k}, \tag{6}$$

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,

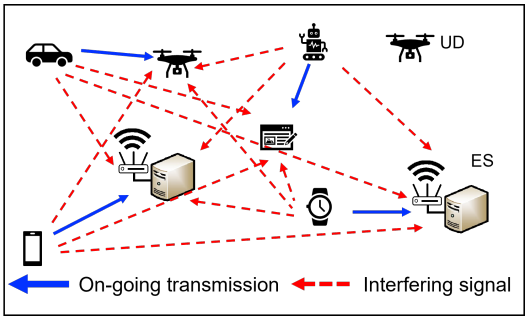


Fig. 2: Illustration of Interference.

where all UDs in the network contribute to interference during transmission. As shown in Fig. 2, the interference model considers 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 [35], i.e.,

$$G_{u,e} = \frac{1}{10^{\left(\frac{PL_{u,e}}{10}\right)}}, \tag{7}$$

where

$$PL_{u,e} = 20 \cdot \log_{10}(d_0) + 10 \cdot \zeta \cdot \log_{10}(\frac{d_{u,e}}{d_0}) - \omega + \psi. \tag{8}$$

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:

$$\mathbf{C2}: \quad t_{u,e}^{est} \leq \tau_u, \quad \forall u \in \mathcal{U}. \tag{9}$$

2) 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

$$\boldsymbol{y}_u = [y_{i,j}]_{(|\mathcal{U}|) \times (|\mathcal{U}|+|\mathcal{E}|)}, \tag{10}$$

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:

$$\mathbf{C3}: \begin{cases} \sum_{j \in \mathcal{U} \cup \mathcal{E}} y_{u,j} = 1, \sum_{j \in \mathcal{U} \cup \mathcal{E}} y_{j,u} = 0, \\ \sum_{j \in \mathcal{U} \cup \mathcal{E}} y_{j,e} = 1, \sum_{j \in \mathcal{U} \cup \mathcal{E}} y_{e,j} = 0, \\ \sum_{j \in \mathcal{U} \cup \mathcal{E}} y_{i,j} = \sum_{j \in \mathcal{U} \cup \mathcal{E}} y_{j,i} \leq 1, \end{cases} \tag{11}$$

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 [36].

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 end-to-end (E2E) throughput serves as a critical performance metric that reflects the overall transmission capability of the selected path [37]. 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

$$R_{u,e}^{ave} = \frac{\sum_{i \in \mathcal{U} \cup \mathcal{E}} \sum_{j \in \mathcal{U} \cup \mathcal{E}} y_{i,j} \cdot R_{i,j}}{\sum_{i \in \mathcal{U} \cup \mathcal{E}} \sum_{j \in \mathcal{U} \cup \mathcal{E}} y_{i,j}}. \tag{12}$$

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.,

$$\textbf{C4}: \quad t_{u,e}^{ser} \leq t_{u,e}^{est}, \quad \forall u \in \mathcal{U}, \tag{13}$$

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 ($t_{u,e}^{down}$):

$$t_{u,e}^{ser} = t_{u,e}^{up} + t_{u,e}^{down} + t_{u,e}^{proc}. \tag{14}$$

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

$$t_{u,e}^{up} = \frac{L_u}{R_{u,e}^{ave}}, \quad t_{u,e}^{down} = \frac{L_u^{res}}{R_{u,e}^{down}}, \tag{15}$$

where R_u^{ave} is the average transmission rate across the multihop 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:

$$\gamma_{u,e}^{down} = \frac{G_{u,e} \cdot P_e}{\eta \cdot B + \sum_{e' \in \mathcal{E}, e' \neq e} G_{u,e'} \cdot P_{e'}}, \tag{16}$$

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:

$$R_{u,e}^{down} = B \cdot \log_2 \left(1 + \gamma_{u,e}^{down}\right). \tag{17}$$

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 $t_{u,e}^{ser} \leq t_{u,e}^{est}$.

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 Fig. 3.

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 $T = \{1, 2, \cdots, \}$. At the beginning of

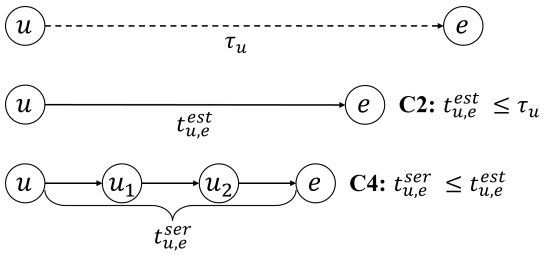


Fig. 3: Illustration of the Relationship between **C2** and **C4**.

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.

C. 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 MWMN. The optimization problem in the OS is formulated as follows:

$$\begin{aligned} \mathbf{P1}: & \min_{x_{u,e}} \sum_{u\in\mathcal{U}} \left[\delta \cdot t_{u,e}^{est} - (1-\delta)\cdot \gamma_{u,e}\right] \\ \mathbf{P2}: & \max_{y_{i,j}, t_{u,e}^{ser}} \quad \sum_{u\in\mathcal{U}} R_{u,e}^{ave} \\ & s.t. \quad \mathbf{C1}, \mathbf{C2}, \mathbf{C3}, \mathbf{C4}, \end{aligned} \tag{18}$$

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 end-to-end 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 Eq. (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, beyond merely satisfying **C4**. 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. The notations defined in this article are listed in TABLE I.

IV. BLSQ Scheme

To effectively address the challenges of optimal server allocation and path selection in MWMNs, we propose a three-stage optimization scheme named BLSQ. The main objective of BLSQ is to minimize task completion time, enhance network capacity, and reduce network interference in dynamically changing network environments.

As shown in Fig. 4, the proposed BLSQ 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 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,

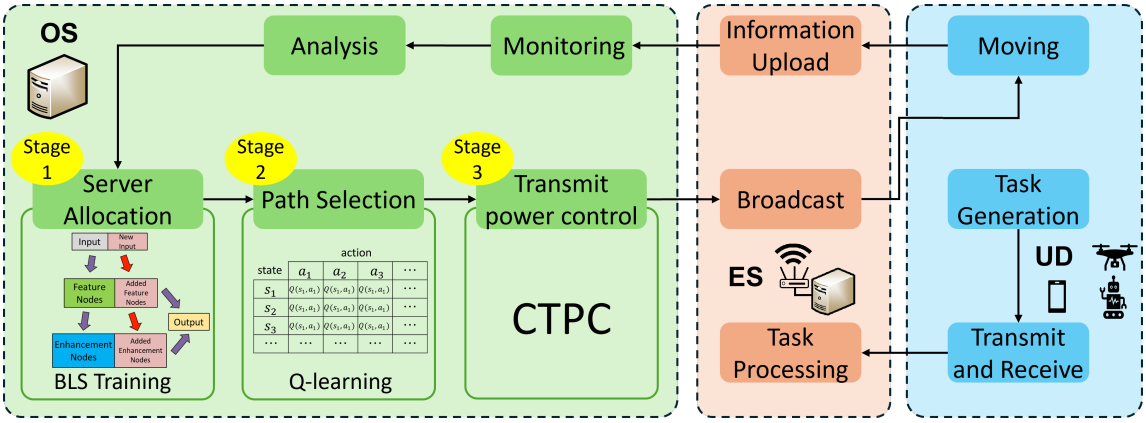


Fig. 4: Overall Framework of the Proposed BLSQ Scheme.

TABLE I: List of Defined Notations.

Symbol	Definition
$\mathcal{U}$	Set of User Devices (UDs)
$\mathcal{E}$	Set of Edge Servers (ESs)
u, e	Indexes for UD and ES, respectively
L_u	Task size of u (bits)
L_u^{res}	Result data size after task processing (bits)
$t_{u,e}^{est}$	Task estimation time (s)
$t_{u,e}^{ser}$	Task server time (s)
$t_{u,e}^{proc}$	Task processing time (s)
$\hat{t}_{u,e}^{up}$	Estimated task upload time from u to e (s)
$\hat{t}_{u,e}^{down}$	Estimated task download time from e to u (s)
$t_{u,e}^{up}$	Actual task upload time via multihop (s)
$t_{u,e}^{down}$	Actual download transmission time via multihop (s)
F_e	Computation speed of e (bits/s)
τ_u	Task tolerable time defined by u (s)
$R_{u,e}$	Transmission rate between u and e (bps)
$R_{i,j}$	Transmission rate of link between i and j (bps)
$R_{u,e}^{ave}$	Average end-to-end throughput (bps)
$\gamma_{u,e}$	SINR between u and e
η	Noise level (W/Hz)
P_u	Transmission power of u (W)
$G_{u,e}$	Channel power gain between u and e
$d_{u,e}$	Distance between u and e (m)
ζ	Attenuation constant
ψ	Shadowing attenuation
B	Channel bandwidth (Hz)
δ	Weighting factor
$x_{u,e}$	Binary indicator of server allocation from u to e
$y_{i,j}$	Binary indicator of link selection between i and j

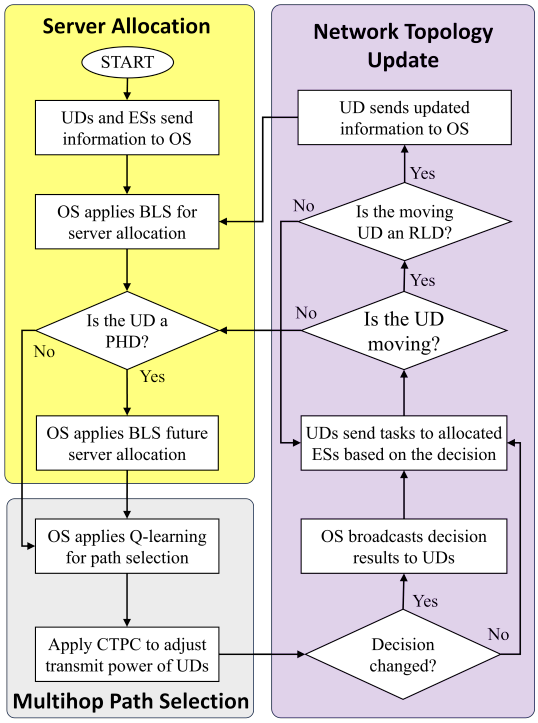


Fig. 5: BLSQ Flowchart.

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 BLSQ scheme also incorporates distinct strategies tailored to different mobility types, ensuring route robustness and task delivery reliability under dynamic topology conditions. The novelty of the proposed BLSQ scheme lies in its unified optimization of server allocation, multihop routing, and transmit power control within a single framework. Compared with existing works, BLSQ leverages a BLS for scalable server association, incorporates SINR-driven Q-learning for interference-aware path selection, and integrates consensus-based transmit power control for robust interference mitigation. These features distinguish our approach and enhance its applicability to dynamic MWMNs. The flowchart of BLSQ is shown in Fig. 5

A. *Broad Learning System-based Server Allocation*

BLS is an incremental supervised learning algorithm designed to update and adapt to dynamic environments without requiring deep network architectures. BLS employs a flat network structure comprising mapped features and enhancement nodes, enabling efficient training and updating processes. The illustration of BLS is shown in Fig. 6

As a supervised learning, BLS requires a structured dataset composed of labeled samples. Each sample consists of input predicted features and corresponding output labels, which are used to train the model for accurate predictions. The input

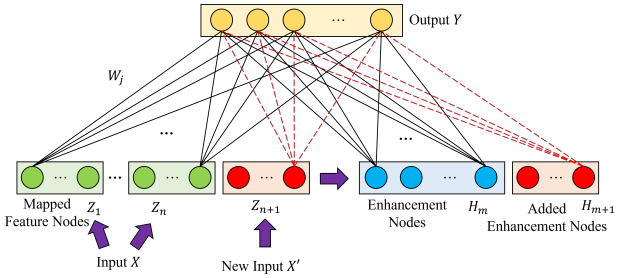


Fig. 6: Illustration of Broad Learning System.

features include the locations of all UDs and ESs, the task size of UDs, and the computational capacity of ESs. To generate the labels, we use the Gurobi optimizer [38] to solve the server allocation subproblem (**P1**) defined in Eq. (18), under a variety of network conditions. The optimal server assignment obtained for each UD serves as the predefined label in the training set. Once the dataset is generated, it is integrated into BLS for training and inference.

The operation of the BLS algorithm consists of the following stages:

1) Data Preprocessing and Feature Extraction: Initially, the datasets are normalized using the min-max normalization defined as

$$z' = \frac{z - z_{min}}{z_{max} - z_{min}}, \tag{19}$$

where z is a sample from the dataset, z' is its normalized value, and z_{min} and z_{max} are the minimum and maximum values of the respective feature across the dataset. Normalization ensures numerical stability and equal importance among features.

To further enhance feature representation, feature extraction is performed using the Random Forest algorithm. Specifically, the importance of each feature is quantified using the Gini impurity metric calculated by:

$$\mathcal{I}(S_n|X) = \sum_{l=1}^{z} \frac{|S_{n,l}|}{|S_n|} \cdot \mathcal{I}(S_{n,l}), \tag{20}$$

where z is the number of branches derived from feature S_n, and $S_{n,l}$ represents the l-th sub-branch. The Gini impurity at the final layer is computed as:

$$\mathcal{I}(S_{n,l}) = \sum_{l=1}^{z} p(x_l)(1 - p(x_l)), \tag{21}$$

where $p(x_l)$ is the proportion of samples belonging to category l. Features with high importance are retained, effectively reducing dimensionality, enhancing prediction accuracy, and mitigating overfitting.

The dataset labels are generated by solving the server allocation optimization problem Eq. (18) using Gurobi solver. The optimal server assignments obtained from Gurobi serve as predefined labels for supervised training. Feature extraction is then performed using the Random Forest algorithm, which identifies and retains the most significant features for server allocation, thereby enhancing model accuracy and reducing overfitting.

2) Feature and Enhancement Node Construction: Following preprocessing, the normalized and extracted data are mapped into feature nodes through randomly generated weight matrices and biases, i.e.,

$$Z_i = \phi(XW_{ei} + \beta_{ei}), \quad i = 1, \ldots, n, \tag{22}$$

where X denotes the input data, W_{ei} and β_{ei} are randomly generated weight and bias matrices, respectively, and $\phi(\cdot)$ represents an activation function.

These feature nodes are further expanded into enhancement nodes by

$$H_j = \xi(Z_n W_{hj} + \beta_{hj}), \quad j = 1, \ldots, m, \tag{23}$$

where W_{hj} and β_{hj} are additional randomly generated weights and biases, and $\xi(\cdot)$ is another activation function. This wide structure enhances the learning capability of BLS without requiring deep architectures.

3) Connecting Weights Calculation: With feature and enhancement nodes established, the optimal connection weights to the output nodes are determined through ridge regression approximation, formulated as

$$W = (\lambda I + A^T A)^{-1} A^T Y, \tag{24}$$

where A is the concatenation of feature and enhancement nodes, Y is the labeled dataset obtained by solving the optimization problem with a Gurobi solver, and λ is the regularization parameter that mitigates overfitting.

4) Incremental Learning: A significant advantage of BLS is its incremental learning capability, allowing rapid model updating in response to network changes, such as new devices joining the network or changes in device mobility patterns. This process involves adding new features and enhancement nodes incrementally, without retraining the entire model from scratch, significantly reducing computational overhead.

B. Q-learning-based Path Selection

Although BLS determines the optimal ES for each UD based on network conditions and task requirements, it lacks the ability to determine the multihop path from UD to ES. This limitation is because BLS is a supervised learning algorithm that only maps input features to predefined labels without considering real-time network dynamics. However, in wireless multihop networks, path selection is a sequential decision-making process that depends on dynamic factors such as interference, device mobility, and SINR. Since BLS does not incorporate graph-based decision-making and cannot adapt to real-time network variations, a separate approach is required to optimize routing.

The multihop path selection problem can be formulated as a Markov Decision Process (MDP) as the system transition only depends on the real-time state and employed action. In this context, the state corresponds to the current transmitting node, characterized by its position, link quality conditions such as SINR, and its set of available neighboring nodes. The action represents the selection of the next-hop relay node among the neighbors, effectively deciding the subsequent transmission link. The reward is defined based on the quality

of the selected wireless link, encouraging the selection of paths with better channel conditions and lower interference, typically measured by achievable throughput or SINR metrics. The transition probability captures the dynamics of moving from one state to another after taking an action, influenced by factors such as device mobility, environmental variations, and channel fluctuations. Finally, the policy defines a strategy that maps each state to an action, aiming to maximize the expected cumulative rewards over time. By solving this MDP, the system can determine an optimal routing policy that adaptively selects high-quality multihop paths in a dynamic network environment. Accordingly, we apply Q-learning, a model-free reinforcement learning algorithm, to enable the agent to learn optimal routing decisions through real-time interactions with the environment without requiring prior knowledge.

We integrate Q-learning into the BLSQ scheme, enabling an adaptive multihop path selection mechanism that dynamically learns and optimizes routing decisions based on real-time network feedback. For any MDP, Q-learning seeks to maximize the reward from the current state across all iterations. In other words, given sufficient exploration time and either completely random or partially random strategies, Q-learning can identify the optimal policy. In Q-learning, the Q-function calculates the expected instantaneous reward for taking a specific action in a given state. The Q-function is typically updated via

$$Q^{new}(s_m, a_m) = (1-\alpha)Q(s_m, a_m) + \alpha\left(r_m + \epsilon Q^{max}(m+1)\right), \tag{25}$$

where s_m is the current state and s_{m+1} is the next state. a_m is the current action. r_m is the immediate reward received after taking action a_m in state s_m. α is the learning rate. ϵ is the discount factor, weighting future rewards. $Q_{max}(m+1)$ represents the highest estimated future Q-value from the next state. The Q-values form a Q-table, which is continuously updated through training. Based on the Q-table, the agent keeps making decisions and doing actions until the reward reaches the target value or the reward changes become smaller than a predefined threshold over multiple iterations.

In our considered scenario, the OS, working as the agent, needs to select multihop links for each UD, forming paths that lead to an ES. This scenario perfectly aligns with Q-learning. Specifically, we design the state, action, and reward for path selection as

- **State** (s_m): The state at decision step m is represented by the currently transmitting node u_m, i.e.,

$$s_m = u_m, \quad u_m \in \mathcal{U} \cup \mathcal{E}, \tag{26}$$

  where $\mathcal{U}$ is the set of UDs and $\mathcal{E}$ is the set of ESs. Each state implicitly includes the node's current network position and the link quality metric with its neighboring nodes.
- **Action** (a_m): The action a_m at step m is the selection of the next-hop receiving node v_m from the neighbors of the current node u_m, i.e.,

$$a_m = v_m, \quad v_m \in \mathcal{N}(u_m), \tag{27}$$

  where $\mathcal{N}(u_m)$ denotes the set of neighboring nodes available for transmission from node u_m.
- **Reward** (r_m): The reward r_m after performing action a_m in state s_m is defined as the link quality metric of the link between nodes u_m and v_m, i.e.,

$$r_m(s_m, a_m) = \gamma_{u_m, v_m}. \tag{28}$$

  Higher SINR values directly indicate better channel conditions and lower interference, motivating the agent to select high-quality communication links.

The transmitting device in a given path represents the "state," while the selected receiving device represents the "action." With each selection of a receiving device, the OS makes decisions based on local network conditions and rewards for each UD accordingly, thus optimizing network performance.

However, Q-learning evaluates Q-values based only on the immediate rewards from current actions. This short-sighted approach may result in sub-optimal overall multihop path performance because it ignores the quality of subsequent links. Even if an immediate link choice performs well, all subsequent links may underperform. Therefore, improvements to the Q-learning algorithm are necessary to encourage longer-term, strategic decision-making in wireless multihop networks. Based on this idea, we propose an enhanced Q-function as

$$Q' = \begin{cases} (1-\alpha)Q(s_m, a_m) + \alpha r_m(s_m, a_m) + \epsilon Q^{\max}(m+1), \\ \qquad\qquad \underline{\text{if} \quad r_m(s_m, a_m) > r_{m+1}} \\ (1-\alpha)Q(s_m, a_m) + \alpha r_{m+1} + \epsilon Q^{\max}(m+1), \\ \qquad\qquad \underline{\text{if} \quad r_m(s_m, a_m) < r_{m+1}} \end{cases} \tag{29}$$

where $Q' \equiv Q^{new}(s_m, a_m)$ for simplified presentation. $r_m(s_m, a_m)$ is the reward of the current link, and r_{m+1} is the reward of the next link, which is the best action that can be made with the current receiving device as state. r_{m+1} can be calculated by

$$r_{m+1} = r_{m+1}(a_m, a'_{m+1}), \tag{30}$$

where a'_{m+1} is the best action with the highest Q-value, i.e.,

$$a'_{m+1} = \arg\max_{a_{m+1}} Q(a_m, a_{m+1}). \tag{31}$$

The maximum Q-value achievable from the next transmitting device ($Q^{max}(m+1)$) can be calculated by

$$Q^{max}(m+1) = Q(a_m, a'_{m+1}). \tag{32}$$

According to the enhanced Q-function, when selecting a path, it is imperative to compare the reward of the current link with the reward of the link in the next hop and select the larger one for the Q-value calculation. This approach enables the OS to prioritize the assessment of the multihop path's overall performance, thereby ensuring the maximization of the total reward upon the selection of a path.

Considering the mobility of UDs, the proposed Q-learning method integrates dynamic updates to the Q-tables to adaptively manage network topology changes. PHDs autonomously update their routing paths based on pre-calculated future Q-values, thus reducing the overhead associated with frequent

recalculations. In contrast, RLDs periodically initiate recalculations by the OS to refine and optimize their routing paths based on updated network states. This adaptive capability ensures robust and efficient network performance, even under highly dynamic conditions.

To effectively realize this adaptive path selection capability and ensure the robustness of the multihop routing decisions, we introduce a two-phase training strategy: the SNR-based Training stage and the SINR-based Training stage.

1) SNR-based Training: In this phase, we employ SNR as the reward metric. This phase identifies paths resilient to noise, laying the groundwork for an interference-aware topology. During this training, each link's SNR is calculated and stored in reward tables. Devices construct Q-tables initialized to zero, representing all possible state-action pairs. Iterative training continues until the SNR-based topology stabilizes.

2) SINR-based Training: Using the topology established in the previous phase, we further refine paths by employing the SINR as the reward metric. This phase considers interference effects, thus enhancing the effectiveness of path optimization. During this training stage, the OS calculates SINR values for each link by accounting for interference from all active transmissions and subsequently updates the Q-values based on these SINR metrics.

In the path selection stage, each device consults its respective Q-table to select relay nodes that maximize the Q-values iteratively. Specifically, each transmitting device identifies and selects the optimal relay device within its transmission range, continuously adapting its choices in response to dynamic network conditions. To ensure the learning process terminates efficiently, a convergence criterion is applied during Q-learning. After each episode of path selection reaching an ES, the UD evaluates $R_{u,e}^{ave}$ of the learned path. If the change in throughput between consecutive iterations falls below a predefined threshold, the learning process is considered to have converged. Once convergence is achieved, the corresponding transmission time is calculated. If this transmission time satisfies the time constraint defined in Eq. (13), the path is recorded as the optimal route for the corresponding UD. Otherwise, the learning continues until both convergence and constraint satisfaction are met.

C. Consensus Transmit Power Control

Multihop transmissions can lead to increased network interference, negatively affecting the overall network performance. To address this issue, we introduce the CTPC algorithm [39], which dynamically optimizes the transmit power at each node, balancing throughput improvement and interference reduction across 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.

1) 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 (denoted as R^{avg} and the average transmit power (denoted as P, which serves as benchmarks for subsequent optimization.

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

$$TG = \frac{R_{CTPC}^{avg}}{R^{avg}}. \tag{33}$$

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

$$PG = \frac{P^{avg}}{P_{CTPC}^{avg}}. \tag{34}$$

These gains are normalized using two weight factors:

$$WT = \frac{PG}{TG + PG}, \quad WP = 1 - WT. \tag{35}$$

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

$$Cost = WT \cdot TG + WP \cdot PG. \tag{36}$$

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

3) 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

$$P_u^{new} = P_u \cdot \mu, \tag{37}$$

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.

V. Simulation Results and Discussion

A. Parameters 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:

$$\Theta = \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} t_{u,e}^{ser}. \tag{38}$$

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:

$$C = \frac{\sum_{u \in \mathcal{U}} (L_u + L_u^{res})}{\Theta}. \tag{39}$$

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.,

$$P^{int} = \sum_{u \in \mathcal{U}} \sum_{k \in \mathcal{U}, k \neq u} G_{k,u} \cdot P_k, \tag{40}$$

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

$$O = \frac{\sum_{i \in \mathcal{U} \cup \mathcal{E}} P_i}{\sum_{i \in \mathcal{U} \cup \mathcal{E}} \sum_{j \in \mathcal{U} \cup \mathcal{E}} R_{i,j}} \quad \text{(J/bit)}, \tag{41}$$

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 setting.

In order to comprehensively evaluate the service quality of the proposed algorithms, we introduce a QoS metric, 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

$$\text{QoS} = \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} \mathbb{I}(u \text{ satisfies } \mathbf{C2} \text{ and } \mathbf{C4}), \tag{42}$$

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 BLSQ, 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 II.

TABLE II: Simulation Parameters and Settings.

Parameter	Value
Network coverage area	500 m × 500 m
Number of UDs	100 ∼ 140
Number of ESs	5 ∼ 9
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

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 BLSQ 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.

B. 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 BLSQ scheme with the Greedy Search (GS) and the existing reinforcement learning-based multihop relaying algorithm, referred to as QBMR [14]. The GS algorithm employs a greedy strategy for server allocation by selecting the server with the minimum $t_{u,e}^{est}$. In the routing 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 BLSQ 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. The evaluation results are shown in Fig. 7.

Fig. 7(a) shows that all three algorithms exhibit an increasing trend in network capacity as the number of UDs grows, due to more effective utilization of the available server resources. Among these, our proposed BLSQ algorithm consistently outperforms both the GS and QBMR algorithms significantly. On average, BLSQ demonstrates about a 30% higher network capacity compared to QBMR and approximately 80% higher than GS. These substantial performance gains by BLSQ 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.

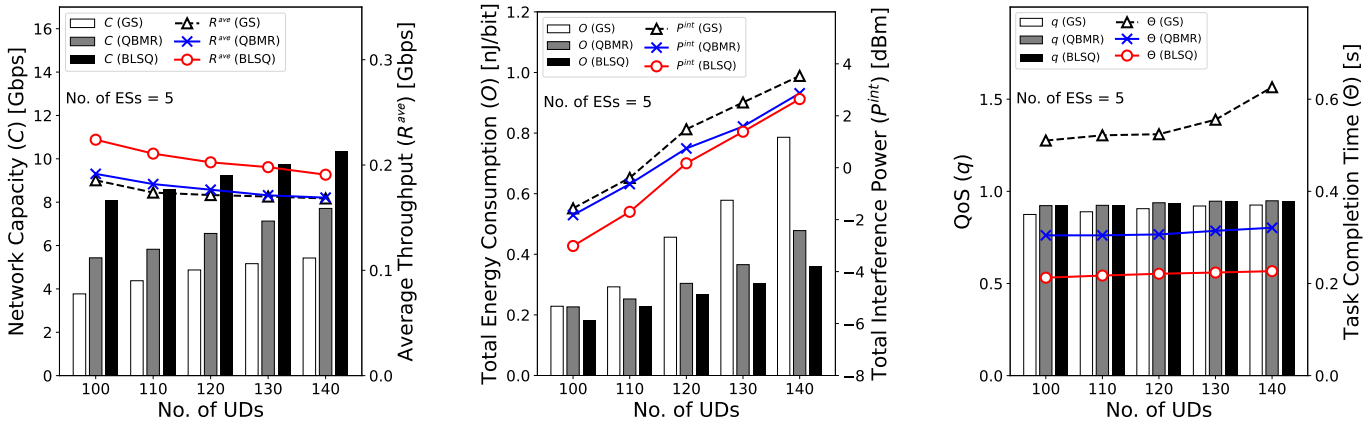
As for average throughput, it is noticeable that all methods shows a decreasing trend as the number of user UDs increases. This behavior is expected due to the intensified resource contention and congestion resulting from a larger number of devices sharing the same fixed server resources. However, even under these challenging conditions, the proposed BLSQ algorithm consistently maintains a higher average throughput compared to GS and QBMR. Overall, BLSQ achieves an average throughput approximately 12%-17% higher than QBMR and 21%-30% higher than GS across the evaluated scenarios. This improved performance clearly highlights the effectiveness of the BLSQ scheme in alleviating network congestion through optimized resource allocation and efficient multihop routing.

As shown in Fig. 7(b), 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 BLSQ algorithm consistently achieves the lowest total interference power across all scenarios. On average, BLSQ significantly reduces interference power by about 1–2 dBm compared to both GS and QBMR. These outcomes highlight BLSQ'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. BLSQ achieves approximately 20% lower energy consumption compared to QBMR and roughly 50% lower than GS. The superior energy efficiency of BLSQ can be attributed to its integrated optimization scheme that incorporates transmission power control and efficient path selection, thereby substantially reducing unnecessary energy expenditure.

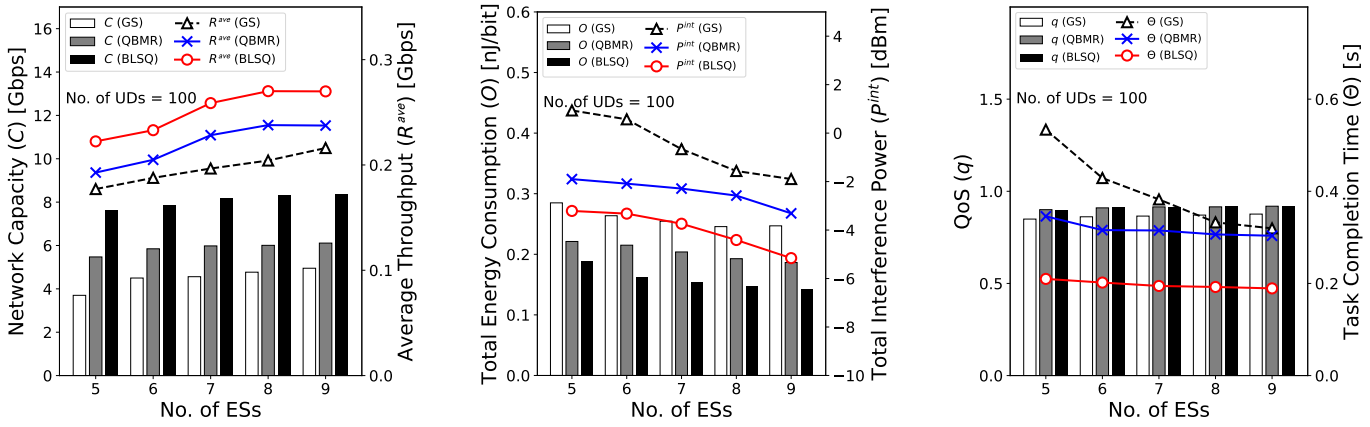
Fig. 7(c) illustrates that the task completion time increases gradually for all algorithms when the number of UDs grows. This is expected since additional UDs introduce higher competition for computing and communication resources, consequently increasing delays. However, our proposed BLSQ algorithm consistently demonstrates significantly lower task completion times compared to both GS and QBMR. BLSQ achieves about 30% shorter task completion time compared to QBMR and roughly 60% lower compared to GS. These improvements indicate that BLSQ 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, the proposed BLSQ algorithm consistently achieves the highest



(a) Network Capacity and Average Throughput. (b) Interference Power and Energy Consumption. (c) Task Completion Time and QoS.

Fig. 7: Performance Comparison with Varying No. of UDs.



(a) Network Capacity and Average Throughput. (b) Interference Power and Energy Consumption. (c) Task Completion Time and QoS

Fig. 8: Performance Comparison with Varying No. of ESs.

QoS among the three. When UDs increase from 100 to 140, BLSQ improves from approximately 92.27% to 94.95%, QBMR increases from 92.24% to 94.86%, and GS rises from 87.45% to 92.56%. Although the performance differences between BLSQ and QBMR are relatively small (less than 1%), BLSQ still consistently maintains a slight edge. Compared to GS, BLSQ demonstrates an evident performance advantage, achieving around 2%-5% higher QoS. These results reflect that BLSQ'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. The results are presented in Fig.8.

As shown in Fig. 8(a), 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 ES, thus enhancing overall network capacity. Specifically, BLSQ'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. BLSQ 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, BLSQ achieves roughly 10%-20% higher throughput than QBMR and around 20%-30 higher throughput compared to GS. These results underline that BLSQ effectively leverages the additional ESs to maximize resource utilization and improve overall network performance.

As shown in Fig. 8(b), 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. BLSQ 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 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. BLSQ 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, BLSQ consumes around 20%-30% less energy compared to QBMR and roughly 40%-50% less compared to GS, emphasizing its superior energy efficiency.

Fig. 8(c) 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, BLSQ consistently delivers the highest QoS. These results clearly demonstrate that BLSQ provides superior quality of service guarantees compared to the baseline methods. Regarding task completion time, all three algorithms exhibit a clear reduction. The proposed BLSQ algorithm 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, BLSQ 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.

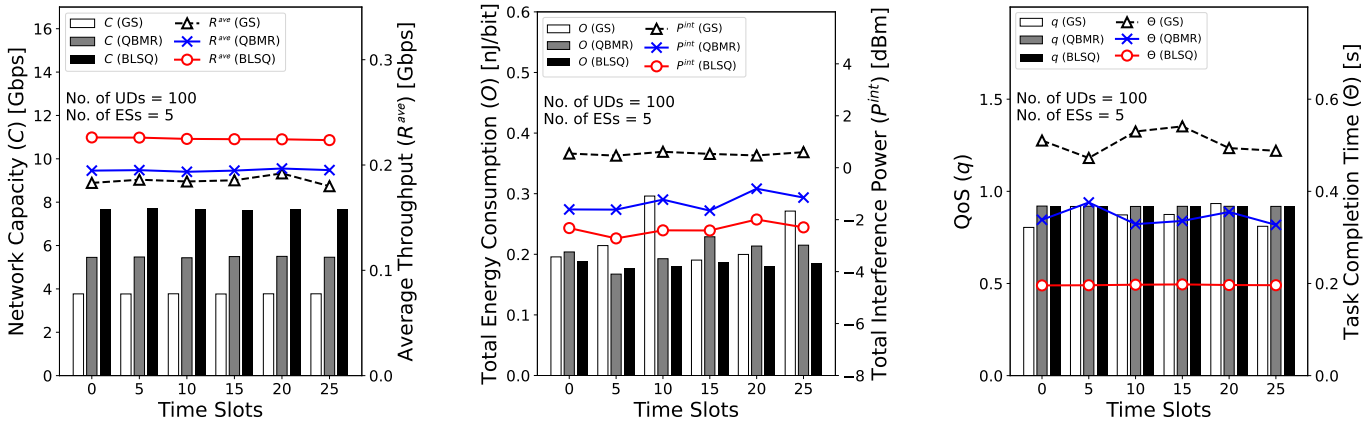
In the third scenario, we focus on the impact of UD mobility on the network performance. We set the number of ESs to 5 and the number of UDs to 100. The time slots for updating location information is varied from 0 to 25 seconds to investigate its influence on the performance metrics. The corresponding results are depicted in Fig. 9.

Fig. 9(a) 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 BLSQ 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 BLSQ 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. BLSQ 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 lowest, varying between 179–192 Mbps. The stable and superior throughput performance of BLSQ highlights its effective management of mobility-related variations through intelligent server allocation and optimized path selection.

Fig. 9(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, BLSQ 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 BLSQ's capability to effectively manage and reduce interference through adaptive power control and optimized path selection strategies, even under mobility-induced network variations. Regarding total energy consumption, measured in nJ/bit, the BLSQ algorithm maintains stable and lowest 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 BLSQ highlights its adaptive energy-aware optimization, effectively handling the extra energy demands typically associated with device mobility.

As shown in Fig. 9(c), for the QoS, BLSQ shows remarkable stability across different location update intervals, both consistently achieving around 92%. Specifically, BLSQ 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 BLSQ effectively ensures reliable QoS in dynamic scenarios, demonstrating notable resilience to mobility-induced uncertainties. In terms of task completion time, BLSQ is consistently better than other algorithms, maintaining a low and stable completion time around 0.196 seconds across all intervals. QBMR shows moderate variations 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 BLSQ 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.



(a) Network Capacity and Average Throughput. (b) Interference Power and Energy Consumption. (c) Task Completion Time and QoS.

Fig. 9: Performance Comparison under UD Mobility with Periodic Location Updates.

VI. Conclusion

This paper proposed a novel three-stage optimization scheme named BLSQ, designed to address the key challenges of scalability, interference, and energy efficiency in MWMNs. The proposed BLSQ 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 BLSQ 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 BLSQ scheme presents a scalable and flexible solution for next-generation wireless networks. Future work will explore extending the learning models to incorporate real-time mobility prediction, task migration mechanisms, and federated intelligence across network nodes to enhance the system's robustness and responsiveness further. In addition, future studies will also consider modeling device heterogeneity, such as variations in frequency, bandwidth, to better reflect real-world network conditions and improve the adaptability of the proposed framework to diverse deployment environments.

References

[1] "IMT towards 2030 and beyond," ITU, 2023. [Online]. Available: https://www.itu.int/en/ITU-R/study-groups/rsg5/rwp5d/imt-2030/Pages/default.aspx

[2] Y. Sanjalawe, S. Fraihat, S. Al-E'Mari, M. Abualhaj, S. Makhadmeh, and E. Alzubi, "A review of 6G and AI convergence: Enhancing communication networks with artificial intelligence," *IEEE Open Journal of the Communications Society*, vol. 6, pp. 2308–2355, Mar 2025.

[3] T. Chen, J. Deng, Q. Tang, and G. Liu, "Optimization of quality of AI service in 6G native AI wireless networks," *Electronics*, vol. 12, no. 15, Aug 2023.

[4] S. Dong, J. Tang, K. Abbas, R. Hou, J. Kamruzzaman, L. Rutkowski, and R. Buyya, "Task offloading strategies for mobile edge computing: A survey," *Computer Networks*, vol. 254, p. 110791, Sep 2024.

[5] P. Peng, W. Lin, W. Wu, H. Zhang, S. Peng, Q. Wu, and K. Li, "A survey on computation offloading in edge systems: From the perspective of deep reinforcement learning approaches," *Computer Science Review*, vol. 53, p. 100656, Jun 2024.

[6] S. Dash, A. U. Khan, S. K. Swain, and B. 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)*, Bhubaneswar, India, Dec 2021, pp. 167–172.

[7] Q. Gan, G. Li, W. He, Y. Zhao, Y. Song, and C. Xu, "Delay-minimization offloading scheme in multi-server MEC networks," *IEEE Wireless Communications Letters*, vol. 12, no. 6, pp. 1071–1075, Mar 2023.

[8] T. S. Rappaport, Y. Xing, O. Kanhere, S. Ju, A. Madanayake, S. Mandal, A. Alkhateeb, and G. C. Trichopoulos, "Wireless communications and applications above 100 GHz: Opportunities and challenges for 6G and beyond," *IEEE Access*, vol. 7, pp. 78 729–78 757, Jun 2019.

[9] C. Han, Y. Wang, Y. Li, Y. Chen, N. A. Abbasi, T. Kürner, and A. F. Molisch, "Terahertz wireless channels: A holistic survey on measurement, modeling, and analysis," *IEEE Communications Surveys and Tutorials*, vol. 24, no. 3, pp. 1670–1707, Jun 2022.

[10] F. S. Shaikh and R. Wismüller, "Routing in multi-hop cellular device-to-device networks: A survey," *IEEE Communications Surveys and Tutorials*, vol. 20, no. 4, pp. 2622–2657, Jun 2018.

[11] N. Todtenberg and R. Kraemer, "A survey on bluetooth multi-hop networks," *Ad Hoc Networks*, vol. 93, p. 101922, Jun 2019.

[12] Z. Yao, W. Cheng, W. Zhang, T. Zhang, and H. Zhang, "The rise of UAV fleet technologies for emergency wireless communications in harsh environments," *IEEE Network*, vol. 36, no. 4, pp. 28–37, Oct 2022.

[13] I. Chandran and K. Vipin, "Multi-UAV networks for disaster monitoring: challenges and opportunities from a network perspective," *Drone Systems and Applications*, vol. 12, pp. 1–28, Jul 2024.

[14] X. Wang and X. Wang, "Reinforcement learning-based multihop relaying: A decentralized q-learning approach," *Entropy*, vol. 23, no. 10, Oct 2021.

[15] X. Li, J. Zhang, and C. Pan, "Federated deep reinforcement learning for energy-efficient edge computing offloading and resource allocation in industrial internet," *Applied Sciences*, vol. 13, no. 11, 2023.

[16] C. L. Philip Chen and Z. Liu, "Broad learning system: An effective and efficient incremental learning system without the need for deep architecture," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 1, pp. 10–24, Jul 2018.

[17] J. Chi, Z. Cui, and Y. Lim, "BLSO: Broad learning system-based scheme for adaptive task offloading in industrial IoT," in *Proceedings of International Conference on Mobile Computing and Ubiquitous Networking (ICMU)*, Kyoto, Japan, Nov 2023, pp. 1–6.

[18] Y. Zheng, B. Chen, S. Wang, and W. Wang, "Broad learning system based on maximum correntropy criterion," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 7, pp. 3083–3097, Jul 2021.

[19] X. Gong, T. Zhang, C. L. Philip Chen, and Z. Liu, "Research review for broad learning system: Algorithms, theory, and applications," *IEEE Transactions on Cybernetics*, vol. 52, no. 9, pp. 8922–8950, Sep 2022.

[20] S. Wang, M. Chen, X. Liu, C. Yin, S. 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*, vol. 8, no. 3, pp. 1358–1372, Feb 2021.

[21] Y. Chen, Y. Sun, C. Wang, and T. Taleb, "Dynamic task allocation and service migration in edge-cloud IoT system based on deep reinforcement

learning," *IEEE Internet of Things Journal*, vol. 9, no. 18, pp. 16 742–16 757, Sep 2022.
[22] Y. Chen, Y. Sun, H. Yu, and T. 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*, vol. 11, no. 4, pp. 3479–3494, Jul 2024.
[23] Z. Lin, S. Bi, and Y.-J. A. Zhang, "Optimizing AI service placement and resource allocation in mobile edge intelligence systems," *IEEE Transactions on Wireless Communications*, vol. 20, no. 11, pp. 7257–7271, Nov 2021.
[24] S. Goudarzi, S. Ahmad Soleymani, M. Hossein Anisi, A. Jindal, and P. Xiao, "Optimizing UAV-assisted vehicular edge computing with age of information: An sac-based solution," *IEEE Internet of Things Journal*, vol. 12, no. 5, pp. 4555–4569, Mar 2025.
[25] D.-H. Tran, V.-D. Nguyen, S. Chatzinotas, T. X. Vu, and B. Ottersten, "UAV relay-assisted emergency communications in IoT networks: Resource allocation and trajectory optimization," *IEEE Transactions on Wireless Communications*, vol. 21, no. 3, pp. 1621–1637, Mar 2022.
[26] M. Ahmed, S. Raza, H. Ahmad, W. U. Khan, F. Xu, and K. Rabie, "Deep reinforcement learning approach for multi-hop task offloading in vehicular edge computing," *Engineering Science and Technology, an International Journal*, vol. 59, p. 101854, Oct 2024.
[27] N. T. Hoa, D. Van Dai, L. H. Lan, N. C. Luong, D. Van Le, and D. Niyato, "Deep reinforcement learning for multi-hop offloading in UAV-assisted edge computing," *IEEE Transactions on Vehicular Technology*, vol. 72, no. 12, pp. 16 917–16 922, Dec 2023.
[28] W. Zhao, T. Weng, Y. Cheng, Z. Liu, and N. Kato, "Deep reinforcement learning for optimizing multi-hop distributed collaborative task offloading in R2X," *IEEE Transactions on Vehicular Technology*, pp. 1–15, Jan 2025.
[29] X. Chen, G. Zhu, Y. Deng, and Y. Fang, "Federated learning over multi-hop wireless networks with in-network aggregation," *IEEE Transactions on Wireless Communications*, vol. 21, no. 6, pp. 4622–4634, Jun 2022.
[30] M. Ji, Q. Wu, P. Fan, N. Cheng, W. Chen, J. Wang, and K. B. Letaief, "Graph neural networks and deep reinforcement learning-based resource allocation for V2X communications," *IEEE Internet of Things Journal*, vol. 12, no. 4, pp. 3613–3628, Feb 2025.
[31] O. Akyıldız, F. Y. Okay, İbrahim Kök, and S. Özdemir, "Mx-TORU: Location-aware multi-hop task offloading and resource optimization protocol for connected vehicle networks," *Computer Networks*, vol. 259, p. 111094, Feb 2025.
[32] B. Fan, Z. He, Y. Wu, J. He, Y. Chen, and L. Jiang, "Deep learning empowered traffic offloading in intelligent software defined cellular V2X networks," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 11, pp. 13 328–13 340, Sep 2020.
[33] H. Li, K. Ota, and M. Dong, "Learning IoT in edge: Deep learning for the internet of things with edge computing," *IEEE Network*, vol. 32, no. 1, pp. 96–101, Jan 2018.
[34] M. Xu, M. Han, C. L. Philip Chen, and T. Qiu, "Recurrent broad learning systems for time series prediction," *IEEE Transactions on Cybernetics*, vol. 50, no. 4, pp. 1405–1417, Apr 2020.
[35] Z. Cui, Y. Lim, and Y. Tan, "Factor graph-based deep reinforcement learning for path selection scheme in full-duplex wireless multihop networks," *Ad Hoc Networks*, vol. 161, p. 103542, May 2024.
[36] R. J. Wilson, *Introduction to Graph Theory*, 5th ed. New York: Prentice Hall, May 2010.
[37] S. Rezaei, M. Gharib, and A. Movaghar, "Throughput analysis of IEEE 802.11 multi-hop wireless networks with routing consideration: A general framework," *IEEE Transactions on Communications*, vol. 66, no. 11, pp. 5430–5443, Nov 2018.
[38] L. Gurobi Optimization, "Gurobi optimizer," https://www.gurobi.com/, 2025.
[39] A. T. P. Khun, H. Wang, Y. Lim, and Y. 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)*, Kuala Lumpur, Malaysia, Nov 2021, pp. 62–67.

Zhihan Cui received the B.E. degree in Electronic Information Engineering from Dalian Maritime University, China, in 2021, and the M.E. degree in Information Science from the Japan Advanced Institute of Science and Technology (JAIST), Japan, in 2023. He is currently pursuing the Ph.D. degree at JAIST. His research interests include wireless multihop networks, edge computing, and artificial intelligence-based network optimization.

Yan Chen received the B.E. degree in Information Engineering and the Ph.D. degree in Information and Communication Engineering from China University of Mining and Technology, China, in 2016 and 2022. He is currently a Postdoctoral Researcher with Ruhr University Bochum, Bochum, Germany. He is also a senior researcher with the ICTFICIAL OY, Espoo, Finland. From December 2020 to December 2021, he was also a visiting Ph.D. student at Aalto University, Finland. His research interests include Edge Computing, Internet of Things, Network Intelligence, and Multi-agent systems.

Yuto Lim received the B.Eng. (Hons) and M.Inf.Tech. degrees from University Malaysia Sarawak (UNIMAS), Malaysia, in 1998 and 2000, respectively. He received the Ph.D. degree in Communications and Computer Engineering from Kyoto University, Japan, in 2005. In November 2005, he joined the National Institute of Information and Communications Technology (NICT), Japan, as an Expert Researcher, where he worked until September 2009. Since October 2009, he has been with the Japan Advanced Institute of Science and Technology (JAIST), where he served as an Associate Professor. In April 2025, he was appointed Full Professor at JAIST. His research interests include quantum networks, future wireless communications and networks, wireless network coding, smart homes, smart cities, cyber-physical systems, the Internet of Things (IoT), and smart energy distribution. He is a member of the IEEE, IEICE, and IPSJ.

Tarik Taleb (Senior Member, IEEE) received the B.E. degree (with distinction) in information engineering and the M.Sc. and Ph.D. degrees in information sciences from Tohoku University, Sendai, Japan, in 2001, 2003, and 2005, respectively. He is currently a Full Professor at Ruhr University Bochum, Germany. He was a Professor with the Center of Wireless Communications, University of Oulu, Oulu, Finland. He is the founder of ICTFICIAL Oy, and the founder and the Director of the MOSA!C Lab. From October 2014 to December 2021, he was an Associate Professor with the School of Electrical Engineering, Aalto University, Espoo, Finland. Prior to that, he was working as a Senior Researcher and a 3GPP Standards Expert with NEC Europe Ltd., Heidelberg, Germany. Before joining NEC and till March 2009, he worked as an Assistant Professor with the Graduate School of Information Sciences, Tohoku University, in a lab fully funded by KDDI. From 2005 to 2006, he was a Research Fellow with the Intelligent Cosmos Research Institute, Sendai. Taleb has been directly engaged in the development and standardization of the Evolved Packet System as a member of the 3GPP System Architecture Working Group. His current research interests include AI-based network management, architectural enhancements to mobile core networks, network softwarization and slicing, mobile cloud networking, network function virtualization, software-defined networking, software-defined security, and mobile multimedia streaming.