HWDSQP: A Historical Weighted and Dynamic Scheduling Quantum Protocol to Enhance Communication Reliability

Liwei Lin, Rongbo Ma, Zejian Wang, Zinuo Cai, Haochen Xu, Baoheng Zhang, Ruhui Ma, *Member, IEEE*, and Rajkumar Buyya, *Fellow, IEEE*

***Abstract*—Quantum computing holds the promise of solving problems difficult for classical computers. However, we are still in the era of Noisy Intermediate-Scale Quantum (NISQ) computers, necessary to establish effective distributed quantum communication protocols to distribute complex quantum computing tasks across different quantum computers for execution. Significant progress has been made in quantum communication technology, particularly in quantum path creation and resource scheduling. The establishment of quantum paths relies on quantum entanglement and quantum relay technologies, achieving long-distance, high-fidelity quantum state transmission through entanglement swapping between multiple relays. However, resources in quantum communication networks are limited and expensive, making efficient resource scheduling strategies crucial for improving overall network efficiency. To address these issues, we design a network protocol that includes the Historical Weighted Fidelity Routing (HWFR) algorithm and the Dynamic Multi-Priority Quantum Scheduling (DMPQS) algorithm to enhance communication reliability across quantum computers. Both algorithms aim to enhance the reliability of quantum links, optimize resource utilization, and adapt to dynamic changes in the links. The former algorithm dynamically selects the optimal path by considering factors such as link length, noise level, entanglement success rate, and quantum relay resource constraints, ensuring high-fidelity and reliable quantum communication. The latter dynamically adjusts request priorities based on the urgency of quantum service requests and fidelity requirements, optimizing resource utilization. Experimental results show that the proposed protocol performs excellently in terms of an average response time of requests and link utilization, effectively improving the utilization efficiency of network resources and the overall performance of the system.**

***Index Terms*—Quantum computing, routing, network resource scheduling, communication reliability.**

I. Introduction

QUANTUM computing [1], [2], [3], due to its powerful computational capabilities, holds the promise of solving problems that are difficult for classical computers, and has potential applications in fields such as cryptography [4] and deep learning [5]. However, due to the limitations of current quantum hardware, we remain in the era of Noisy Intermediate-Scale Quantum (NISQ) computers [6]. Although existing quantum devices have reached a medium scale and are capable of performing meaningful computational tasks, they are still unable to fully eliminate noise and errors. To overcome the limitations of NISQ devices, Distributed Quantum Computing (DQC) systems [7] have emerged to allow complex computational tasks distributed across geographically dispersed quantum computers. Such distributed computation is built on the foundation of quantum communication networks [8], [9], enabling the transfer of qubits from the source computer to the destination computer through the network topology. Therefore, finding suitable quantum data networks is crucial and essential for distributed quantum computing systems in the NISQ era to establish reliable communication paths between geographically isolated quantum computers. To support quantum data networks for distributed quantum computing, significant advancements have been made in quantum communication technology in recent years, particularly in the areas of quantum path creation and resource scheduling. The establishment of quantum paths relies on quantum entanglement [10], [11], [12], [13] and quantum repeater [14], [15], [16] technologies. By performing entanglement swapping between multiple repeaters, long-distance, high-fidelity quantum state transmission can be achieved. The creation of quantum paths is fundamental to stable quantum communication. Researchers have made breakthroughs in quantum key distribution (QKD) and the development of quantum repeaters, making the construction of quantum communication networks possible [17], [18], [19], [20]. Resources in quantum communication networks are limited and expensive, making efficient resource scheduling strategies crucial

Received 1 August 2024; revised 10 December 2024; accepted 11 January 2025. Date of publication 8 May 2025; date of current version 4 September 2025. This work was supported in part by the Educational Scientific Research Project of Fujian Provincial Department of Education under Grant JAT210291, in part by the Science Foundation of Fujian University of Technology under Grant GY-Z220206, and in part by the Eighth Research Institute of China Aerospace Science and Technology Group Company Ltd. under Grant USCAST2023-17 and Grant USCAST2023-21. *(Liwei Lin and Rongbo Ma are co-first authors.) (Corresponding author: Ruhui Ma.)*

Liwei Lin is with the School of Computer Science and Mathematics, Fujian University of Technology, Fuzhou 350118, China (e-mail: llw02@fjut.edu.cn).

Rongbo Ma, Zejian Wang, Zinuo Cai, Haochen Xu, and Ruhui Ma are with the School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: marongbo@sjtu.edu.cn; zejianwang@sjtu.edu.cn; xhc12138@sjtu.edu.cn; kingczn1314@sjtu.edu.cn; ruhuima@sjtu.edu.cn).

Baoheng Zhang is with Aerospace System Engineering Shanghai, Shanghai 201109, China (e-mail: beowulfbhz@126.com).

Rajkumar Buyya is with the School of Computing and Information Systems, The University of Melbourne, Melbourne, VIC 3010, Australia (e-mail: rbuyya@unimelb.edu.au).

Digital Object Identifier 10.1109/JSAC.2025.3568051

for improving overall network efficiency. The core of optimizing scheduling strategies lies in the rational allocation and utilization of these limited quantum resources to maximize network throughput and ensure high-fidelity quantum state transmission. Resource scheduling is key to optimizing the performance of quantum communication networks. In recent years, researchers have developed various scheduling algorithms, such as priority-based scheduling and two-phase traversal [21], which dynamically adjust resource allocation based on different quantum service requests, significantly enhancing the performance of quantum networks.

Although significant progress has been made in the creation of quantum paths, there are still many challenges that are particularly relevant to our research questions. Firstly, the reliability of quantum links poses a significant challenge. Quantum links are affected by environmental noise, quantum state decay, and entanglement generation failures, leading to reduced fidelity in quantum communication. Current routing algorithms, such as Dijkstra's shortest path first (SPF) algorithm [22], do not fully account for these characteristics of quantum links, resulting in potentially unreliable and suboptimal path selections. These factors directly impact the integrity and effectiveness of quantum states during transmission. Therefore, it is essential to comprehensively consider quantum link length, noise [23], entanglement success rate, and quantum repeater resource constraints. This ensures that path selection is more reliable and efficient in practical operations, adapts to various metrics of quantum links, and handles the dynamic allocation of quantum repeater resources, thereby enhancing the overall performance and reliability of quantum communication.

Secondly, the throughput of quantum links is another important research focus. Quantum network resources are scarce and expensive [23], [24], [25], [26], making it crucial to enhance link throughput to improve overall network efficiency under limited resource conditions. However, traditional scheduling algorithms often fail to fully utilize these resources, leading to resource wastage and low network efficiency. The current challenge lies in how to fully utilize network resources to serve as many quantum service requests as possible with high fidelity and low latency.

Therefore, we design `HWDSQP` to address the issues of quantum link reliability and link throughput. `HWDSQP` comprehensively considers quantum link length, noise, entanglement success rate, and quantum repeater resource constraints to select the optimal path for qubits. Simultaneously, based on the urgency, importance, and requirements for fidelity and latency of quantum service requests, it dynamically adjusts the scheduling of service requests to optimize resource utilization.

To be more specific, we propose Historical Weighted Fidelity Routing (HWFR) Algorithm to resolve the first challenge. The algorithm addresses the reliability and efficiency issues of quantum links. When selecting paths, it not only considers the length of the link but also evaluates the noise level, entanglement success rate, and resource constraints of quantum repeaters. By dynamically assigning weights to these metrics, the algorithm can choose the optimal path for each qubit, ensuring high fidelity and reliability in quantum communication. Additionally, the algorithm leverages historical information, optimizing path selection based on past choices and performance data to enhance accuracy and stability. Through real-time adjustments, the algorithm can adapt to the ever-changing link conditions and resource status in quantum networks, ensuring the continuity and stability of quantum communication.

For the second challenge, we design Dynamic Multi-Priority Quantum Scheduling (DMPQS) algorithm to solve it. The algorithm maximizes overall network efficiency and reliability by effectively managing and allocating the limited resources of the quantum network. Based on a multi-priority feedback mechanism, it dynamically adjusts the priority of service requests according to their urgency, resource usage, and other requirements. By categorizing requests into different priority queues and employing various scheduling strategies, the algorithm can effectively reduce resource wastage and enhance resource utilization. Additionally, the algorithm includes a retry mechanism that appropriately retries unsuccessful requests based on the failure reasons and current network conditions. This ensures that as many service requests as possible are completed within the limits of network resources. The algorithm also ensures fairness in request processing, reduces average waiting time, and further improves the overall throughput and reliability of the network.

To testify the performance of the Dynamic Multi-Priority Quantum Scheduling (DMPQS) algorithm, we structure our experiment to verify its efficiency and robustness across three critical dimensions. We assess the average response time, observing a notable 30% improvement over the Traditional FIFO method. We also measure the link utilization rate, which demonstrates a significant increase, reflecting optimized resource use, which also shows a 15% improvement using DMPQS algorithm. Furthermore, we analyze the state of requests, focusing on the completion, delay, and abandonment conditions, to evaluate the algorithm's robustness. The DMPQS algorithmconsistently outperforms the Traditional FIFO, showcasing its superior capabilities in reducing response times, enhancing link utilization, and optimizing request management with less abandonments. These results collectively confirm the DMPQS algorithm as an efficient and robust solution for network scheduling, adept at maintaining high performance under diverse network conditions.

In summary, our contributions are highlighted as follows.

- We design an efficient quantum communication network protocol that comprehensively considers the reliability, fidelity, and resource utilization of quantum links. This protocol dynamically adapts to changing conditions in the quantum network, ensuring continuous, stable, and efficient communication.
- We propose the Historical Weighted Fidelity Routing Algorithm which considers link length, noise level, entanglement success rate, and historical cost when selecting paths, enabling it to choose the optimal path for qubits and ensure high-fidelity and reliable quantum communication.
- We develop the Dynamic Multi-Priority Quantum algorithm that adjusts the priority of service requests based

on their urgency, importance, and latency requirements, enhancing the overall throughput and reliability of the network.

- We conduct extensive experiments in various of aspects, with the benchmark of Traditional FIFO method to validate the high-performance and robustness of the DMPQS algorithm under a randomly generated network condion.

II. Background and Motivation

A. Quantum Computing and Communication

In quantum computing, qubits are the basic units of information transmission. Unlike classical bits, qubits can be in a superposition state, representing both 0 and 1 simultaneously. Quantum computers can use algorithms like Shor's algorithm [27] to efficiently solve problems such as integer factorization and discrete logarithms in polynomial time, threatening many existing classical encryption schemes.

Quantum communication leverages fundamental principles of quantum mechanics, such as quantum entanglement and superposition, to provide fundamentally different security assurances. For example, Quantum Key Distribution (QKD) ensures the absolute security of key transmission by utilizing the no-cloning theorem and the destructive nature of measurement. Any eavesdropping attempt alters the quantum state, alerting the communicating parties.

The security of quantum communication also involves complex practical factors. Qubits are affected by environmental noise and decoherence [28], [29] during transmission, which reduces information fidelity. Creating and maintaining entangled states is crucial in quantum communication. For instance, entanglement swapping between multiple repeaters and Bell-state measurements (BSM) are used for quantum state transmission. However, these operations are not always successful and are limited by equipment precision and environmental conditions. In linear optical systems, for example, the success rate of Bell-state measurements is about 0.6 [30], and any measurement failure destroys the quantum state.

Due to the no-cloning theorem [31] of quantum information, failed transmissions result in irreversible loss of qubit information. This imposes higher reliability demands on quantum networks. If a data qubit is lost during transmission, it can jeopardize the entire distributed quantum computing (DQC) process. Therefore, quantum communication systems require higher transmission reliability and more complex error correction mechanisms than traditional systems to ensure the correct transmission and processing of quantum information.

B. Quantum Characteristics

1) Fidelity: Fidelity is a measure of how well a quantum state maintains its initial state during transmission. Its value ranges from 0 to 1, quantifying the quality of the state based on its "closeness" to the desired state (a fidelity of 1 indicates the state is exactly as desired, while a value below 0.5 indicates the state is no longer usable). Quantum applications can operate with imperfect quantum states, as long as the fidelity is above a specific threshold for the application (for basic QKD, this threshold is around 0.8). Higher fidelity means less interference and noise affecting the quantum state during transmission. Specifically, fidelity can be defined as [32]:

$$F = \langle \Psi^+ | \rho | \Psi^+ \rangle, \tag{1}$$

where ρ is the density matrix representation of the state, and $|\Psi^+\rangle$ is the state we aim to create. To ensure successful key exchange, a fidelity $F > 0.9$ is typically required for practical quantum applications, such as QKD [33].

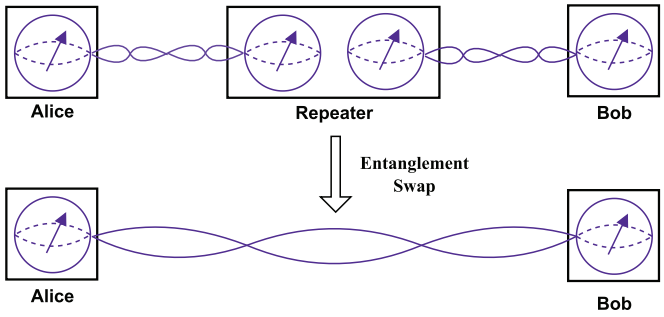


Fig. 1. Alice and the repeater, as well as the repeater and Bob, each share an entangled pair. Alice can use her entangled pair to transmit a data qubit to Bob via teleportation. This process consumes the entangled pair, enabling the long-distance transfer of the quantum state.

2) Decoherence: Decoherence is the phenomenon where the quality of qubits deteriorates over time, leading to a decrease in the fidelity of quantum states. Decoherence is one of the key challenges in quantum networks because it significantly limits the time qubits can be stored in memory before being used. In current experimental hardware, the decoherence time of qubits is approximately a few milliseconds, but in devices disconnected from the network, qubit storage time can extend up to a minute.

3) Entanglement Swapping: Entanglement swapping is a core concept in quantum communication. Due to the no-cloning theorem, decoherence, and transmission loss, the distribution of quantum states is highly limited if amplification or retransmission cannot be used. To address this problem, Briegel et al. propose a quantum repeater scheme in 1998 [34], which connects a series of short-distance entangled qubit pairs through a process known as entanglement swapping, thereby generating long-distance entanglement. As illustrated in Fig. 1, Alice and the repeater share an entangled pair, and similarly, the repeater and Bob share another entangled pair. By performing a Bell-state Measurement (BSM) [35] at the repeater, these two pairs of entangled states can be "swapped" between Alice and Bob, achieving long-distance entanglement without direct quantum connection.

Considering the imperfect quantum operations at each node, the fidelity of the newly established entangled pair can be expressed as:

$$F_{AC} = F_{AB} \cdot F_{BC}, \tag{2}$$

where F_{AB} and F_{BC} represent the fidelities of the two short-distance entangled pairs, respectively, and F_{AC} represents the fidelity of the newly established long-distance entangled pair.

Additionally, for multiple entanglement swapping operations, the final fidelity can be calculated using the following formula:

$$F_n = \prod_{i=1}^{n-1} F_{i,i+1}, \tag{3}$$

where n represents the number of hops in the path, and $F_{i,i+1}$ denotes the fidelity of the entangled pair at the hop i. Therefore, performing entanglement swapping results in a longer-distance entangled pair with reduced fidelity [36], [37].

Although the underlying physical processes are quantum, quantum networks require classical connections between all quantum nodes to exchange control messages. Entanglement swapping necessitates that intermediate nodes send messages to at least one other node to make the entanglement useful. Moreover, quantum networks, similar to classical networks, require control and management protocols that use classical channels for communication.

C. Motivation

In recent years, quantum communication technology has made significant advancements, driving the transition of quantum networks from laboratory research to practical deployment. As the first intercity quantum network is about to go online, a new challenge arises: how to effectively build and manage large-scale quantum communication systems. Although existing research has proposed quantum network stacks and link layer protocols to provide robust entanglement generation services for directly connected nodes [38], these solutions primarily focus on short-distance communication. Achieving widespread application of quantum communication requires developing an efficient quantum network layer protocol capable of providing long-distance end-to-end entanglement between any nodes in the network.

The design and optimization of quantum communication network protocols need to consider multiple factors. First, the reliability of quantum links is one of the primary considerations. The fragility of quantum states and interference from environmental noise can lead to decreased fidelity and entanglement failure. Traditional routing algorithms do not fully account for these characteristics, potentially choosing less reliable paths. Resources in quantum communication networks, such as quantum repeaters and quantum memory slots, are limited and expensive. Therefore, efficient utilization of these resources is essential to maximize the network's throughput and service capacity. Traditional scheduling algorithms often fail to make full use of available resources, resulting in resource wastage and low network efficiency. Additionally, the dynamic variability of quantum links necessitates protocols with real-time adjustment and dynamic adaptation capabilities to ensure that communication requests are processed promptly and effectively, thereby enhancing communication reliability and efficiency.

To address these issues, our goal is to design a network protocol that improves the reliability of quantum links, optimizes resource utilization, and adapts to the dynamic changes of links. Through this optimization, we aim to achieve more efficient and stable quantum communication, meeting the needs of practical applications and promoting the widespread adoption and development of quantum communication technology.

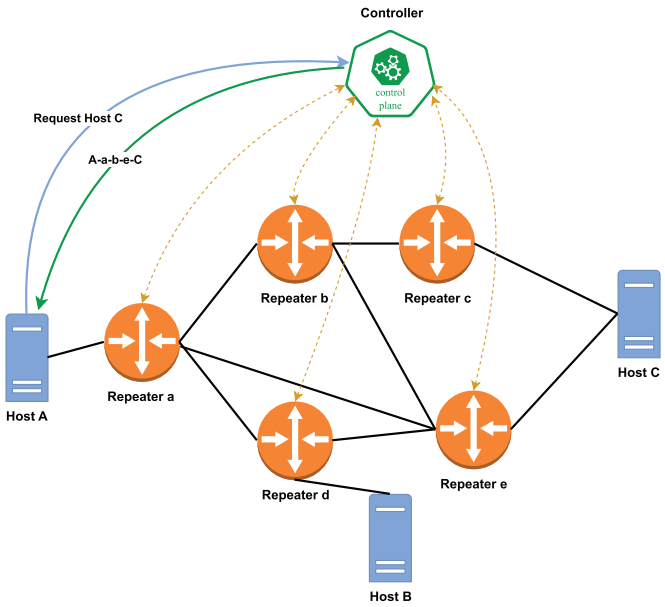


Fig. 2. Quantum network architecture.

III. Design

In this section, we first introduce the architecture of the quantum network and time slot. And then provide a detailed introduction to the two algorithms.

A. Quantum Network Architecture

The architecture of a quantum network is illustrated in Fig. 2 It consists of the following components.

- **Quantum Computer**: The terminal device responsible for initiating and receiving communication requests, aiming to create end-to-end entanglement through the quantum communication network, enabling the exchange and processing of quantum states.
- **Quantum Repeater**: A device that transmits and amplifies quantum signals, connecting other repeaters or quantum computers [39], [40], [41]. Using entanglement swapping and purification techniques [42], [43], repeaters can extend the transmission distance of quantum states, ensuring that quantum information maintains high fidelity over long-distance transmissions.
- **Quantum Network Controller**: The network control plane, typically a classical computer, connects all repeaters and quantum computers, managing and coordinating the resources of the entire quantum communication network [21].

Our algorithm primarily optimizes the quantum network controller. The controller receives end-to-end entanglement requests from quantum computers in the network and returns the path information to these computers once the operation is completed. The controller also receives local link entanglement results, including both successful and failed attempts, from repeaters within the network. Based on this information,

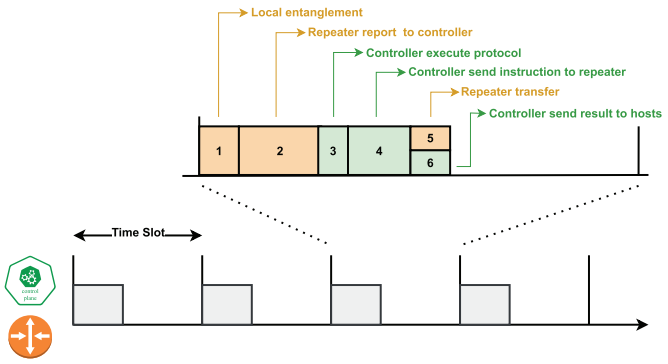


Fig. 3. Time slot structure.

the controller selects the optimal path for each request and performs path allocation according to the algorithm. The controller then instructs intermediate repeaters to perform the entanglement swapping operations on the designated links to activate the end-to-end entanglement on the selected path.

B. Time Slot

In quantum communication networks, the introduction of time slots is used to synchronize the operations of all nodes, ensuring the effective transmission and processing of quantum states. The duration of each time slot is set based on the technical characteristics of the specific equipment and configured to an appropriate length by the link layer. Within each time slot, only one of three quantum operations is allowed: entanglement generation, entanglement swapping, or entanglement purification. The scheduling algorithm determines which entangled pairs in the repeater need to be swapped or purified in each time slot. Meanwhile, new entangled pairs are generated by the quantum source, transmitted through quantum channels, and stored in the repeaters to supplement resources.

In our paper, time slots are used to coordinate quantum link operations and resource scheduling. Our algorithm is designed to optimize operations within each time slot. By carefully designing the time slot length and scheduling strategy, we ensure efficient resource utilization and high-fidelity quantum state transmission.

The time slot structure is illustrated in Fig. 3. We now describe all the subphases in a time slot.

1) The repeater performs local entanglement operations. Assuming the existence of synchronized quantum sources, when the time slot begins, all synchronized quantum sources emit a pair of entangled qubits. These qubits are received by the nodes, stored in the nodes' quantum memories, and checked for success. Successful entangled qubits are marked as available for subsequent quantum information transmission, while failed qubits are discarded.
2) The repeater feeds the results of the local entanglement operations back to the quantum network controller. These results include whether entangled pairs were successfully generated and related costs, allowing the controller to fully understand the current state of the network.

Algorithm 1 HWFR Algorithm

Initialize quantum resources and request queues $\mathcal{Q}_a, \mathcal{Q}_b, \mathcal{Q}_c$
Initialize network topology and set historical cost to an initial average cost
Pending set $\mathcal{S}$, Requests set $\mathcal{R}$
for each $(\text{repeater}, \text{neighbor}, \text{cost})$ in links **do**
    Report link between repeater and neighbor with cost
end for
Update network topology and link status
if $\mathcal{S} \neq \emptyset$ **then**
    for $r \in \mathcal{S}$ **do**
        $\sigma \leftarrow \text{PathSelection}(r)$
        $\mathcal{Q} \leftarrow \sigma$
    end for
    $\mathcal{S} \leftarrow \emptyset$
end if
for $r \in \mathcal{R}$ **do**
    $\sigma, \text{min\_cost} \leftarrow \text{PathSelection}(r)$
    if $\text{min\_cost} < \text{historical\_cost}$ **then**
        $\mathcal{Q}_b \leftarrow \sigma$
    else
        $\mathcal{Q}_c \leftarrow \sigma$
    end if
end for
Update historical_cost

3) The controller executes appropriate protocols based on the received reports, such as routing protocols and scheduling protocols. The controller makes optimal decisions to maintain the stability of quantum communication and ensure the effective transmission of quantum information.
4) The controller sends scheduling information to the repeater, including a list of pairs of quantum storage node locations for performing entanglement swapping, to create end-to-end entanglement.
5) Each intermediate node forwards the projection measurement results (classical information) for obtaining entanglement swapping to an endpoint node in the path. These results are assumed to be forwarded directly by the intermediate nodes without involving the controller. In this way, the endpoint nodes can perform corresponding quantum state corrections based on the received measurement results, ensuring accurate transmission of the entangled state.
6) The controller sends the final quantum communication status to the host node, completing the entire quantum information transmission process.

C. Historical Weighted Fidelity Routing (HWFR) Algorithm

In quantum computing and communication networks, the design of routing algorithms is crucial for ensuring efficient data transmission and resource allocation. We design HWFR Algorithm 1 aimed at reducing communication costs and improving network performance by optimizing the network topology and path selection.

The algorithm begins by initializing quantum resources and the request queue and setting up the network topology (Lines 1-2). For each link in the network, the algorithm reports the link costs between repeaters and neighbors, updating the network topology and link status accordingly (Lines 4-6).

First, the algorithm processes requests that were not successfully scheduled in the previous time slice, selecting paths for each request. If path selection is successful, the request is placed back into its original priority queue, and the algorithm proceeds to handle the remaining requests (Lines 9-12). For each request in the newly arrived request set, the algorithm selects a path and calculates the minimum cost. If the minimum cost is lower than the historical cost, the request is placed in a higher priority queue; otherwise, it is placed in a lower priority queue (Lines 15-22).

In this process, the path selection algorithm is a critical component, considering multiple metrics to choose the optimal path. Link cost is a critical metric that represents the transmission overhead between a repeater and its neighbor. In quantum communication networks, we primarily consider the total length and fidelity of the transmission path. The calculation formula is as follows:

$$\mathit{Cost} = (\mathit{distance})^{\alpha} \times (1 - \mathit{fidelity})^{\beta}, \tag{4}$$

where distance represents the physical distance between two nodes, fidelity represents the fidelity of the link, and α and β are weight parameters that adjust the importance of distance and fidelity.

Besides, node hop count is another key factor, indicating the number of repeaters a signal passes through from the source node to the target node. Fewer hops can reduce the delay and loss caused by repeaters during transmission, typically resulting in higher reliability. Therefore, the path selection algorithm tends to select paths with fewer hops.

Additionally, network load is another important factor, referring to the current communication traffic on the path. To avoid excessive load on certain links or nodes, the algorithm selects paths with lower loads for transmission. When a link has been used in previous routing decisions, the algorithm prioritizes unallocated paths with the same cost, effectively distributing network traffic, preventing single points of overload, and enhancing overall network performance.

For a path P,the optimization objectives are to maximize the transmission efficiency of the path and minimize the transmission cost:

$$\min \sum_{(i,j)\in P} \left(\mathit{distance}_{ij}^{\alpha} \times (1 - \mathit{fidelity}_{ij})^{\beta}\right), \tag{5}$$

$$\text{subject to:} \quad f_{\sigma} \geq f_{\min}, \tag{6}$$

$$|P| \leq H_{\max}, \tag{7}$$

$$\sum_{i\in V(P)} q_i \leq Q_{\max}, \tag{8}$$

where f_{σ} defined as the product of the fidelities of all edges in the path:

$$f_{\sigma} = \prod_{e\in\sigma} f_e,$$

$f_{\min}$ is the acceptable minimum fidelity requirement for the application:

$$\sigma^{*} = \arg \min_{\sigma\in\Sigma, f_{\sigma}\geq f_{\min}} \mathrm{Cost}(\sigma),$$

$|P|$ is the number of hops in the path, $H_{\max}$ is the maximum allowable number of hops for the path, q_i is the quantum bit usage at node i, and $Q_{\max}$ is the total capacity of the node.

For each path σ, its total cost function $\mathrm{Cost}(\sigma)$ is a linear combination of the link costs $C_e = d_e^{\alpha}\cdot(1-f_e)^{\beta}$. Since $d_e^{\alpha} > 0$ and $(1-f_e)^{\beta} > 0$, the cost function $\mathrm{Cost}(\sigma)$ is a non-negative function.

Additionally, the key influencing factors of link cost, d_e and $(1 - f_e)$, involve power operations that are monotonic over their domains. Therefore, under a fixed network topology, the solution space of the problem exhibits convexity, ensuring the existence and uniqueness of an optimal solution.

If the fidelity of each link e satisfies $f_e > 0.5$, the lower bound of the total fidelity f_{σ} is:

$$f_{\sigma} \geq (0.5)^{k},$$

where $k = |\sigma|$ is the number of links in the path. Therefore, by limiting the path length k, it is possible to ensure that the total fidelity f_{σ} meets the application requirements $f_{\min} > 0$.

The core design of the HWFR algorithm lies in the combination of historical information and real-time link status to ensure that path selection is both stable and dynamically adaptable. During the path selection process, the HWFR algorithm employs a dynamic updating mechanism that adjusts the path cost function based on the latest link reports (including fidelity, noise level, link latency, etc.) in each round, while also weighting historical performance data to form a comprehensive cost model. This design allows the algorithm to quickly respond to fluctuations in link conditions. For instance, when a link's fidelity decreases due to increased noise, the weight of the real-time link status is increased, thereby reducing the priority of that link in path selection.

This dynamic adjustment ensures that the algorithm does not solely rely on historical data but can flexibly optimize path selection based on the real-time environment. Therefore, the HWFR algorithm can adapt to the challenges of topology changes and link status fluctuations in quantum networks, ensuring the reliability and efficiency of path selection, while also taking into account the long-term performance reference provided by historical data, achieving a balance between stability and flexibility.

D. Dynamic Multi-Priority Quantum Scheduling (DMPQS) Algorithm

In the quantum computing environment, effective scheduling of service requests is crucial for optimizing resource utilization and improving system response speed. To address this, we design DMPQS algorithm in Algorithm 2. This algorithm aims to maximize resource utilization and system response speed by appropriately allocating quantum computing resources within each time slice.

The algorithm begins by initializing the request queue and sets up three sets to store deferred requests, completed

Algorithm 2 DMPQS Algorithm

```
1: Request queues Q_a, Q_b, Q_c
2: S ← ∅, C ← ∅, F ← ∅
3: for each time slot do
4:     R ← MergeQueues(Q_a, Q_b, Q_c)
5:     while R ≠ ∅ do
6:         r ← SelectRequest(R)
7:         σ ← r.path
8:         if PathAllocation(r, σ) == successful then
9:             E ← E \ {e ∈ σ}
10:            C ← C ∪ σ
11:        else
12:            r.attempts ← r.attempts + 1
13:            if r.attempts > 3 then
14:                Discard request r
15:            else
16:                if r ∈ Q_c then
17:                    Promote r to Q_b
18:                else if r ∈ Q_b then
19:                    Promote r to Q_a
20:                end if
21:                S ← S ∪ {r}
22:            end if
23:        end if
24:        R ← R \ {r}
25:    end while
26: end for
```

requests, and failed requests, respectively. During each time slice, the algorithm sequentially merges request queues of different priorities to form a new request set. When the request set is non-empty, the algorithm processes each request. It selects a request from the set based on different strategies and attempts to allocate a path for it. If the path allocation is successful, the resources on that path are removed from the available resources collection, and the request is added to the completed requests collection (Lines 8-10). If the path allocation fails, the request's attempt count is recorded. If the attempt count exceeds three, the request is discarded, and a scheduling failure is reported; otherwise, the request is placed in the deferred requests collection. Additionally, if the request's current priority is not the highest, its priority is elevated by one level (Lines 13-22).

In the aforementioned quantum service request scheduling algorithm, the process of selecting requests significantly impacts the overall efficiency of the algorithm. We employ three different selection algorithms to address various request types and scenario requirements. These three selection algorithms are the First-In-First-Out (FIFO) algorithm, the Best Selection algorithm, and the Priority-Based Scheduling algorithm. The following sections will detail the principles and applicable scopes of these three selection algorithms.

Policy#1: FIFO. The FIFO algorithm is a simple and commonly used scheduling strategy. This algorithm processes requests in the order they arrive, giving priority to the earliest arrivals for resource allocation. Specifically, when a new service request arrives, the algorithm adds it to the end of the queue. During each scheduling cycle, the algorithm retrieves the request at the head of the queue and allocates resources to it. If the allocation is successful, the request is removed from the queue; otherwise, it is re-added to the queue to wait for the next scheduling cycle.

The advantages of the FIFO algorithm include its simplicity, low overhead, and high fairness, making it suitable for scenarios where request processing times are relatively uniform. However, the FIFO algorithm may lead to low resource utilization because some complex requests might occupy a significant amount of time, thus extending the wait time for subsequent requests. To mitigate this issue, our research introduces priority queues that categorize requests based on their complexity and urgency. This way, complex requests do not block other requests for extended periods, thereby optimizing overall resource utilization and system response time.

Policy#2: Best Selection. The Best Selection algorithm aims to process the most suitable requests based on certain optimization criteria. This algorithm typically evaluates each request using a scoring mechanism or cost function, selecting the highest-scoring or lowest-cost request for priority processing. For instance, in network routing, the Best Selection algorithm may determine the optimal path based on factors such as bandwidth, latency, or hop count.

In our algorithm, the Best Selection method evaluates requests in the waiting queue during each scheduling cycle, selecting the current optimal request for resource allocation. If the allocation is successful, the request is removed from the queue; otherwise, its score is updated and re-evaluated in the next cycle. Our research employs a multidimensional cost function, incorporating factors such as request wait time, resource consumption, and request priority to achieve more precise and efficient scheduling. The Best Selection algorithm can significantly enhance overall system performance and resource utilization, particularly when request characteristics vary widely. However, its higher computational complexity may increase scheduling overhead.

Policy#3: Random Selection. The Random Selection algorithm processes requests by randomly selecting one from the queue, thus achieving load balancing and preventing certain requests from being neglected for extended periods. In each scheduling cycle, the algorithm randomly selects a request from the waiting queue for resource allocation. If the allocation is successful, the request is removed from the queue; otherwise, it is reinserted into the queue to wait for the next scheduling cycle.

The advantages of the Random Selection algorithm include its simplicity and ability to somewhat prevent unfair distribution of resources. Due to its randomness, this algorithm can effectively prevent certain requests from experiencing prolonged wait times.

The core mechanism and performance guarantee of the algorithm are analyzed from a theoretical perspective.

The resource allocation of DMPQS adopts a dynamic optimization model, with the goal of maximizing resource utilization $U(t)$ within each time slice. Specifically, the algorithm

defines the optimization function for resource allocation as:

$$U(t) = \sum_{r_i \in \mathcal{R}_t} w(r_i) \cdot \delta(r_i, t),$$

where $w(r_i)$ represents the weight of task r_i, typically related to the task's priority, and $\delta(r_i, t)$ is an indicator function that shows whether the task has been successfully allocated resources. To ensure effective resource utilization, DMPQS must meet the following constraint condition in each time slice:

$$\sum_{r_i \in \mathcal{R}_t} r_i.\text{resource} \leq \text{Total Resource}(t),$$

which means that the total resource demand of all tasks cannot exceed the system's total resource capacity. Under this mechanism, DMPQS prioritizes high-priority tasks while avoiding resource wastage.

In terms of task completion rate, DMPQS ensures that resource allocation meets the total resource constraint while prioritizing high-priority tasks. If the total resource demand of all tasks does not exceed the system's total resources, the lower bound of the task completion rate can be expressed as:

$$P_{\text{complete}} \geq \frac{\sum_{r_i \in \mathcal{R}_t} \delta(r_i, t)}{|\mathcal{R}_t|},$$

where $|\mathcal{R}_t|$ is the total number of tasks in the current time slice. This analysis shows that DMPQS is able to complete the vast majority of tasks when resources are sufficient, and maximizes the completion rate of high-priority tasks through reasonable scheduling strategies.

DMPQS algorithm also performs excellently in optimizing task response time. Through dynamic priority adjustment, the waiting time of high-priority tasks is significantly reduced, and the average response time $\bar{T}$ is defined as:

$$\bar{T} = \frac{1}{n} \sum_{i=1}^{n} (t_i - t_i^{\text{arrive}}),$$

where t_i is the completion time of task r_i, and t_i^{arrive} is the arrival time of the task. Compared to traditional FIFO scheduling, DMPQS prioritizes the allocation of high-priority tasks, ensuring that:

$$\bar{T}_{\text{high}} < \bar{T}_{\text{FIFO}}.$$

This priority allocation strategy effectively reduces the response time of high-priority tasks, while minimizing the blocking delay of low-priority tasks, thereby improving the overall system performance.

For the time complexity of DMPQS algorithm, assume the number of requests is N and the number of paths is M. In the worst case, the merging of the priority queue is $O(N \log N)$, and the cost evaluation of path selection is $O(M)$. The overall time complexity is $O(N \log N + N \cdot M)$.

The complexity of DMPQS is better for the scalability of large-scale networks and is suitable for distributed environments.

In quantum networks, task failures may be due to poor link conditions, resource contention, or other uncertain factors. To address this, the DMPQS algorithm designe a dynamic retry mechanism to maximize task completion rates and reduce resource waste. When a task fails to allocate a path, the algorithm first analyzes the cause of failure and decides whether to retry based on conditions such as link fidelity and resource availability. If the link fidelity is below a preset threshold, the retry will wait for the link status to improve in subsequent time slots; if the failure is due to resource contention (e.g., insufficient qubit storage space or high path load), the algorithm will attempt to reallocate resources in subsequent time slots. Moreover, to avoid resource waste due to frequent retries, the algorithm sets a maximum number of retries for each request. Requests that exceed this number will be downgraded or discarded.

IV. Evaluation

In this section, we evaluate the performance of our contribution within the context of a synthetic network, which is randomly generated to include ten repeaters and an array of interconnections. Before diving into the detailed evaluation, we describe the fundamental settings and the platform utilized for this experiment. The network's random generation ensures a diverse and unbiased representation of potential real-world scenarios, thereby enhancing the validity and applicability of our findings.

A. Fundamental Settings and Platform

The results in this section were obtained using the 'NetSquid' platform, an open-source, discrete-event simulator tailored for quantum information processing. Developed in Python, its adaptability has made it a valuable asset for a variety of quantum network communication experiments. Recognizing its effectiveness, we selected 'NetSquid' as our primary experimental tool.

One of the most important building blocks in the simulations are illustrated in Fig. 4, which shows two nodes (Alice and Bob) interconnected through a combination of quantum and classical links. The lengths of these channels are randomly generated as previously described. Each node is equipped with a quantum source that emits EPR-entangled pairs to its connected nodes in certain frequency. The quantum and classical channels are configured with a propagation delay $\delta = \frac{d}{c'}$, where d is the length of one channel and c' is the propagation speed. Furthermore, the loss of transmitting qubits are added to the quantum channel with a probability of:

$$p_{loss} = 1 - \left(1 - p_{init} \cdot 10^{-\frac{\eta d}{10}}\right), \tag{9}$$

where p_{init} is the probablility the qubit is lost as soon as its generation, which occurs due to imperfections in the physical devices used for generation and entanglement. Additionally, η denotes the fiber optic attenuation coefficient, measured in decibels per kilometer (dB/km). In contrast, we consider the classical channel to be free of errors, introducing only a propagation delay introduced before. These assumptions are justified given the current capabilities of high-speed optical fiber communications, especially when the data exchanged is of a small size.

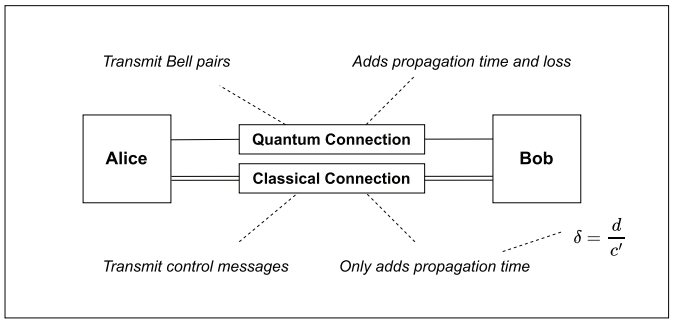


Fig. 4. The end-to-end links established in our experiment.

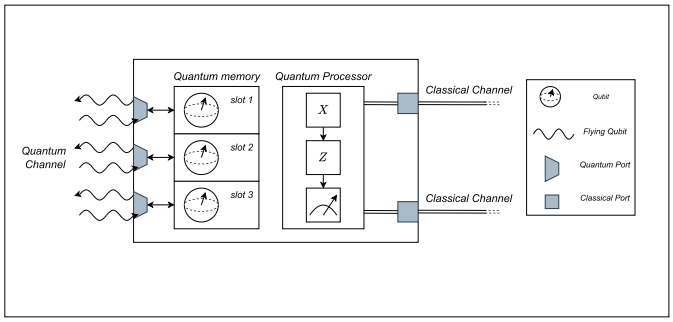


Fig. 5. The node construction using in our experiment.

Another key infrastructure in our simulation is the node. As shown in Fig. 5, each node is equipped with essential components that facilitate quantum communication: a quantum memory for storing qubits and a quantum processor to manipulate them. Quantum ports are added to the node for the purpose of reception and transmission of EPR-pairs to its adjacent nodes, and classical ports enable nodes to exchange correction bits to achieve quantum teleportation. The quantum processor processes the qubits in the quantum memory to carry out entanglement swapping. This is achieved by performing a Bell measurement on a pair of qubits and then making corrections using X and Z-gates, based on the received correction bits.

To simulate the real conditions in the pyhsical world, we integrate two distinct noise models within our node simulations. Specifically, we have applied the dephazing noise to the quantum memory, which accounts for the decoherence due to environmental interactions. The realization of dephazing noise is utilization of Pauli Z-gate stochastically with probability:

$$p_{dephase} = 1 - e^{\Delta t \cdot R_{dephase}}, \tag{10}$$

where $R_{dephase}$ is the dephasing rate (in Hz) and Δt is the duration time since the qubit is stored in the quantum memory slot. Concurrently, the X-gate and Z-gate operations in quantum processor are subjected to the depolarizing noise, which reflects the impact of random energy fluctuations on the qubit states. The depiction of depolarizing noise is achieved by stochastically applying Pauli X, Y, and Z-gates, each with the defined probability:

$$p_{depo} = 1 - e^{\Delta T \cdot R_{depo}}, \tag{11}$$

where R_{depo} denotes the depolarizing rate, measured in hertz (Hz), and ΔT represents the duration necessary for the instruction's execution.

In summary, the overview of our experiment's settings is as follows:

- *Simulation Platform*: We conduct this experiment using the Netsquid library, specifically version 1.1.7, which offers quantum information processing simulation on a classical computer. And the simulations are run in a Python 3.8.19 environment.
- *Nodes Configuration*: The synthetic network includes 10 repeaters equipped with a quantum source that emits EPR-entangled pairs at a frequency of 10 MHz and interconnected through a combination of quantum and classical links.
- *Noise Model*: We apply dephase noise model to quantum memory, accounting for decoherence due to environmental interactions. And depolarizing noise model is applied to quantum processor affecting the X-gate and Z-gate operations, reflecting the impact of random energy fluctuations on the qubit states. Both of these noise models are realized in Netsquid internally.
- *Channel Characteristics*: Both classical and quantum channel use optical fiber and the length of each channel is randomly generated from 1 km to 10 km with an initial loss rate $p_{init} = 0.05\%$. Moreover, we apply fiber noise model offered by Netsquid to each channel to configure the proper propagation speed.

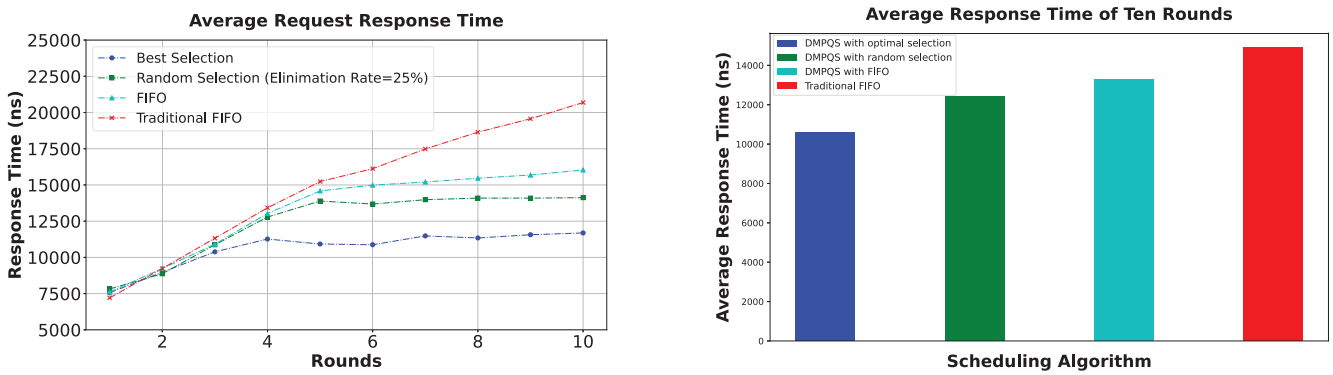
In subsequent sections, we examine the performance of our Dynamic Multi-priority Quantum Scheduling(DMPQS) algorithm in aspects of average response time of requests, requests' state and link utilization. This analysis is conducted in three standard algorithms previous described: Best selection, First-In-First-Out (FIFO), and Random selection. And tranditional FIFO will be our simulation's benchmark algorithm.

B. Performance on Average Response Time of Requests

In this section, we evaluate the performance of the DMPQS algorithm with respect to the average response time of requests. We conduct the simulations over ten rounds, with each round generating 10 new requests randomly.

Fig. 6 shows the performance comparison of different scheduling algorithms on the average response time of requests in quantum networks. We use the traditional First-In-First-Out (FIFO) algorithm as a benchmark and compared it with the three selection strategies of our proposed Dynamic Multi-Priority Quantum Scheduling (DMPQS) algorithm: random selection, FIFO, and optimal selection. Experimental results show that the DMPQS algorithm with optimal selection performs best in terms of average response time, followed by the DMPQS algorithm with random selection, the DMPQS algorithm with FIFO, and the traditional FIFO algorithm.

The experimental results indicate that scheduling strategies using the optimized algorithm significantly outperform strict FIFO without the optimized algorithm. In the first ten rounds of experiments, the DMPQS algorithm with optimal selection consistently exhibited the lowest request response time,



(a) Performance on average responce time of requests.

(b) Average response time of ten rounds.

Fig. 6. Average response time.

especially as the number of rounds increased, and its performance advantage became more pronounced. The DMPQS algorithm with random selection improved response speed by randomly eliminating some requests, thereby reducing system load. In contrast, the response times of the DMPQS algorithm with FIFO and the traditional FIFO algorithm significantly increased as the number of rounds increased.

As shown in Fig. 6, the DMPQS algorithm with optimal selection had an average response time of approximately 10605.6 nanoseconds in the first ten rounds, while the response time of the traditional FIFO reached 14891.7 nanoseconds, representing an efficiency improvement of about 30%. The average response times of the DMPQS algorithm with random selection (elimination rate = 25%) and the DMPQS algorithm with FIFO were 12420.5 nanoseconds and 13274.9 nanoseconds, respectively. This indicates that even with the use of the optimized algorithm, different scheduling strategies can still significantly impact system performance.

In conclusion, the DMPQS algorithm with best selection performs best in improving request response time, followed by the DMPQS algorithms with random selection and FIFO. Strict FIFO, due to the lack of an optimized algorithm, lags significantly in performance. These results demonstrate that adopting appropriate scheduling and selection algorithms in quantum communication networks can significantly enhance overall system performance and response speed.

C. Performance on Link Utilization Rate

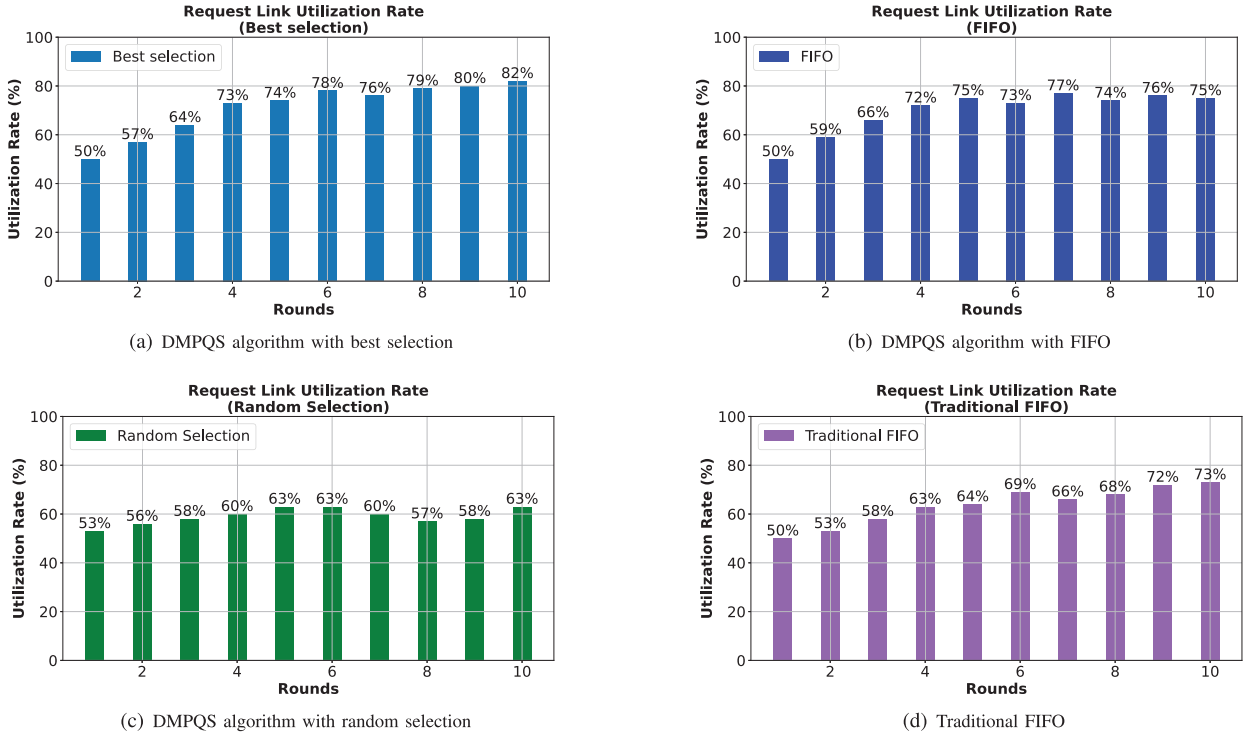
In this section, we evaluate the performance of the DMPQS algorithm with respect to link utilization rate. As described before, we conduct the simulations over ten rounds, with each round generating 10 new requests randomly. And traditional FIFO algorithm will stil serve as a benchmark and be compared with the three selection strategies of our proposed Dynamic Multi-Priority Quantum Scheduling (DMPQS) algorithm: random selection, FIFO, and optimal selection.

The experimental results denotes that scheduling strategies using the optimized algorithm outperform traditional FIFO without the optimized algorithm. The best selection strategy's link utilization rate reached 71.3%, which is an increase of 12.4% over the traditional FIFO algorithm's rate of 63.6%. However, building upon our previous findings that demonstrated the best selection strategy in the DMPQS algorithm outperformed the benchmark by a significant 29.1% reduction in average response time, which means that the DMPQS algorithm using best selection not only improves the speed of service but also enhances the efficiency of network resource use. This dual improvement is a clear indication of the effectiveness of our algorithm in managing network traffic.

The trends showing in the Fig. 7 shows that as the number of rounds increased, the performance of DMPQS algorithm with three optimized algorithm advantage became even more pronounced. Through theoretical analysis, this phenomenon can be explained by that as the number of rounds increases, the requests become a stream of continuous flow, allowing the DMPQS algorithm to fully leverage its dynamic prioritization capabilities. As the simulation progresses and more requests are received, the DMPQS algorithm gradually reaches a steady state where its efficiency and performance stabilize. This is due to the algorithm's ability to adapt and allocate resources dynamically, ensuring that even as the volume of requests grows, the system can handle them effectively without significant delays.Conversely, the traditional FIFO algorithm, which lacks this dynamic adaptation, begins to exhibit signs of congestion as the number of requests grows. The inherent characteristic of the FIFO algorithm is to process requests in the order they are received, without considering their relative importance or urgency. This can lead to a backlog of requests, as higher priority tasks may be delayed by a queue of lower priority ones.

As depicted in Fig. 7, the performance of the traditional FIFO algorithm declines further as the number of rounds—and consequently, the number of requests—increases. The congestion caused by this backlog of requests results in increased response times and reduced overall network efficiency. This decline is particularly evident when compared to the DMPQS algorithm, which maintains higher link utilization rates and lower response times even under the same conditions.

In summary, the simulation results highlight the long-term benefits of using the DMPQS algorithm in scenarios where request volume is expected to grow. The ability of the DMPQS



(a) DMPQS algorithm with best selection

(b) DMPQS algorithm with FIFO

(c) DMPQS algorithm with random selection

(d) Traditional FIFO

Fig. 7. Performance on link utilization rate.

algorithm to maintain high performance as the system scales up is a testament to its robustness and suitability for modern network environments where adaptability and efficiency are key.

D. Performance on State of Requests

As previously demonstrated, our evaluation of the DMPQS algorithm has encompassed its performance concerning the average response time of requests and the link utilization rate. we now consider the impact of the algorithm on the state of requests. We will assess how the DMPQS algorithm and traditional FIFO algorithms manage an increasing number of service requests, including the possibility that some requests may be delayed or dropped by the algorithms. This assessment is essential for understanding how each algorithm performs under load and its ability to handle request overflow without discarding requests unnecessarily.

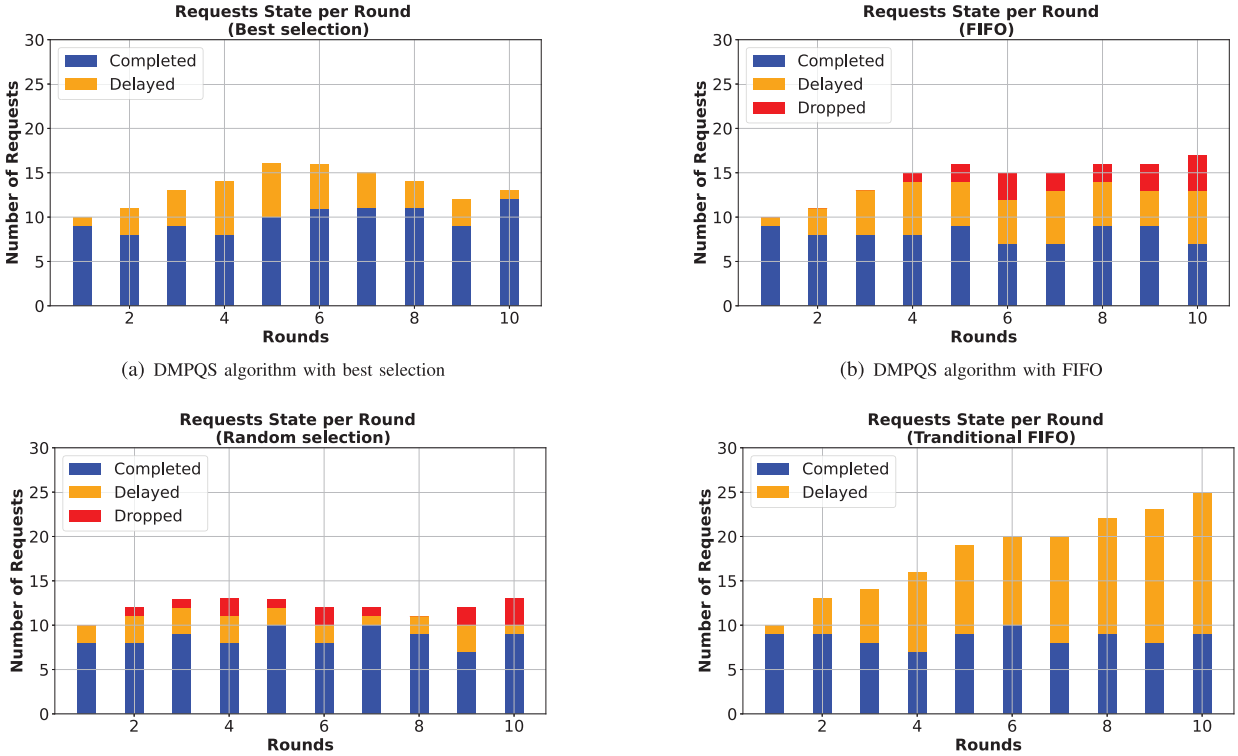
As shown in Fig. 8 the data reveals significant insights into how the different scheduling algorithms manage service requests. The DMPQS algorithm with the best selection strategy completes an average of 9.8 requests per round, with 3.6 requests experiencing delays. This is notably higher than the Traditional FIFO, which completes the same number of requests but with a higher delay rate of 9.6 requests per round. The DMPQS algorithm with FIFO selection lags slightly with 8.1 completed requests per round, yet it has a higher rate of request abandonment at 1.7 per round, compared to the best selection strategy's 0 abandoned requests. It is worth noting that although the DMPQS algorithm with the best selection strategy did not abort any requests in the experiment, there is a theoretical possibility that such occurrences could happen.

When we combine these findings with the previous results showing a 29.1% reduction in average response time and a 12.4% increase in link utilization for the best selection strategy of the DMPQS algorithm, a clear pattern emerges. The DMPQS algorithm, particularly with the best selection, not only reduces response times and increases link utilization rates but also effectively minimizes task delays and abandonment. This suggests that the dynamic prioritization inherent in the DMPQS algorithm allows for more efficient request handling, leading to better overall network performance.

In summary, the DMPQS algorithm outperforms the Traditional FIFO across multiple aspects. The best selection strategy within DMPQS algorithm demonstrates superior task completion rates with fewer delays and no abandonments, aligning with its previously observed efficiency in reducing response times and enhancing link utilization. These comprehensive results underscore the DMPQS algorithm's robustness and its potential as a preferred scheduling strategy in network management, offering a balanced approach to optimizing both the speed and quality of service. Future work should focus on implementing the DMPQS algorithm in various network conditions to further validate its performance benefits.

E. Scalability Analysis and Resource Overhead

1) Small-Scale Versus Large-Scale Networks: The scalability of HWDSQP, particularly the HWFR and DMPQS algorithms, can be evaluated through their computational complexity and the observed experimental trends. While the study focused on a network with 10 repeaters, the results provide a foundation for extrapolation to larger and more dynamic networks. The HWFR algorithm's path selection complexity



(a) DMPQS algorithm with best selection

(b) DMPQS algorithm with FIFO

(c) DMPQS algorithm with random selection

(d) Traditional FIFO

Fig. 8. Performance on state of requests.

scales with the network size, as it evaluates link costs across all available paths. With a complexity of $O(N \log N)$, where N is the number of nodes, the algorithm remains computationally efficient for medium-to-large networks. Similarly, the DMPQS algorithm, which leverages dynamic multi-priority queues for request scheduling, has a complexity of $O(M \cdot L)$, where M is the number of requests, and L is the number of candidate paths. This ensures scalable scheduling performance under increasing network sizes.

2) Memory Requirements: Each quantum repeater and the network controller require quantum memory for storing entangled pairs. For a network with 10 nodes, the memory requirements remain modest, scaling linearly with the number of active links. The network controller also requires classical memory to store topology graphs and historical cost data for path optimization. These demands are manageable with modern computational resources.

3) Computational Load: The HWFR algorithm evaluates metrics such as distance, fidelity, and resource constraints for path selection. In the tested 10-node network, the processing time per request aligns with real-time operational requirements. With a complexity of $O(N \log N)$, the algorithm remains practical for larger networks. Similarly, the DMPQS algorithm dynamically schedules requests with a complexity of $O(M \cdot L)$, scaling with the number of requests and available paths. This ensures efficient resource utilization while maintaining system responsiveness.

4) Discussion: The HWDSQP protocol demonstrates strong potential for scalable deployment in distributed quantum networks. Its modular design ensures adaptability to various network scales, making it suitable for applications such as quantum key distribution and distributed quantum computing. The protocol's ability to dynamically prioritize and optimize resource allocation enhances network reliability and efficiency. Future work will validate these findings through large-scale simulations and explore integrating quantum error correction for increased robustness.

V. Related Work

A. Distributed Quantum computing

Due to hardware constraints, the current quantum computers support a limited number of qubits, which is far from satisfying the requirements of complex computational tasks. Consequently, distributed quantum computing (DQC) has been proposed to divide complex computational tasks into relatively weakly correlated sub-tasks and distribute them across different quantum computers for execution. Caleffi et al. [44] conduct a systematic survey, categorizing the challenges faced by DQC into four aspects: quantum algorithms, quantum networks, quantum compilation, and quantum simulation. Regarding quantum networks, they discussed the impact of noise and quantum coherence on communication quality and efficiency. DiAdamo et al. [45] take the variational quantum eigensolver (VQE) problem as an example, and improved the design of quantum circuits and network control protocols to enable efficient distributed execution. To support parallel primitives in quantum computing, Häner et al. extend the classical parallel computing primitive, message

passage interface (MPI), to QMPI [46], which enables the implementation, debugging, and evaluation of quantum computing algorithms. Cuomo et al. [47] formalize the distributed compilation problem as a dynamic network flow problem to address the compatibility issues between quantum circuits and actual quantum hardware, and used an approximate algorithm to solve it. Although the aforementioned work has been done, there is a long way to go to realize large-scale, general-purpose distributed quantum computing. *Our work is dedicated to the implementation and optimization of quantum data networks, which is orthogonal to the existing work on parallel primitives and compilation.*

B. Routing

In the context of distributed quantum data networks, quantum routing considers how to efficiently and securely transmit qubits from a source quantum computer to a destination computer through the principles of quantum mechanics. The key technologies involved include repeater selection, quantum entanglement allocation, and channel noise error correction. Mehic et al. [17] consider quantum key exchange and the foundations of quantum routing from the perspective of network design. Amer et al. [18] assume a trusted quantum relay and node environment, and compare different routing protocols and their performance in various use cases. Shi and Qian [19] envision how to use quantum entanglement to improve the success rate of long-distance quantum communication when the relay nodes are untrusted. They design an entanglement routing model distinct from classical routing problems and propose the Q-CAST algorithm to solve the long-distance quantum entanglement problem. To address the failure of entanglement pair generation during the routing process in existing routing protocols, Li et al. [13] propose purification-enabled entanglement routing designs. They also extend the Dijkstra algorithm to Q-LEAP to reduce the complexity of routing lookup. Chehimi and Saad [20] study the rate of quantum entanglement generation and its distribution on quantum memories during the routing process in a heterogeneous quantum network environment. They formalize this problem as an integer nonlinear programming problem and verify the superiority of their algorithm through simulation experiments. *We propose an optimized routing scheme for the controller in quantum networks, which considers key metrics such as the transmission distance and fidelity of qubits when calculating path costs. The algorithm incorporates historical cost considerations, allowing it to optimize based on past path selections and performance data and enhance the overall efficiency and reliability of the network.*

C. Scheduling

Quantum communication relies on limited quantum channel resources, such as qubits and quantum relay nodes. How to reasonably allocate these scarce resources among different users and applications is a key issue. Addressing the new features of distributed quantum communication, such as multi-peer connection and fluctuating qubit exchange rate, Cicconetti et al. [48] design an empirically-based communication optimization scheme from the perspective of network resource allocation. Zhu et al. [49] extend the application scenario of quantum key distribution networks to cloud-edge collaboration, and formalized the network resource allocation problem as an integer linear programming problem, using heuristic algorithms to achieve a trade-off between the demands of communication, computation, caching, and cryptography. Grillo et al. [50] envision how to optimize the exchange rate in quantum key distribution networks through satellite communication. Through a centralized resource scheduling scheme, they can select the most convenient communication path and resource allocation plan. Since considering routing and resource allocation in quantum communication is a complex decision problem, Sharma et al. [51] propose a deep reinforcement learning method to select the appropriate routing path and the optimal network resource configuration.

Our scheduling algorithm employs a multi-priority queue scheduling mechanism that dynamically adjusts resource allocation based on the current network status. This approach effectively handles fluctuations in qubit exchange rates and multipoint connection demands. The algorithm also considers the possibility of quantum entanglement generation failures and incorporates an efficient retry mechanism, further enhancing the network's stability and reliability.

VI. Conclusion

In this paper, we design and implement `HWDSQP` aimed at addressing the reliability and throughput issues of quantum links in quantum communication networks. This protocol comprehensively considers the reliability, fidelity, and resource utilization of quantum links, enabling it to dynamically adapt to changing conditions within the quantum network, ensuring continuous, stable, and efficient communication. We propose the HWFR algorithm, which optimizes path selection by dynamically assigning weights to metrics such as link length, noise levels, entanglement success rates, and quantum repeater resource limitations, thus improving the high fidelity and reliability of quantum communication. In addition, we design a DMPQS algorithm that dynamically adjusts request priorities based on the urgency of service requests, resource usage, and other requirements. The experimental results show that, compared to the traditional FIFO algorithm, the DMPQS algorithm reduces the average response time by approximately 30% and improves link utilization by 12.4%.

References

[1] S. S. Gill et al., "Quantum computing: A taxonomy, systematic review and future directions," *Softw., Pract. Exper.*, vol. 52, no. 1, pp. 66–114, Jan. 2022.

[2] Z. Yang, M. Zolanvari, and R. Jain, "A survey of important issues in quantum computing and communications," *IEEE Commun. Surveys Tuts.*, vol. 25, no. 2, pp. 1059–1094, 2nd Quart., 2023.

[3] S. K. Sood and Pooja, "Quantum computing review: A decade of research," *IEEE Trans. Eng. Manag.*, vol. 71, pp. 6662–6676, 2023.

[4] M. Mehic et al., "Quantum cryptography in 5G networks: A comprehensive overview," *IEEE Commun. Surveys Tuts.*, vol. 26, no. 1, pp. 302–346, 1st Quart., 2024.

[5] D. Peral-García, J. Cruz-Benito, and F. J. García-Peñalvo, "Systematic literature review: Quantum machine learning and its applications," *Comput. Sci. Rev.*, vol. 51, Feb. 2024, Art. no. 100619.
[6] J. W. Z. Lau, K. H. Lim, H. Shrotriya, and L. C. Kwek, "NISQ computing: Where are we and where do we go?," *AAPPS Bull.*, vol. 32, no. 1, p. 27, Sep. 2022.
[7] D. Cuomo, M. Caleffi, and A. S. Cacciapuoti, "Towards a distributed quantum computing ecosystem," *IET Quantum Commun.*, vol. 1, no. 1, pp. 3–8, Jul. 2020.
[8] Z. Li et al., "Entanglement-assisted quantum networks: Mechanics, enabling technologies, challenges, and research directions," *IEEE Commun. Surveys Tuts.*, vol. 25, no. 4, pp. 2133–2189, 4th Quart., 2023.
[9] Z. Li et al., "Building a large-scale and wide-area quantum internet based on an OSI-alike model," *China Commun.*, vol. 18, no. 10, pp. 1–14, Oct. 2021.
[10] M. Erhard, M. Krenn, and A. Zeilinger, "Advances in high-dimensional quantum entanglement," *Nat. Rev. Phys.*, vol. 2, no. 7, pp. 365–381, Jun. 2020.
[11] L. Chen et al., "A heuristic remote entanglement distribution algorithm on memory-limited quantum paths," *IEEE Trans. Commun.*, vol. 70, no. 11, pp. 7491–7504, Nov. 2022.
[12] L. Chen et al., "REDP: Reliable entanglement distribution protocol design for large-scale quantum networks," *IEEE J. Sel. Areas Commun.*, vol. 42, no. 7, pp. 1723–1737, Jul. 2024.
[13] J. Li et al., "Fidelity-guaranteed entanglement routing in quantum networks," *IEEE Trans. Commun.*, vol. 70, no. 10, pp. 6748–6763, Oct. 2022.
[14] Y. Cao, Y. Zhao, J. Li, R. Lin, J. Zhang, and J. Chen, "Hybrid trusted/untrusted relay-based quantum key distribution over optical backbone networks," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 9, pp. 2701–2718, Sep. 2021.
[15] F. Hahn, A. Pappa, and J. Eisert, "Quantum network routing and local complementation," *npj Quant. Inf.*, vol. 5, no. 1, p. 76, Dec. 2019.
[16] M. Wang et al., "A segment-based multipath distribution method in partially-trusted relay quantum networks," *IEEE Commun. Mag.*, vol. 61, no. 12, pp. 184 –190, 2023.
[17] M. Mehic et al., "Quantum key distribution: A networking perspective," *ACM Comput. Surv.*, vol. 53, no. 5, pp. 1–41, 2020.
[18] O. Amer, W. O. Krawec, and B. Wang, "Efficient routing for quantum key distribution networks," in *Proc. IEEE Int. Conf. Quantum Comput. Eng. (QCE)*, Oct. 2020, pp. 137–147.
[19] S. Shi and C. Qian, "Concurrent entanglement routing for quantum networks: Model and designs," in *Proc. Annu. Conf. ACM Special Interest Group Data Commun. Appl., Technol., Archit., Protocols Comput. Commun.*, 2020, pp. 62–75.
[20] M. Chehimi and W. Saad, "Entanglement rate optimization in heterogeneous quantum communication networks," in *Proc. 17th Int. Symp. Wireless Commun. Syst. (ISWCS)*, Sep. 2021, pp. 1–6.
[21] C. Cicconetti, M. Conti, and A. Passarella, "Request scheduling in quantum networks," *IEEE Trans. Quant. Eng.*, vol. 2, pp. 2–17, 2021.
[22] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik*, vol. 1, pp. 287–290, Dec. 2022.
[23] M. Pant et al., "Routing entanglement in the quantum internet," *NPJ Quantum Inf.*, vol. 5, no. 1, p. 25, Mar. 2019.
[24] Z.-S. Yuan, Y.-A. Chen, B. Zhao, S. Chen, J. Schmiedmayer, and J.-W. Pan, "Experimental demonstration of a BDCZ quantum repeater node," *Nature*, vol. 454, no. 7208, pp. 1098–1101, Aug. 2008.
[25] G. Vardoyan, S. Guha, P. Nain, and D. Towsley, "On the stochastic analysis of a quantum entanglement switch," *ACM SIGMETRICS Perform. Eval. Rev.*, vol. 47, no. 2, pp. 27–29, Dec. 2019.
[26] L. Chen et al., "Q-DDCA: Decentralized dynamic congestion avoid routing in large-scale quantum networks," *IEEE/ACM Trans. Netw.*, vol. 32, no. 1, pp. 368 –381, 2023.
[27] P. Shor, "Algorithms for quantum computation: Discrete logarithms and factoring," in *Proc. 35th Annu. Symp. Found. Comput. Sci.*, 1994, pp. 124–134.
[28] M. Schlosshauer, "Quantum decoherence," *Phys. Rep.*, vol. 831, pp. 1–57, Jan. 2019.
[29] K. C. Miao et al., "Universal coherence protection in a solid-state spin qubit," *Science*, vol. 369, no. 6510, pp. 1493–1497, Sep. 2020.
[30] M. J. Bayerbach, S. E. D'Aurelio, P. van Loock, and S. Barz, "Bell-state measurement exceeding 50% success probability with linear optics," *Sci. Adv.*, vol. 9, no. 32, Aug. 2023, Art. no. eadf4080.
[31] J. L. Park, "The concept of transition in quantum mechanics," *Found. Phys.*, vol. 1, no. 1, pp. 23–33, 1970.
[32] R. Van Meter, T. Satoh, T. D. Ladd, W. J. Munro, and K. Nemoto, "Path selection for quantum repeater networks," *Netw. Sci.*, vol. 3, pp. 82–95, Dec. 2013.
[33] S. Bratzik, S. Abruzzo, H. Kampermann, and D. Bruß, "Quantum repeaters and quantum key distribution: The impact of entanglement distillation on the secret key rate," *Phys. Rev. A, Gen. Phys.*, vol. 87, no. 6, Jun. 2013, Art. no. 062335.
[34] H.-J. Briegel, W. Dür, J. I. Cirac, and P. Zoller, "Quantum repeaters: The role of imperfect local operations in quantum communication," *Phys. Rev. Lett.*, vol. 81, p. 5932, Dec. 1998.
[35] M. Żukowski, A. Zeilinger, M. A. Horne, and A. K. Ekert, "'Event-ready-detectors' bell experiment via entanglement swapping," *Phys. Rev. Lett.*, vol. 71, no. 26, pp. 4287–4290, Dec. 1993.
[36] Z. Wang et al., "An efficient scheduling scheme of swapping and purification operations for end-to-end entanglement distribution in quantum networks," *IEEE Trans. Netw. Sci. Eng.*, vol. 11, no. 1, pp. 380–391, Jan. 2024.
[37] Z. Li et al., "NarrowGap: Reducing bottlenecks for end-to-end entanglement distribution in quantum networks," *IEEE Trans. Netw.*, vol. 33, no. 1, pp. 162–177, Feb. 2025.
[38] A. Dahlberg et al., "A link layer protocol for quantum networks," in *Proc. Annu. Conf. ACM Special Interest Group Data Commun. Appl., Technol., Archit., Protocols Comput. Commun. (SIGCOMM)*. ACM, 2019, pp. 159–173.
[39] M. K. Bhaskar et al., "Experimental demonstration of memory-enhanced quantum communication," *Nature*, vol. 580, no. 7801, pp. 60–64, Apr. 2020.
[40] N. Tomm et al., "A bright and fast source of coherent single photons," *Nature Nanotechnol.*, vol. 16, no. 4, pp. 399–403, Apr. 2021.
[41] J. Yin et al., "Satellite-based entanglement distribution over 1200 kilometers," *Science*, vol. 356, no. 6343, pp. 1140–1144, Jun. 2017.
[42] Z. Xiao et al., "Purification scheduling control for throughput maximization in quantum networks," *Commun. Phys.*, vol. 7, no. 1, p. 307, Sep. 2024.
[43] Z. Li et al., "Swapping-based entanglement routing design for congestion mitigation in quantum networks," *IEEE Trans. Netw. Service Manage.*, vol. 20, no. 4, pp. 3999–4012, 2023.
[44] M. Caleffi, M. Amoretti, D. Ferrari, J. Illiano, A. Manzalini, and A. S. Cacciapuoti, "Distributed quantum computing: A survey," *Comput. Netw.*, vol. 254, Dec. 2024, Art. no. 110672.
[45] S. DiAdamo, M. Ghibaudi, and J. Cruise, "Distributed quantum computing and network control for accelerated VQE," *IEEE Trans. Quantum Eng.*, vol. 2, pp. 1–21, 2021.
[46] T. Häner, D. S. Steiger, T. Hoefler, and M. Troyer, "Distributed quantum computing with QMPI," in *Proc. SC21: Int. Conf. High Perform. Comput., Netw., Storage Anal.*, Nov. 2021, pp. 1–15.
[47] D. Cuomo et al., "Optimized compiler for distributed quantum computing," *ACM Trans. Quantum Comput.*, vol. 4, no. 2, pp. 1–29, Jun. 2023.
[48] C. Cicconetti, M. Conti, and A. Passarella, "Resource allocation in quantum networks for distributed quantum computing," in *Proc. IEEE Int. Conf. Smart Comput. (SMARTCOMP)*, Jun. 2022, pp. 124–132.
[49] Q. Zhu, X. Yu, Y. Zhao, A. Nag, and J. Zhang, "Resource allocation in quantum-key-distribution-secured datacenter networks with cloud-edge collaboration," *IEEE Internet Things J.*, vol. 10, no. 12, pp. 10916–10932, 2023.
[50] M. Grillo, A. A. Dowhuszko, M.-A. Khalighi, and J. Hämäläinen, "Resource allocation in a quantum key distribution network with LEO and GEO trusted-repeaters," in *Proc. 17th Int. Symp. Wireless Commun. Syst. (ISWCS)*, Sep. 2021, pp. 1–6.
[51] P. Sharma, S. Gupta, V. Bhatia, and S. Prakash, "Deep reinforcement learning-based routing and resource assignment in quantum key distribution-secured optical networks," *IET Quantum Commun.*, vol. 4, no. 3, pp. 136–145, Sep. 2023.

Liwei Lin received the Ph.D. degree in computer science from Shanghai Jiao Tong University (SJTU), China, in 2020. He is currently a Lecturer with the School of Computer Science and Mathematics, Fujian University of Technology, China. His research interests include quantum communication, data center networks, mobile computing, and cloud computing.

Rongbo Ma is currently pursuing the bachelor's degree with the Department of Information Security Engineering, Shanghai Jiao Tong University, Shanghai, China. His research interests include quantum communication, machine learning systems, and cloud computing.

Zejian Wang received the bachelor's degree in information security from Shanghai Jiao Tong University, China, where he is currently pursuing the degree in computer science. His research interests include quantum communication, machine learning systems, and acceleration on inference of machine learning algorithms.

Zinuo Cai received the bachelor's degree in software engineering from Shanghai Jiao Tong University, China, where he is currently pursuing the degree in computer science. His research interests include resource schedule and quantum computing in cloud.

Haochen Xu is currently pursuing the bachelor's degree in network information security with Shanghai Jiao Tong University, China. His research interests include distributed systems and AI security.

Baoheng Zhang received the master's degree in aircraft design from Northwestern Polytechnical University. He is currently a Senior Engineer with the Aerospace System Engineering Shanghai. He mainly engages in information processing and motion simulation.

Ruhui Ma (Member, IEEE) received the Ph.D. degree in computer science from Shanghai Jiao Tong University. He is currently an Associate Professor with the Department of Computer Science and Engineering, Shanghai Jiao Tong University. His research interests include cloud computing systems, quantum computing, and machine learning.

Rajkumar Buyya (Fellow, IEEE) is currently a Redmond Barry Distinguished Professor and the Director of the Cloud Computing and Distributed Systems (CLOUDS) Laboratory, The University of Melbourne, Australia. He has authored more than 625 publications and seven textbooks, including *Mastering Cloud Computing*, published by McGraw Hill, China Machine Press, and Morgan Kaufmann for Indian, Chinese, and international markets, respectively. He is one of the highly cited authors in computer science and software engineering worldwide.