Entanglement Routing Design Over Quantum Networks

Yiming Zeng, *Member, IEEE*, Jiarui Zhang, *Member, IEEE*, Ji Liu, *Member, IEEE*, Zhenhua Liu, *Member, IEEE*, and Yuanyuan Yang, *Fellow, IEEE*

***Abstract*— Quantum networks have emerged as a future platform for quantum information exchange and applications, with promising capabilities far beyond traditional communication networks. Remote quantum entanglement is an essential component of a quantum network. How to efficiently design a multi-routing entanglement protocol is a fundamental yet challenging problem. In this paper, we study a quantum entanglement routing problem to simultaneously maximize the number of quantum-user pairs and their expected throughput. Our approach is to formulate the problem as two sequential integer programming problems. We propose efficient entanglement routing algorithms for these two optimization problems and analyze their time complexity and performance bounds. Evaluation results highlight that our approach outperforms existing solutions in both the number of quantum-user pairs served and network throughput.**

***Index Terms*— Quantum networks, multi-entanglement routing, swapping, integer programming.**

I. Introduction

QUANTUM networks are capable of generating, transmitting, and computing quantum information (qubits) in addition to classical data (ebits) between quantum processors [1]. They support massive quantum applications in both quantum computing and quantum communication systems, such as distributed quantum computing [2], quantum communication [3], quantum machine learning [4] and quantum key distribution [5]. Several trial quantum communication systems in research labs have been constructed, such as long-distance link (40 kilometers) teleportation over the fiber link [6], mobile quantum network [7], and integrated entanglement system through satellites that can support the entanglement over 4,600 kilometers [8].

Entanglement is an essential component of almost all quantum applications mentioned above. For example, the quantum key distribution system has provable security for the distributed information [1] by taking advantage of the entanglement and no-cloning theorem [9]. Supporting long-distance entanglement is critical for quantum networks. However, the probabilistic entanglement process is unstable. Different from binary ebits in traditional communication, qubits created by photons are extremely fragile. The successful entanglement rate among qubits decreases exponentially with the transmission length. Hence, to enable long-distance entanglement of quantum users in the quantum network, quantum switches are placed in the network as relays to supply end-to-end entanglements for multiple quantum users that demand them [10], [11]. Quantum switches are equipped with quantum memories (qubits) and have the ability to perform multi-qubits measurement (swapping) [11].

The *entanglement routing* problem about *how to build long-distance entanglement through quantum switches* is crucial in a quantum network. Thoughtful design for the entanglement routing in the quantum network can boost the network performance by efficiently utilizing resources, e.g., switch memories.

While large-scale quantum networks have not been implemented outside of the lab due to physical and experimental challenges, it is still valuable to investigate the entanglement routing problem from the network layer for the future. The entanglement routing problem has been drawing great attention in previous studies. The network model with a single switch and multiple users is considered in [11] and [12]. In [13], the entanglement waiting time for a single path with one source-destination pair is discussed. These papers focus on the theoretical analyses of performance on a single switch or a single path and do not address routing in large-scale quantum networks. References [11], [13], [14], [15], [16], and [17] study the entanglement routing problem or theoretical entanglement performance on the special network topologies such as a single switch, single entanglement path, rings, grids, or spheres. References [18] and [19] consider a general quantum network for multiple quantum-user pairs entanglement. However, their strategy is a greedy algorithm to maximize the throughput of the quantum user pair one by one which may assign the majority of resources to a limited number of quantum users, while others are starving. The proposed algorithm incurs high time complexity and lacks performance guarantees. In addition, the number of users that a network can serve is another crucial criterion in evaluating its performance, as it represents the network's service capacity. This aspect has been widely discussed in the networking area (e.g., wireless networks [20], [21], optical switching networks [22]), but has

Manuscript received 26 August 2022; revised 21 February 2023 and 3 May 2023; accepted 31 May 2023; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor S. M. Kim. Date of publication 19 June 2023; date of current version 16 February 2024. This work was supported in part by the U.S. National Science Foundation under Grant 1717731, Grant 1730291, Grant 2231040, Grant 2230620, Grant 2214980, Grant 2046444, Grant 2106027, and Grant 2146909. *(Yiming Zeng and Jiarui Zhang contributed equally to this work.) (Corresponding author: Yuanyuan Yang.)*

Yiming Zeng, Jiarui Zhang, Ji Liu, and Yuanyuan Yang are with the Department of Electrical and Computer Engineering, Stony Brook University, Stony Brook, NY 11794 USA (e-mail: yiming.zeng@stonybrook.edu; jiarui.zhang.2@stonybrook.edu; ji.liu@stonybrook.edu; yuanyuan.yang@stonybrook.edu).

Zhenhua Liu is with the Department of Applied Mathematics and Statistics, Stony Brook University, Stony Brook, NY 11794 USA (e-mail: zhenhua.liu@stonybrook.edu).

Digital Object Identifier 10.1109/TNET.2023.3282560

never been discussed in quantum networks before. Maximizing the number of users allows the network to fully utilize its resources and serve as many users as possible, rather than focusing on a limited number of users. In this paper, we will consider optimizing both the network throughput and the number of served quantum user pairs to improve the overall performance of the network.

Moreover, most existing works treat the transmission link capacity as the main bottleneck of the network. However, the switch resources (i.e., the number of qubits in the switch) are the major limitation of the quantum network in reality. A most recent quantum processor can only have up to 127 qubits [23]. The average cost to build a single qubit in a quantum processor can be up to 10,000 U.S. dollars [24]. Meanwhile, it is still very difficult, if not impossible, to build a quantum processor with a large number of qubits embedded. On the other hand, optical fibers have mature technology with relatively low cost (i.e., 0.5 U.S. dollars per kilometer). One optical fiber cable can contain up to 19 cores to support information processing [25], [26], each of which can be used as an independent link for entanglement. In addition, multiple optical fiber cables can be placed between quantum switches. Hence, the transmission link has enough capacity to serve the entanglement demands of the quantum users.

In this paper, we consider a general quantum network structure and present a comprehensive entanglement process for multiple pairs of quantum users. Our goal is to *maximize the number of quantum-user pairs and the (expected) network throughput at the same time*. Our contributions are as follows:

1) We describe the detailed multi-entanglement routing process for multiple quantum-user pairs as *Offline Stage* and *Online Stage*.
2) In *Offline Stage*, we formulate the problem as two integer linear programming problems that are NP-Complete and NP-Hard, respectively, whose goals are to maximize the number of quantum-user pairs served and expected throughput.
3) We first design an algorithm to maximize the number of quantum-user pairs that can be served by the network. Then, we propose an algorithm to maximize the expected network throughput of served quantum-user pairs. The proposed efficient algorithms are for *Offline Stage* with lower time complexity and performance guarantees.
4) We further propose an algorithm to design the recovery path set and design the swapping policy to recover the failed entanglement links for *Online Stage*.
5) Results of evaluation highlight that our approach can improve the number of served quantum-user pairs by 85% and the expected throughput by 27% on average compared with existing baselines.

To the best of our knowledge, this is the first paper to maximize the number of quantum-user pairs served and expected throughput simultaneously. We also elaborate on the background of the quantum network including its necessary components and clarify its relationship with the traditional Internet.

Fig. 1. A teleportation example. The source node teleports a qubit by a pair of entanglement qubits.

The organization of the remaining of this paper is as follows. We first introduce the background of the quantum network and the multi-entanglement routing process in Section II. Then, we present the quantum network model and formulate the routing entanglement process as two integer linear programming problems in Section III based on the routing process introduced in Section II. The entanglement routing algorithms for *Offline Stage* are proposed in Section IV and Section V for two integer linear programming problems, respectively. In Section VI, we propose the algorithm for constructing the recovery path set and design the swapping policy for *Online Stage*. We conduct extensive simulations to discuss and analyze the performance of our proposed algorithms and compare them with previous work in Section VII, followed by related work in Section VIII and the conclusion in Section IX.

II. Quantum Network Background

In this section, we introduce some basic quantum network backgrounds, including quantum network components and multi-routing entanglement processes.

A. Basic Quantum Terminologies

1) Qubit: In the quantum network or quantum computing, a qubit is a basic unit to represent quantum information. A qubit can be an electron or a photon or a nucleus from an atom. A qubit is described by its state [1]. Different from an ebit in the classical Internet representing 0 or 1, a qubit can present a coherent superposition of both. For example, a basic qubit state $|\phi\rangle = x|0\rangle + y|1\rangle$, where $|\rangle$ is a ket which denotes a vector (i.e.,$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$), x and y are complex numbers that satisfy $|x|^2 + |y|^2 = 1$.

2) Entanglement: Entanglement is a phenomenon in which a group of qubits expresses a high correlation state which can not be explained by individual qubits states. In this paper, we consider the simplest case of two qubits entanglement which is a bipartite entangled state. In quantum physic, a simple way to entangle two independent qubits is by using CNOT gate [27]. When the entanglement qubits number is two, Bell-state measures (BSMs) can be applied to measure the entanglement.

3) Teleportation: If a pair of entanglement qubits are shared by two nodes, the secret information can be transmitted from one node to another one with the help of quantum measurement. This process is called teleportation. An example is illustrated in Figure 1.

4) Entanglement Swapping: Swapping is a quantum operation in which if two processors each have a different qubit entangled with another common processor, then the qubits of these two processors are entangled directly with the help of the

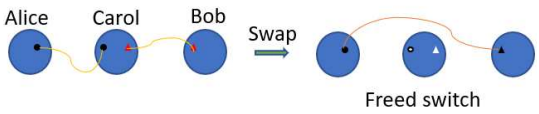


Fig. 2. A swapping example.

common processor. Figure 2 presents an example of swapping. If Alice shares an entangled qubit pair (Bell pair) with the middle node Carol, and Carol shares another entangled qubit pair with Bob, Carol can teleport its qubit entangled with Alice to Bob, then Alice and Bob are entangled directly [28].

B. Quantum Network Components

With these basic concepts, we first introduce several important components of the quantum network.

1) Quantum Users: A quantum user is a quantum processor or a quantum virtual machine consisting of several quantum processors that has demands to entangle with the other user in the quantum network for quantum applications. Qubits in a processor have a range of potential applications, including quantum computing and communication. In this paper, we mainly consider the communication function of quantum users. The user who intends to entangle with the other user is called a source node. Another user who tried to be entangled is called a destination node.

2) Quantum Switches: The quantum switch is a quantum processor with quantum memories to work as relays for the entanglement process in the quantum network [11], [29]. The qubits in the processor are mainly for communication. They can either transmit qubits or establish the entanglement at distant nodes without physically sending an entangled qubit by swapping.

3) Quantum Links: Quantum links are the links used for connecting quantum switches and quantum users. In this paper, we assume that the quantum network is connected by optical fiber cables among quantum switches and quantum users. The successful entanglement generation probability is related to the material and the length of the quantum link, i.e., $p = e^{-\alpha L}$, where α is a positive constant related to the material of the quantum link and L is the length of the quantum link.

4) The Traditional Internet (The Cloud): The quantum network cooperates with the traditional Internet together for quantum users' entanglement routing. The Internet is responsible for exchanging information among networks. Quantum users and quantum switches are equipped with traditional computing devices (e.g., computers) and can communicate with others through the traditional Internet.

We list several of the most important roles of the traditional Internet (the cloud) in the quantum network but do not include all.

- The cloud is the center of the network that knows detailed information about the quantum network including quantum-user pairing information, the quantum network topology, the quantum switch capacity, and so on.
- The cloud computes the offline routing paths of quantum-user pairs with network information available.
- The cloud shares network information through the Internet such as quantum-user pairing information and routing paths to quantum switches.
- During the entanglement process, adjacent switches (e.g., the graph distance between switches is small) communicate through the Internet to inform each other about link and switch states.

C. Entanglement Process

Reference [18] presents a detailed quantum network entanglement process for one quantum-user pair. Here, we summarize the routing entanglement process for multiple quantum-user pairs as a two-stage process including an offline stage and an online stage.

1) Offline Stage: In *Offline Stage*, the main tasks of the quantum network are offline entanglement routing design for quantum-user pairs and transmitting the routing paths to switches for the entanglement in *Online Stage*.

The offline routing protocol design is conducted by the cloud. We assume that the following offline information of the network is known by the cloud: the quantum-user pairing information; the network topology (switches placement and connection); switches information (the number of qubits in each switch). With all information available, the cloud computes the routing paths for quantum-user pairs with the limitation of switch capacity. After that, the routing paths computed by the cloud are transmitted through the Internet to switches for the entanglement.

2) Online Stage: In *Online Stage*, switches try to generate entanglement among links over routing paths sent from the cloud and then swap in their interiors.

The entanglement and swapping process is probabilistic, e.g., the successful entanglement rate over optical fiber is typically 0.01% [30]. The duration of the entanglement over a link is short, e.g., 1.46s [30]. The entanglement generation time of one attempt is usually 165 μs [30]. All the entanglement and swapping processes over a path should be processed in the duration of the entanglement T. The short duration of T requires the entanglement and swapping process to be carefully considered.

The detailed entanglement process is as follows.

- First, all the switches are time-synchronized through the Internet [14], which can ensure that the whole quantum network starts entanglement at the same time.
- Second, given the routing paths of all quantum-user pairs, all the switches try to process entanglement over links and swap in the interiors. Each switch can try multiple times until the entanglement is generated or the time out (greater than T).
- Third, some switches may fail to generate entanglement over part of links to build paths for quantum-user pairs. Then, the switches will try to build recovery paths for quantum-user pairs locally. Link states (successful entanglement or not) and swapping states cannot be efficiently sent to the cloud for rescheduling in T due to the Internet delay. The switch can access link states near it through communication with nearby switches with the Internet.

TABLE I
TABLE OF NOTATIONS USED IN THE NETWORK MODEL

Notation	Definition
$\mathcal{M}$	The set of $\langle S, D\rangle$ pairs
$\mathcal{A}$	The path set of $\langle S, D\rangle$ pairs
$\mathcal{A}'$	The path set of $\langle S, D\rangle$ pairs with M^2 shortest distance paths
$\hat{\mathcal{M}}$	The set of $\langle S, D\rangle$ pairs selected in STEP I
$\mathcal{S}$	The set of source $\{s_1, s_2, \ldots, s_M\}$
$\overline{\mathcal{V}}$	The set of quantum switches and users
$\mathcal{D}$	The set of destinations $\{d_1, d_2, \ldots, d_M\}$
$\mathcal{V}$	The set of switch nodes $\{v_1, v_2, \ldots, v_N\}$
$\mathcal{E}$	The set of connection links between switches $\{e_{ij}, v_i, v_j \in \overline{\mathcal{V}}\}$
L_{ij}	Length of link e_{ij}
Q_i	The number of qubits contained by switch v_i
$\hat{Q}_i$	The number of available qubits of switch v_i after STEP I
p_{ij}	The successful entanglement rate of edge e_{ij}
α	Link transmission efficiency
$\mathcal{A}_m$	The set of all paths for $\langle s_m, d_m\rangle$
A_m	A path belongs to $\mathcal{A}_m$
$\mathcal{A}'_m$	The set of M shortest distance paths for $\langle s_m, d_m\rangle$
A'_m	A path belongs to $\mathcal{A}'_m$
$\mathcal{P}_{A_m}$	The expected throughput qubits of path A_m
Q^{A_m}	The number of qubits assigned to path A_m
$x_{A_m} \in \{0, 1\}$	Binary variable indicates whether path A_m is selected in the network
h_m	Binary variable indicates whether pair m is checked in recursion
H	The set of h_m

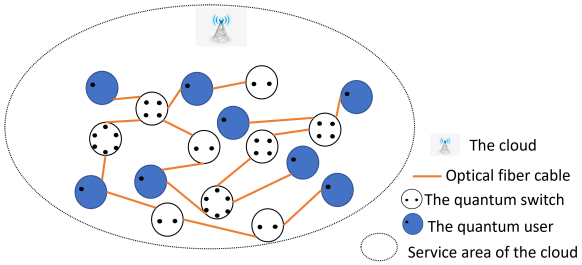


Fig. 3. An example of network.

The transmission delay from a switch to other switches in a few hops is acceptable compared with T. The exact number of hops depends on the Internet latency condition. A typical communication time between two switches within one hop is around 1 ms [30]. With the link states and swapping states available, the switches decide the recovery paths for quantum-user pairs locally.

III. QUANTUM NETWORK MODEL

In this section, we first describe the quantum network model, and then formulate the routing entanglement problem with the goal to maximize the number of quantum-user pairs that can be served by the network and their expected throughput. The network model described here follows real quantum network entanglement experiments [31], [32], [33] and previous studies about quantum entanglement routing [10], [11], [14], [18]. Figure 3 shows an example of the proposed quantum network. The key notations are summarized in Table I.

A. Network Model

Quantum Users: The Quantum user set $\mathcal{M}$ consists of M quantum-user pairs $\langle s_1, d_1\rangle, \langle s_2, d_2\rangle, \cdots, \langle s_M, d_M\rangle$. $\mathcal{S} = \{s_1, s_2, \cdots, s_M\}$ denotes the set of sources, and $\mathcal{D} = \{d_1, d_2, \cdots, d_M\}$ denotes the set of destinations. In this paper, we assume that quantum users are private entities connected to the quantum network and request entanglement. As a result, quantum users do not act as switches in the entanglement processes of other quantum user pairs. We also assume that all switches are honest and controlled by the cloud to serve the network.

Network Graph: The transmission graph consists of quantum switches and quantum links. The network is abstracted as an undirected graph which is denoted as $G = (\overline{\mathcal{V}}, \mathcal{E})$, where $\overline{\mathcal{V}} = \{\mathcal{S} \cup \mathcal{D} \cup \mathcal{V}\}$ denotes the set of quantum switches and quantum users, and $\mathcal{E} = \{e_{ij}\} \subset \{(v_i, v_j) : v_i, v_j \in \overline{\mathcal{V}}\}$ denotes the set of the quantum links.

Quantum Switch: Each quantum switch or quantum user $v_i \in \overline{\mathcal{V}}$ has Q_i qubits that can be assigned for the entanglement. We focus on 2-qubit entanglement, and the switch uses Bell-state measurements (BSMs). Since exact 2 qubits will be involved in the swapping process, we assume that Q_i is a positive even number. The successful swapping rate in each switch for any pair of qubits is uniform and denoted as $q \in [0, 1]$.

Quantum Link: e_{ij} is an edge which is an optical fiber cable connecting v_i and v_j for transmitting qubits. In each cable, there are several cores. Each core can be used as a quantum link for the entanglement of a pair of qubits. Therefore, multiple qubits can be assigned at one edge for the entanglement at the same time. We assume that the optical fiber cable contains enough cores for the entanglement between switches. The length of e_{ij} is denoted as L_{ij}. The success rate of each attempt to generate an entanglement over e_{ij} is $p_{ij} = e^{-\alpha L_{ij}}$, where α is a positive value that is determined by the properties of the physical material involved. Since p_{ij} only depends on the link length and link material, successful entanglement rates for different pairs of qubits over different cores at the same edge are the same. If a pair of qubits from a quantum-user pair successfully generate the entanglement, there will be a quantum channel between qubits. Each channel can transmit an ebit at each time.

B. Routing Metrics

We use the expected throughput of a path as a routing metric to evaluate the performance of the quantum network.

For a quantum-user pair $\langle s, d\rangle$, let $\mathcal{A}$ denote the set of all paths between s and d. Fix a path $A \in \mathcal{A}$, where $A = \{v_0, v_1, v_2, \cdots, v_{l-1}, v_l\}$, where $v_0 = s$, $v_l = d$, and l denotes the distance of A, i.e., the number of its edges. The nodes in A are listed as the order in the path from the source s to the destination d, and the adjacent nodes are connected by one quantum link. Every switch in path A assigns Q^A qubits for the entanglement, which implies the number of parallel quantum channels in path A can be up to $\frac{Q^A}{2}$.

From Section II-C, building a successful quantum channel for a quantum-user pair along a path requires all links to

generate entanglement and switches to swap successfully during the fixed time period. The probability of one attempt to generate the entanglement successfully of all links in a quantum channel at the same time is the product of the successful entanglement rate of every single link in the channel, i.e., $\Pi_{i=0}^{l} p_{i(i+1)}$. The probability of one attempt to swap successfully in all switches of a channel at the same time is the product of every switch's successful swapping rate in the channel, i.e., q^{l-1}. Then, the successful probability to build a quantum channel for the entanglement is $\Pi_{i=0}^{l} p_{i(i+1)} q^{l-1}$. Formally, the routing metrics are defined as the expected throughput of path A with $\frac{Q^A}{2}$ quantum channels for the quantum-user pair $\langle s, d \rangle$:

$$P = \frac{Q^A}{2} \cdot \Pi_{i=0}^{l} p_{i(i+1)} \cdot q^{l-1} = \frac{Q^A}{2} e^{-\alpha \sum_{i=0}^{l} L_{i,i+1}} q^{l-1}, \tag{1}$$

which indicates the expected number of ebits that can be transmitted from the source to the destination in a fixed time period. In the current setting, the routing metrics also correspond to the expected number of entangled pairs of qubits along the path that are successfully established during the fixed time period.

C. Problem Formulation

We divide our objectives into two steps, named STEP I and STEP II. In STEP I, our goal is to maximize the number of quantum-user pairs that can be served by the network, and a main routing path is selected for every chosen quantum-user pair. In STEP II, we aim to maximize the expected throughput of all selected quantum-user pairs from STEP I.

Motivation of the Two-step Design: In STEP I, we try to maximize the number of entanglement source-destination pairs and select the major path of each source-destination pair. The number of users that can be served by the network is another important criterion to describe the network performance, which has never been discussed in previous studies about the quantum network before. The width of each selected path in STEP I is one and each quantum-user pair has at most one path. However, there could be multiple paths with more than 1-width between one quantum-user pair. For example, as shown in Figure 4, after STEP I, the network can serve two source-destination pairs. The green square dash-dot lines indicate one group of possible routing paths for $\langle s_1, d_1 \rangle$ and $\langle s_2, d_2 \rangle$. However, there are still enough qubits in switches to generate another entanglement routing path for $\langle s_2, d_2 \rangle$ (the red long dash-dot line). The expected throughput of the network can be improved by adding a new path for $\langle s_2, d_2 \rangle$ pairs chosen in STEP I. Therefore, to fully utilize the network resources, we will maximize the expected throughput of the network in STEP II. Meanwhile, the major paths selected in STEP I will be kept in STEP II which ensures the selected quantum-user pairs have at least one path.

STEP I: We first formulate the problem of STEP I. To maximize the number of quantum-user pairs, we assume that $Q^A = 2$ for any path $A \in \mathcal{A}$, and at most one path can be selected for a quantum-user pair. Let the binary variable $x_{A_m} \in \{0, 1\}$ denote whether the path A_m of $\langle s_m, d_m \rangle$ is chosen to be entangled in the network or not. The formulation to maximize the number of quantum-user pairs is as follows:

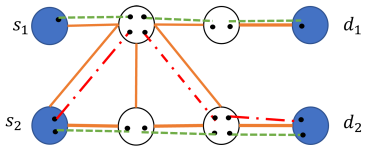


Fig. 4. An routing example. The orange lines are the optical fiber links, the green square dash-dot lines are the routing paths determined in STEP I, and the red long dash-dot lines indicate a routing path undiscovered in STEP I.

$$\textbf{Problem } \mathbf{S}_1: \max \sum_{m \in \mathcal{M}} x_{A_m}, \tag{2}$$

$$\text{subject to}: \quad x_{A_m} \in \{0, 1\}, \ \forall m \in \mathcal{M} \tag{3}$$

$$\sum_{A_m \in \mathcal{A}_m} x_{A_m} \leq 1, \ \forall m \in \mathcal{M} \tag{4}$$

$$\sum_{m \in \mathcal{M}, x_{A_m}=1} |v_i \cap (A_m x_{A_m})| \leq \frac{Q_i}{2}, \ \forall v_i \in \overline{\mathcal{V}}, \tag{5}$$

$$A_m \in \mathcal{A}_m, \forall m \in \mathcal{M}, \tag{6}$$

where $\mathcal{A}_m$ denotes the set of all paths between $\langle s_m, d_m \rangle$, and A_m is a path in $\mathcal{A}_m$. Constraint (4) denotes that at most one path can be selected for each quantum-user pair which can ensure the network serves quantum-user pairs as many as possible. Constraint (5) indicates that for any switch $v_i \in \overline{\mathcal{V}}$, the total number of qubits assigned for all paths through v_i cannot over its capacity Q_i, $|\cdot|$ in (5) denotes the number of elements in the set.

STEP II: Next, we formulate the problem in STEP II to maximize the expected throughput of selected quantum-user pairs from STEP I by determining the qubits assigned to possible paths from the path set. We first reserve the qubits in the network assigned for the main paths selected in STEP I and then maximize the expected throughput for quantum-user pairs from STEP I in the residual graph. Let $\hat{\mathcal{M}}$ denote the set of quantum-user pairs selected from STEP I, and $\hat{M}$ denote the number of pairs in $\hat{\mathcal{M}}$. $\hat{Q}_i$ denotes the available qubits of switch v_i after STEP I. The formulation is as follows:

$$\textbf{Problem } \mathbf{S}_2: \max_{Q^{A_{\hat{m}}}} \sum_{\hat{m}=1}^{\hat{M}} \sum_{A_{\hat{m}} \in \mathcal{A}_{\hat{m}}} P_{A_{\hat{m}}}, \tag{7}$$

$$\text{subject to} \quad A_{\hat{m}} \in \mathcal{A}_{\hat{m}}, \forall \hat{m} \in \hat{\mathcal{M}}, \tag{8}$$

$$Q^{A_{\hat{m}}} \in \mathbb{N}, \ \forall A_{\hat{m}} \in \mathcal{A}_{\hat{m}}, \hat{m} \in \hat{\mathcal{M}}, \tag{9}$$

$$0 \leq Q^{A_{\hat{m}}} \leq \frac{\hat{Q}_i}{2}, \quad \forall A_{\hat{m}} \in \mathcal{A}_m, \hat{m} \in \hat{\mathcal{M}}, \ \forall v_i \in \overline{\mathcal{V}}, \tag{10}$$

$$\sum_{m \in \mathcal{M}} Q^{A_{\hat{m}}} |v_i \cap A_{\hat{m}}| \leq \frac{\hat{Q}_i}{2}, \ \forall v_i \in \overline{\mathcal{V}}, \tag{11}$$

where $P_{A_{\hat{m}}}$ is the expected throughput of path $A_{\hat{m}}$ defined in (1) and $\mathbb{N}$ denotes the set of non-negative integers. $Q^{A_{\hat{m}}}$ is

the number of qubits assigned for path $A_{\hat{m}}$. (10) means that switch v_i cannot assign the qubits to the path over its capacity. (11) indicates that the number of qubits in the switch is the main limitation for the path selection of quantum-user pairs.

IV. Entanglement Routing Algorithm of STEP I

We first propose algorithms to solve Problem $\mathbf{S}_1$ in STEP I and analyze their performance and time complexity. There are two parts to solve Problem $\mathbf{S}_1$. First, we relax the binary variable x_{A_m} from $\{0,1\}$ to $[0,1]$. Let Problem $\hat{\mathbf{S}}_1$ denote the relaxed problem which is a standard linear programming problem. However, the time complexity to solve Problem $\hat{\mathbf{S}}_1$ is extremely high because of the huge size of the path set (the detailed analyses are presented in Section IV-A). Hence, we construct a smaller path set that contains efficient paths to reduce the complexity to solve Problem $\hat{\mathbf{S}}_1$. Second, we derive the feasible integer solution from the solution of Problem $\hat{\mathbf{S}}_1$ which may be fractional and not feasible.

A. Challenges

Problem $\mathbf{S}_1$ in STEP I is a binary multi-commodity flow problem. It has been proved that the problem is NP-Complete [34]. When relaxing the binary variable to be continuous, the fractional solution can be solved by the standard Linear-Programming techniques such as simplex [35].

However, the overhead for computing the path set is not considered in the previous papers [19], [36]. An inevitable prerequisite to solving Problem $\mathbf{S}_1$ is that the routing paths set $\mathcal{A}_m, \forall m \in \mathcal{M}$ should be calculated. This will add extra extremely huge computing complexity to solve Problem $\mathbf{S}_1$. More specifically, there could be up to $|\mathcal{E}|!$ paths between one quantum-user pair in a complete graph (the switches can be selected multiple times), where $|\mathcal{E}|$ is the number of edges in $\mathcal{G}$. Such a huge path set will cause great computational overhead to solve the problem. The computing complexity will be unacceptable.

B. Problem $\hat{\mathbf{S}}_1$ Solution

As we have discussed above, using a standard linear programming technique to solve Problem $\hat{\mathbf{S}}_1$ with huge path sets will bring unacceptable complexity. To address this challenge, we select the shortest distance paths of quantum-user pairs as the path set instead of all possible paths. The distance of a path indicates the number of edges on this path. Building a 1-width path requires one qubit per hop from both endpoints. Choosing shortest distance paths can consume fewer resources (e.g., the qubits in switches) to satisfy more commodities. Since we consider the number of qubits as the main limitation in the model, prioritizing a shorter-distance path consumes fewer resources from the feasible path set, which allows the network to serve more quantum-user pairs. More accurate proofs are shown in [37] and [38].

The detailed path set selection algorithm is concluded in Algorithm 1, and we explain how it runs as follows. The goal is to construct a new smaller feasible set $\mathcal{A}'$ with total $O(M^2)$ paths for Problem $\hat{\mathbf{S}}_1$, and each quantum-user pair has at least M paths. We first compute M^2 shortest distance paths by Yen's algorithm [39] for each quantum-user pair. The reasons are as follows. First, M^2 is a large enough number to ensure a source-destination pair has a set with enough paths. Meanwhile, M^2 paths do not bring a huge impact on the time complexity to solve the problem that will be discussed in detail later. Second, if we find the shortest distance paths of all quantum-user pairs directly instead of for individual pairs one by one, some pairs with a small number of shortest distance paths will be less likely considered. From the fairness aspect, we choose the path set of each quantum-user pair one by one.

Algorithm 1 Path Selection Algorithm

Input: $\mathcal{G} = (\overline{\mathcal{V}}, \mathcal{E}), \mathcal{S}, \mathcal{D}, \mathcal{M}$
Output: $\mathcal{A}'$
1: $\mathcal{A}' = \emptyset$
2: **for all** $m \in \mathcal{M}$ **do**
3: Obtain M^2 shortest distance paths of the pair $\langle s_m, d_m \rangle$ by Yen's algorithm, $\mathcal{A'}_m = \{A'^k_m\}, k \in [1, M^2]$
4: $\mathcal{A}' = \mathcal{A}' \cup \mathcal{A'}_m$
5: **end for**
6: Sort paths in $\mathcal{A}'$ by ascending order of length
7: Remove the path with largest length in $\mathcal{A}'$ until $|\mathcal{A}'| = M^2$
8: **for all** $m \in \mathcal{M}$ **do**
9: Remove the path with largest length in $\mathcal{A'}_m$ until $|\mathcal{A'}_m| = M$
10: $\mathcal{A}' = \mathcal{A}' \cup \mathcal{A'}_m$
11: **end for**

Then, we sort those M^3 paths by ascending order of their distance and add M^2 paths with the shortest distance to $\mathcal{A}'$. To ensure that each quantum-user pair has M paths, we reserve M shortest paths for each quantum-user pair and add these M paths to $\mathcal{A}'$. Finally, $\mathcal{A}'$ includes totally $O(M^2)$ paths, and there are at least M paths for each quantum-user pair.

With a smaller path set $\mathcal{A}'$, Problem $\hat{\mathbf{S}}_1$ can be solved by the standard linear programming techniques [35], [40] with the acceptable time complexity. Let $\tilde{x} = \{\tilde{x}_{A'_m} \in [0,1] | \ \forall A'_m \in \mathcal{A'}_m, m \in \mathcal{M}\}$ denote the set of Problem $\hat{\mathbf{S}}_1$ solution.

C. Integer Solution Recovery

The solution of Problem $\hat{\mathbf{S}}_1$ may be fractional which is not feasible to Problem $\mathbf{S}_1$. Hence, we need to recover the feasible integer solution of Problem $\mathbf{S}_1$ from $\tilde{x}$ and select a main routing path for each selected quantum-user pair. In this subsection, we present an integer recovery solution, based on a specialized branch-and-bound method employing an efficient branching rule [38], to enhance the overall effectiveness of the approach.

The detailed algorithm is described in Algorithm 2. Let $x^{\dagger}_{A'_m}$ denote the recovered integer solution from Algorithm 2. We first add the paths for $\tilde{x}_{A'_m} = 1$. Then, we implement an efficient branching rule of the branch-and-bound strategy, as described in [38], to derive a feasible integer solution from the fractional solution.

Branch-and-Bound Algorithm: The basic idea of the branch-and-bound strategy is to compare the results from different search branches. To accelerate the search process, we optimize the search order and cut some poorly performing branches. We only choose two branches that maximize

Algorithm 2 STEP I Integer Solution Recovery Algorithm 1

Input: LP solution to STEP I, $\tilde{x}_{A'_m}, \forall A'_m \in \mathcal{A}'_m, m \in \mathcal{M}$
Output: Integer solution to STEP I, $x^{\dagger}_{A'_m}, \forall A'_m \in \mathcal{A}'_m, m \in \mathcal{M}$

$x^{\dagger}_{A'_m} = 0, \bar{x}_{A'_m} = 0, \forall A'_m \in \mathcal{A}'_m, m \in \mathcal{M}$
Sort $\tilde{x}_{A'_m}$ in descending order
for all $A'_m, m \in \mathcal{M}$ **do**
  if $\tilde{x}_{A'_m} = 1$ **then**
    $\bar{x}_{A'_m} = 1$, mark the pair $\langle s_m, d_m \rangle$
  end if
end for
$h_m = 0, \forall m \in \mathcal{M}$
Find the maximum $\tilde{x}_{A'_m} < 1, \forall A'_m \in \mathcal{A}'_m, m \in \mathcal{M}$ that satisfies the corresponding $\langle s_m, d_m \rangle$ is not entangled
Branch-and-bound ($\emptyset$, m, $\tilde{x}_{A'_m}, \bar{x}_{A'_m}$, $H = \{h_m, m \in \mathcal{M}\}$)

Algorithm 3 Branch-and-Bound Algorithm

Input: Current path $A'_m = \{s_m, v_1, v_2, \ldots, v_l\}$, $\tilde{x}_{A'_m}, \bar{x}_{A'_m}$, m, current visited pair marks H
Output: $x^{\dagger}_{A'_m}, \forall A'_m \in \mathcal{A}'_m, m \in \mathcal{M}$

$\mathcal{A}'' = \emptyset$
for all $A'_m \in \mathcal{A}'_m$ **do**
  if $A_m \cap A'_m = A'_m$ and A'_m is feasible **then**
    $\mathcal{A}'' = \mathcal{A}'' \cup A'_m$
  end if
end for
if $|\mathcal{A}''| \leq 1$ **then**
  Mark the pair m, $h_m = 1$
  if $|\mathcal{A}''| = 1$ **then**
    $\bar{x}_{A''_m} = 1$, $A''_m \in \mathcal{A}''$
  end if
  Compare $\bar{x}_{A'_m}$ and $x^{\dagger}_{A'_m}$, update $x^{\dagger}_{A'_m}$ if necessary
  Find the maximum $\tilde{x}_{A'_{m'}} < 1, \forall A'_{m'} \in \mathcal{A}'_{m'}, m' \in \mathcal{M}$ such that $A'_{m'}$ is feasible and $\langle s_{m'}, d_{m'} \rangle$ is not marked
  Branch-and-bound ($\emptyset$, m', $\tilde{x}_{A'_m}$, $\bar{x}_{A'_m}$,H)
  Unmark the pair m, $h_m = 0$
  $\bar{x}_{A''_m} = 0$
else
  Find the minimum i s.t. exist two paths $A'_m(v_i)(j), A'_m(v'_i)(j) \in \mathcal{A}'$, satisfies $v_i \neq v'_i$, $v_i \in A'_m(v_i)(j)$, $v'_i \in A'_m(v'_i)(j)$
  Choose any $A'_m \in \mathcal{A}'_m$, append $v_{l+1}, \ldots, v_{i-1}$ to A'_m
  for all $A'_m(v_i)(j) \in \mathcal{A}''$ **do**
    $c_{v_i} = c_{v_i} + \tilde{x}'_{A'_m(v_i)(j)}$
  end for
  Find two maximum $c_{v_i}, c_{v'_i}$
  Branch-and-bound($A'_m \cup v_i$, m, $\tilde{x}_{A'_m}$, $\bar{x}_{A'_m}$,H)
  Branch-and-bound($A'_m \cup v'_i$, m, $\tilde{x}_{A'_m}$, $\bar{x}_{A'_m}$,H)
  Choose the better solution
end if

the number of entanglement pairs as the fractional solution instead of all branches. The detailed process is concluded in Algorithm 3.

To start the search process, we first sort the $\tilde{x}_{A'_m}$ in the descending order and select the quantum-user pair $\langle s_m, d_m \rangle$ with the highest $\tilde{x}_{A'_m}$ as the initial pair in the algorithm.

Then, we search the feasible integer path for this pair $\langle s_m, d_m \rangle$. Start from s_m, we search along the path until we find a branch switch v_{b_m}, e.g. $A_1 = \{s_m, v_1, v_2\}$, $A_2 = \{s_m, v_1, v_3\}$, $A_3 = \{s_m, v_1, v_4\}$, the branch switch is v_1. With multiple paths to select, the preference is to search for the paths with the larger x value to reduce branches. For example, the current path to the branch switch v_{b_m} is denoted as $A'_m = \{s_m, v_1, v_2, \cdots, v_{b_m}\}$. For every possible next switch v_i that could be added to the path, we append v_i to A'_m and count the total $x_{A'_m}$ value ensuring that A'_m remains feasible. In Algorithm 3, among all possible switches that could be added to A'_m, we select two switches denoted as v_i and v'_i with the top two total x values. Next, we continue to build paths in the following two branches denoted as $A'_m(v_i)$ and $A'_m(v'_i)$, respectively.

We repeat this process to construct the path from s_m to d_m until there is only one feasible path or no feasible paths exist. If there is only one feasible path A'_m, then $\bar{x}_{A'_m} = 1$, otherwise $\bar{x}_{A'_m} = 0$. Then, we mark the current entanglement pair $\langle s_m, d_m \rangle$ by letting $h_m = 1$ as checked in the recursion.

After traversing branches for $\langle s_m, d_m \rangle$, we choose the next quantum-user pair with the largest x value which has not been searched. The search process will end if there are no quantum-user pairs to be searched.

D. Performance Analyses and Discussion

1) Time Complexity to Solve Problem $\hat{\boldsymbol{S}}_1$ (Algorithm 1): The size of the newly selected path set $\mathcal{A}'$ is $O(M^2)$, thus there are $O(M^2)$ variables in the linear programming. The total time complexity of our Algorithm 1 and solving the corresponding linear programming is $O(M^3|\overline{\mathcal{V}}|(|\overline{\mathcal{V}}|^2+|\overline{\mathcal{V}}|\log|\overline{\mathcal{V}}|)+O((M+M^2)^{2.373}) = O(M^3|\overline{\mathcal{V}}|^3 + M^{4.746})$ (when Problem $\hat{S}_1$ is solved by [40], the linear programming solution with lowest time complexity as far as we know), where $|\overline{\mathcal{V}}|$ is the number of switches and users in $\overline{\mathcal{V}}$.

2) Algorithm 2 and Algorithm 3:
Algorithm 3 is a sub-function in Algorithm 2, so we analyze them together. Algorithm 3 is a specialized branch-and-bound algorithm with the recursion whose complexity is almost impossible to track [38]. Hence, we only analyze the performance guarantee here. Let $f^*(x^*_{A'_m}) = \sum_{m\in\mathcal{M}} x^*_{A'_m}$ and $x^*_{A'_m}$ denote the optimal result and the optimal solution of Problem $\hat{\mathbf{S}}_1$ with path set $\mathcal{A}'$, respectively. Let $f^{\dagger}(x^{\dagger}_{A'_m}) = \sum_{m\in\mathcal{M}} x^{\dagger}_{A'_m}$ and $x^{\dagger}_{A'_m}$ denote the integer result and the integer solution from the Algorithm 2, respectively. The relationship between $f^*(x^*_{A'_m})$ and $f^{\dagger}(x^{\dagger}_{A'_m})$ is stated in the following theorem.

Theorem 1: Algorithm 2 is an approximation algorithm to Problem $\boldsymbol{S}_1$ with path set $\mathcal{A}'$, and it achieves an approximation ratio of 2, i.e., $f^*(x^*_{A'_m}) \leq 2\ f^{\dagger}(x^{\dagger}_{A'_m})$.

Proof Sketch: Let ϵ denote the summation of remaining $x_{A'_m}$ after Algorithm 2. If we assume $\epsilon \leq \sum_{m\in\mathcal{M}} x^{\dagger}_{A'_m}$, then from the definition of $f^*(x^*_{A'_m})$, we obtain that

$$f^*(x^*_{A'_m}) = \sum_{m\in\mathcal{M}} x^*_{A'_m} \leq \sum_{m\in\mathcal{M}} \tilde{x}_{A'_m} = \sum_{m\in\mathcal{M}} x^{\dagger}_{A'_m} + \epsilon \leq \sum_{m\in\mathcal{M}} x^{\dagger}_{A'_m} + \sum_{m\in\mathcal{M}} x^{\dagger}_{A'_m} = 2f^{\dagger}(x^{\dagger}_{A'_m}),$$

where we use the fact that the optimal integer result is less than the relaxed continuous result in the first inequality, and we use the fact that the relaxed continuous result equals the integer result from Algorithm 2 plus the remaining part ϵ in the second equality. Note that the assumption $\epsilon \leq \sum_{m\in\mathcal{M}} x^{\dagger}_{A'_m}$ can be

Algorithm 4 STEP II Integer Solution Algorithm 1

Input: LP solution of STEP II, $\tilde{Q}^{A'_{\hat{m}}}, \forall A'_{\hat{m}} \in \mathcal{A}'_{\hat{m}}, \hat{m} \in \hat{\mathcal{M}}$
Output: Integer solution to step II, $Q^{\dagger}_{A'_{\hat{m}}}, \forall A'_{\hat{m}} \in \mathcal{A}'_{\hat{m}}, \hat{m} \in \hat{\mathcal{M}}$

1: $Q^{\dagger}_{A'_{\hat{m}}} = 0, \forall A'_{\hat{m}} \in \mathcal{A}^{I}_{m}, \hat{m} \in \hat{\mathcal{M}}$
2: Sort $\tilde{Q}^{A'_{\hat{m}}}$ in descending order
3: **for all** $A'_{\hat{m}} \in \mathcal{M}$ **do**
4: **while** $\tilde{Q}^{A'_{\hat{m}}} > 1$ **do**
5: $\bar{Q}^{A'_{\hat{m}}} = \bar{Q}^{A'_{\hat{m}}} + 1, \tilde{Q}^{A'_{\hat{m}}} = \tilde{Q}^{A'_{\hat{m}}} - 1$
6: Remove the corresponding qubits
7: **end while**
8: **end for**
9: Find the maximum $\tilde{Q}^{A'_{\hat{m}}} < 1, \forall A_{\hat{m}} \in \mathcal{A}_{\hat{m}}, \hat{m} \in \hat{\mathcal{M}}$ that satisfies $A'_{\hat{m}}$ is feasible
10: Branch-and-bound2 ($\tilde{Q}^{A'_{\hat{m}}}$, $\bar{Q}^{A'_{\hat{m}}}$)

Algorithm 5 Branch-and-Bound2 Algorithm

Input: $\tilde{Q}^{A'_{\hat{m}}}, \bar{Q}^{A'_{\hat{m}}}$
Output: $Q^{\dagger}_{A'_{\hat{m}}}$

1: Find the maximum $\tilde{Q}^{A'_{\hat{m}}}$, where $A'_{\hat{m}}$ is a feasible and unmarked path
2: **if** Found such $\tilde{Q}^{A'_{\hat{m}}}$ **then**
3: $\bar{Q}^{A'_{\hat{m}}} = \bar{Q}^{A'_{\hat{m}}} + 1$, remove the corresponding qubits
4: Branch-and-bound2 ($\tilde{Q}^{A'_{\hat{m}}}$,$\bar{Q}^{A'_{\hat{m}}}$)
5: $\bar{Q}^{A'_{\hat{m}}} = \bar{Q}^{A'_{\hat{m}}} - 1$, add the corresponding qubits
6: Mark the path $A'_{\hat{m}}$
7: Branch-and-bound2 ($\tilde{Q}^{A'_{\hat{m}}}$,$\bar{Q}^{A'_{\hat{m}}}$)
8: Unmark the path $A'_{\hat{m}}$
9: **else**
10: Compare $\bar{Q}^{A'_{\hat{m}}}$ and $Q^{\dagger}_{A'_{\hat{m}}}$, update $Q^{\dagger}_{A'_{\hat{m}}}$ if necessary
11: **end if**

satisfied in most of cases. In simulation results of [38], the gap between the optimal results and the ones from the branch-and-bound algorithm is 0. One exceptional case is that when the total number of variables is 57404 in the linear programming, the gap ratio between the optimal and the branch-and-bound algorithm is 10.7% since the algorithm is terminated when the largest running time is reached. □

V. Expected Throughput Maximization of STEP II

A. Algorithm Design

In STEP I, we have determined the maximum quantum-user pairs number that can be served by the network and selected one major path for each of them. We then reserve the qubits in the network assigned for the major paths. In STEP II, we aim to maximize the expected throughput of selected quantum-user pairs in STEP I in the residual graph by optimizing the qubits assigned to each path in Problem $\mathbf{S}_2$. The updated formulation of Problem $\mathbf{S}_2$ is,

$$\begin{aligned}
\textbf{Problem } \mathbf{S}_2: \max_{Q^{A'_{\hat{m}}}} & \sum_{\hat{m}=1}^{\hat{M}} \sum_{A'_{\hat{m}} \in \mathcal{A'}_{\hat{m}}} P_{A'_{\hat{m}}}, \\
\text{subject to } & A'_{\hat{m}} \in \mathcal{A'}_{\hat{m}}, \forall \hat{m} \in \hat{\mathcal{M}}, \\
& Q^{A'_{\hat{m}}} \in \mathbb{N}, \ \forall A'_{\hat{m}} \in \mathcal{A'}_{\hat{m}}, \ \hat{m} \in \hat{\mathcal{M}}, \\
& 0 \le Q^{A'_{\hat{m}}} \le \frac{\hat{Q}^i}{2}, \quad \forall A'_{\hat{m}} \in \mathcal{A'}_m, \\
& \hat{m} \in \hat{\mathcal{M}}, \ \forall v_i \in \overline{\mathcal{V}}, \\
& \sum_{\hat{m} \in \hat{\mathcal{M}}} Q^{A'_{\hat{m}}} |v_i \cap A'_{\hat{m}}| \le \frac{\hat{Q}^i}{2}, \\
& \forall v_i \in \overline{\mathcal{V}}.
\end{aligned}$$

Problem $\mathbf{S}_2$ is an integer optimization problem which is NP-Hard [34]. The formulation of Problem $\mathbf{S}_2$ is similar with Problem $\mathbf{S}_1$ except constraints (3) and (4). Hence, we modify algorithms in STEP I to address the problem in STEP II. The path set of Problem $\mathbf{S}_2$ is constrained by the newly constructed path set $\mathcal{A}'$.

First, we relax $Q^{A'_{\hat{m}}} \in \mathbb{N}$ to be a continuous non-negative real number. The relaxed problem of Problem $\mathbf{S}_2$ is denoted as Problem $\hat{\mathbf{S}}_2$. Problem $\hat{\mathbf{S}}_2$ is a continuous linear programming which can be solved by the standard linear programming method [35], [40]. Let $\tilde{Q}^{A'_{\hat{m}}}$ denote the solution solved from Problem $\hat{\mathbf{S}}_2$. $\tilde{Q}^{A'_{\hat{m}}}$ could be fractional which is not feasible. Therefore, we design an integer recovery algorithm to derive the feasible integer solution.

Let $Q^{A'_{\hat{m}}\dagger}$ denote the recovered integer solution, and $\bar{Q}^{A'_{\hat{m}}}$ denote the temporary integer solution iterated in Algorithm 4 and Algorithm 5.

Algorithm 4 first determines $\tilde{Q}^{A'_{\hat{m}}}$ that is equal or greater than 1. $\bar{Q}^{A'_{\hat{m}}}$ equals to the integer part of $\tilde{Q}^{A'_{\hat{m}}}$ solved from Problem $\hat{\mathbf{S}}_2$. The main difference needed to be dealt in Algorithm 4 compared with Algorithm 2 is that the range of $\tilde{Q}^{A'_{\hat{m}}}$ is $[0, \frac{Q_i}{2}]$ instead of $[0, 1]$. This indicates that $\bar{Q}^{A'_{\hat{m}}}$ can be added greater than 1 in process (5th row in Algorithm 4).

Then, we use Algorithm 5 algorithm to deal with the remaining fractional part in which $\tilde{Q}^{A'_{\hat{m}}} < 1$, and try to recover the feasible integer solution from this part. The algorithm is presented in Algorithm 5.

Algorithm 5 is similar but simpler compared with Algorithm 3 because Problem $\mathbf{S}_2$ has no constraint to limit the maximum path number of one quantum-user pair. For each iteration, the algorithm finds a feasible path for iteration and decides whether to occupy the current path. The algorithm marks the path to avoid repeatedly accessing the same path. When there are no paths to continue the iteration, Algorithm 5 updates the optimal solution $Q^{\dagger}_{A'_{\hat{m}}}$ when given $\tilde{Q}^{A'_{\hat{m}}}, \bar{Q}^{A'_{\hat{m}}}$.

B. Performance Analyses and Discussion

1) Performance Analyses:

Theorem 2: Algorithm 4 is an approximation algorithm to Problem $\mathbf{S}_2$ with path set $\mathcal{A}'$, and it achieves an approximation ratio of 2.

Similar to Algorithm 2, Algorithm 4 is an approximation algorithm with a ratio of 2. We do not provide the detailed proof of Theorem 2 since the proof is almost the same as Theorem 1, which is based on the assumption that the summation of the fractional part in the solution is less than the summation of the integer part.

Theorem 3: The output of Algorithm 5 $\{Q^{A'_{\hat{m}}\dagger}, \forall A_{\hat{m}} \in \mathcal{A}_{\hat{m}}, \hat{m} \in \hat{\mathcal{M}}\}$ is the optimal solution of Problem $\mathbf{S}_2$ given the integer part, i.e., $\tilde{Q}^{A'_{\hat{m}}}, \bar{Q}^{A'_{\hat{m}}}$.

Proof: Algorithm 5 is a brute-force algorithm to enumerate all possible solutions for selecting paths in the remaining

graph. When there is an optional path, the algorithm will enumerate both choices, which are providing qubits for 1-width or skipping the path. Finally, the algorithm would enumerate all possible solutions and obtain the optimal solution. □

2) Discussion:

We do not apply Algorithm 5 directly to solve Problem $\mathbf{S}_2$, and Algorithm 5 only deals with the factional part of the results from Algorithm 4 where the integer part is kept as the solution. If we apply Algorithm 5 directly to address Problem $\mathbf{S}_2$, the time complexity will be extremely high due to the large number of branches that need to be searched from the start point. Therefore, keeping the integer part and then dealing with the fraction part can greatly reduce the time complexity.

Algorithm 3 and Algorithm 5 are similar. Their basic idea is to enumerate as many branches as possible to obtain a solution with the best performance among all branches. Algorithm 3 makes decisions for each pair to choose a path, and the algorithm picks at most two paths when there are multiple branches when enumerating solutions. The size of the possible solution set is too large for the algorithm to enumerate all possible cases, thus the algorithm picks two paths with the largest x values. Algorithm 5 makes decisions for each path to choose to allocate qubits for one more width or not, which is a binary selection, and it would finally traverse all possible cases. Since the integer part of the qubits assignment has already been removed in the graph, the size of possible solutions to Algorithm 5 is possible to be traversed.

Algorithm 4 can be implemented as an independent algorithm to maximize the network expected throughput without considering selected quantum-user pairs in STEP I. We conduct simulations about Algorithm 4 to maximize expected throughput directly in Section VII and the results reveal that Algorithm 4 outperforms existing works.

VI. Recovery Path Algorithm Design in *Online Stage*

A. Insights

In this section, we propose algorithms to address the routing problem in *Online Stage* of the entanglement process. In Section IV and Section V, we present algorithms to deal with *Offline Stage*. However, the entanglement process is probabilistic. Some paths may fail to be successfully entangled because of the failed entanglement between switches or the swapping process inside the switches. Therefore, the goal of the online process is to utilize the available qubits in switches to construct recovery paths within the remaining time of the entanglement duration T.

Two primary challenges arise when constructing recovery paths during the *Online Stage*.

- First, each switch can only access its nearby switches' entanglement information including the link state and currently available qubits, i.e., K hops near it. This is because switches communicate with each other through the traditional Internet, which can generate a relatively high transmission delay compared with the entanglement duration T. Therefore, it is impossible for all switches to send their entanglement states to the cloud, and then let the cloud decide the routing design in a centralized manner like in *Offline Stage*. The switch needs to determine the swapping and the entanglement to recover the path in an online manner with only limited nearby switches' state information.
- Second, in *Online Stage*, the majority of qubits in switches have been utilized for the entanglement in *Stage I*. Only a very limited number of qubits are left for the path recovery in *Online Stage*.

In this section, we aim to answer the following question: *how to design the recovery path and swapping policy in Online Stage?*

There are two ways to construct a recovery path set. In the first method, when entanglement links fail, switches need to design and build recovery paths instantly based on the online information during the remaining duration of T. However, it is extremely challenging to determine a delicate recovery path set in such a short time with promising performance. The second method is to pre-calculate the recovery path set in an offline manner which includes all possible recovery paths for every switch in every major path selected in *Offline Stage*. The cloud can calculate the recovery path set at the beginning of *Online Stage*, and let switches decide the swapping policy by selecting recovery paths directly.

In this paper, we select the second method to determine the recovery path set. The reasons are as follows. First, the current quantum network is time-sensitive and only has a very limited entanglement duration (a few seconds). It is almost impossible to jointly determine the recovery path set and the swapping policy in *Online Stage* in such a small duration not mention to considering the communication delays and so on. Second, similar to *Offline Stage*, the cloud has enough computational ability, and the network has enough time to calculate routing for the entanglement.

Therefore, at the beginning of *Online Stage*, the cloud will pre-calculate the recovery path set for switches in every path selected in *Offline Stage* and send the related sub-recovery path set to each switch. Then, the quantum network will generate the entanglement for quantum user pairs. When some paths fail to be entangled, the switches will make the swapping policy in the remaining entanglement duration by selecting feasible paths from the recovery path set.

In the remaining section, we first propose an algorithm to derive the recovery path set. Then, we determine the swapping policy of individual switches to recover failed entanglement paths.

B. The Recovery Path Algorithm

We consider a random path $A = \{v_0, v_1, \ldots, v_l\}$ as a general case to build the recovery path.

First, switches need to compute possible recovery paths. A recovery path $Ar = \{vr_0, vr_1, \ldots, vr_{l'}\}$ is available, when $vr_0 = v_j, vr_l = v_k$ are two different switches in path A. Each switch v_i in the path except v_l can pre-compute the possible recovery paths for edges v_i, v_{i+1} after obtaining the major paths. Although switches need to obtain those paths by traversing the graph, the computational complexity is still acceptable. They only need to search recovery paths so that all switches in the paths are in $K-$hop distance, and the number

Algorithm 6 Search Recovery Paths Algorithm

Input: $\mathcal{G} = (\overline{\mathcal{V}}, \mathcal{E})$, current switch v_i, path $A = \{v_0, v_1, \ldots, v_l\}$
Output: $\mathcal{A}_r$

```
function SEARCH(v, A)
    for all (v, v') ∈ E & v' is in K−hop distance from all switches in A do
        if switch v' has at least 1 qubit & is a switch in {v_{i+1}, ..v_l} then
            A_r = A_r ∪ (A ∪ v')
        else if switch v' has at least 2 qubits & v' ∉ A then
            SEARCH(v', A ∪ v')
        end if
    end for
end function
A_r = ∅
for all h ∈ [0, i − 1] do
    if v_h remains more than 1 qubit & dist(v_i, v_h) ≤ K & dist(v_{i+1}, v_h) ≤ K then
        SEARCH(v_h, ∅)
    end if
end for
```

of remaining qubits is limited. Therefore, it is possible for switches to compute possible recovery paths in time. Switches can apply the depth-first search method [41] to compute the distance between any two switches.

Second, switches need to determine if the recovery path Ar is possible to recover path A when it is disconnected. Path A is disconnected when the entanglement of one edge of the path is failed, i.e. the entanglement of two switches $v_i, v_{i+1} \in A$ is failed. When v_i, v_{i+1} is disconnected, $j \leq i < k$, then Ar that connects v_j and v_k is possible to recover path A.

Third, all switches in the path A between v_j and v_k have to know the information that the recovery path Ar is possible to recover path A. Therefore, all switches in Ar need to be in $K-$hop distance from v_j and v_k. Besides, each switch in Ar needs to know all of the other switches in the recovery path, thus each switch in Ar needs to be in $K-$hop distance from any other switches in Ar.

Through Algorithm 6, switches can find the recovery paths. Each switch v needs to find possible recovery paths $\mathcal{A}_r$ for each major path A that includes v. Assume that v is $v_i \in A$, the mutual distance between v_i, v_{i+1} and all switches on the recovery path do not exceed $K-$hop. Moreover, each switch in the path except start and end switches should have at least 2 qubits, while start and end switches should have 1 qubit.

For each path, the time complexity of Algorithm 6 is $O(l|\mathcal{E}|)$. For each switch in the path, the algorithm may traverse the whole network once. Practically, the real run time is less than the time complexity bound, because the limit of $K-$hop distance restricts the distance of the visited switch when traversing the network.

C. The Swapping Policy

Since resources in the network may not be enough to recover all failed paths, switches need to recover the entanglement path by selecting feasible paths from the recovery path set. The final entanglement path may be different from the original major path, thus we need to consider the swapping process to correctly recover the path. In this section, we will elaborate on the swapping policy about how to recover the entanglement.

There may be multiple quantum-users pairs with several entanglement paths going through one switch. When more than one path failed to be entangled at the same link connected directly with the switch, the switch needs to set the priority for failed paths to determine which path should be recovered when qubits of the switch are not enough to recover all failed paths. We set the priority for the switch to recover failed paths for quantum-user pairs in the following order based on the goal which is to serve quantum-user pairs as many as possible and maximize their expected throughput. Here, the priority order is only related to the online information available to the switch.

1) The quantum-user pair with exact one 1-width entanglement path over the network which passes through the switch. In STEP I and STEP II, a quantum-user pair may be only assigned with only one path whose width is 1. It indicates that once the path failed to be entangled, the quantum-user pair cannot be entangled. Therefore, this type of quantum-user pair has the highest priority for switches to build recovery paths through swapping.
2) The quantum-user pair with larger expected throughput. For the rest of the quantum-user pairs, the switch will utilize the network information a few hops near it to maximize the expected throughput as illustrated in STEP II. Although different recovery paths correspond to different major paths, switches do not consider the differences between different major paths. The reason is that switches only know the information in a $K-$hop distance, and the entanglement situation out of this range is not accessible. If switches consider the corresponding major paths, they may obtain different throughput values and cannot reach a consensus on the priority of recovery paths.

After setting the priority for quantum-user pairs for the single switch, each switch connected with failed entanglement paths needs to determine its swapping policy. Therefore, the entire entanglement path can be recovered for quantum-user pairs. Without loss of generality, we discuss each pair of switches with exactly one major path individually. For each entanglement path, except for the quantum users at both ends, each intermediate switch needs to provide at least one qubit for each adjacent switch.

We consider a switch $v \in \overline{\mathcal{V}}$ that has at least one failed entailment link and the switch only belongs to one 1-width major path. When the switch belongs to a major path, it may belong to at most two recovery paths. We state that a switch may only belong to at most two recovery paths that correspond to this major path. For each recovery path, the switch in the major path has to let a qubit entangle with another switch in the recovery path, instead of a switch in the major path. An intermediate switch provides two qubits for a 1-width major path. Thus, each switch can only belong to at most two recovery paths that correspond to the major path. We discuss all possible cases one by one.

1) **The switch belongs to exactly one path**. The switch can make the swapping decision that entangles the two

qubits on the same path. In specific, the switch provides a qubit for each neighbor in the path and entangles these two qubits.

2) **The switch belongs to a major path and a recovery path**. This situation implies that one side of the major path cannot be entangled. Therefore, the switch will entangle the other side of the major path and the recovery path. The switch provides a qubit for the available neighbor on the major path and the recovery path and entangles these two qubits.
3) **The switch belongs to a major path and two recovery paths**. It indicates that both sides of the major path cannot be entangled. In this case, the switch will entangle two recovery paths.

The recovery path A_v at switch v connects switch v with $v+1$ from another new path instead of the major one, which indicates that the path A_v may have overlapping switches with the major path A. To fully build the entire entanglement path for the quantum-user pairs, we take an exclusive-or (xor, $\oplus$) operation between A_v and A. The swapping policy for the single switch mentioned above can be implemented one by one at the same time for switches to recover paths.

Since each switch can only have access information a few hops near it in *Online Stage*, the qubits allocated to a path from different switches along it may not be consistent, especially for the recover links. To evaluate the expected throughput of a path A with failed entanglement links in *Online Stage*, the number of parallel quantum channels in path A, i.e., Q^A, is equal to half of the minimum number of quantum bits assigned to the path by any switch.

The time complexity of making swapping decisions for one major path is $O(|\overline{\mathcal{V}}|)$. Each switch traverses the whole path and makes decisions based on the mentioned three cases. After traversing, the switch traverses the whole path and takes exclusive OR operations to remove duplicated switches in the path. Therefore, each switch traverses each major path twice with $O(|\overline{\mathcal{V}}|)$ complexity.

VII. Simulation Results

In this section, we will introduce our simulation results. We implement proposed algorithms and compare the performance of our proposed algorithms with existing works. We generate different data on multiple variables to enhance the confidence of the simulations. We mainly focus on two measurements of the simulations, the number of quantum-user pairs that can be served by the network, and the expected throughput.

A. Network Generation

To show the differences between the performance of our proposed algorithms and other existing works, we design controlled experiments under different network parameters. We generate network sets with standard parameters by default first and test various parameters later. The number of switches is set as 100 and the number of quantum-user pairs is set as 20. The total count of edges is defined by the average degree D of nodes, which is set to be 10. The number of qubits in each switch is 4. The successful swapping rate is 0.9.

We randomly generate each network in the standard test network set as follows. The area of the quantum network is set as $10k \times 10k$ unit square, each unit could be 1 kilometer [18]. The switches and quantum-user pairs are randomly placed in the area. The edge generation follows the work [42]. Quantum-user nodes do not connect with other quantum-user nodes directly, and they are connected with switches directly. The length of each edge is less or equals $\frac{50}{\sqrt{N}}$, where N is the number of switches. The edge capacity does not have a limitation according to our assumption in the model. Considering the randomness of the network topology, we generate 10 random networks as a set and take the average value of the measured values, i.e., the expected throughput and the number of quantum-user pairs in the network that can be served.

B. Algorithm Benchmarks

Our routing design is denoted as MULTI-R. To further show the performance of our recovery path algorithm in *Online Stage*, the results denoted as MR-REC consist MULTI-R and the recovery path algorithm. Since our recovery path algorithm does not increase the number of selected quantum-user pairs in Step I, we do not show the number of selected pairs computed by MR-REC in our following figures. We compare our algorithms with the following algorithms:

- FER [19]: First it sorts quantum-user pairs in the descending order of the expected throughput, then selects the pair with the largest expected throughput until no feasible paths exist.
- Q-PASS [18]: Q-PASS is a similar greedy algorithm with FER that uses $\sum \frac{1}{p_{i(i+1)}}$ as the routing metric, where $p_{i(i+1)}$ is the successful entanglement rate of edge $e_{i(i+1)}$. It indicates the summation of each link creation rate in a path.
- BASELINE-1(B1): we use the number of hops of a path (i.e., l in (1)) as the evaluation metrics, and run the greedy selection similar with Q-PASS.
- ALG-4: We skip STEP I and implement Algorithm 4 directly over the path set $\mathcal{A}'$ to maximize the network expected throughput.

C. Performance Evaluation

1) Network Generation Methods: Different from the method we describe in Section VII-A, we generate two different network sets by two different methods. One method is Watts-Strogatz [43]. This method generates networks that reveal the properties of some real communication networks. The other method is to generate the networks by the power-law random graph [44]. This method can generate scale-free power-law random graphs that follow the topology of complex "real-world" networks.

Figure 5 shows the performances of the algorithms in graphs generated by different methods. Figure 5a shows that MULTI-R can serve 20 quantum-user pairs on all different graphs. It shows that our algorithms can serve as many quantum-user pairs as possible on different constructed networks. Other

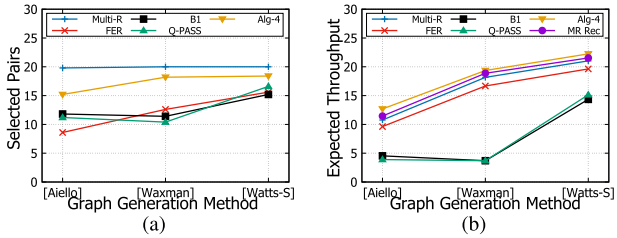


Fig. 5. (a) The selected quantum-user pairs with different network generation methods. (b) The expected throughput with different network generation methods.

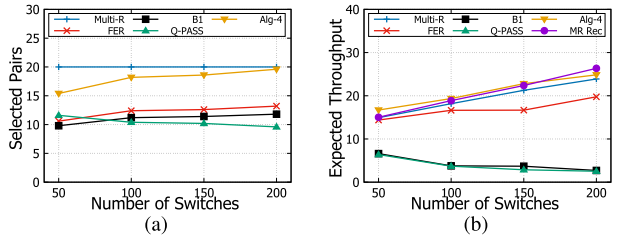


Fig. 6. (a) The selected quantum-user pairs with different numbers of switches. (b) The expected throughput with different numbers of switches.

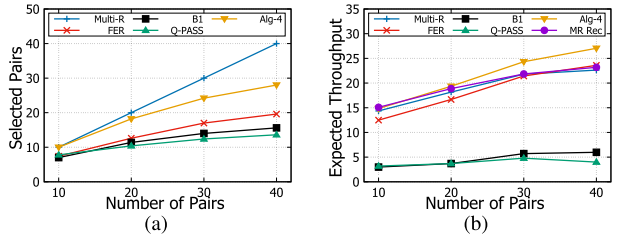


Fig. 7. (a) The selected quantum-user pairs with different numbers of pairs. (b) The expected throughput with different numbers of pairs.

algorithms cannot serve all 20 quantum-user pairs and average less than 9 selected pairs on the power-law random graph. Figure 5b shows that MULTI-R, MR-REC improves around 10%, 13% throughput more than FER. B1 and Q-PASS perform worse than the other three algorithms in all networks.

2) The Number of Switches: In Figure 6, we generate networks with 50, 100, 150, and 200 switches, when other parameters keep the same. Figure 6a shows that the number of selected pairs of FER increases with the number of switches. The reason is that FER can serve more pairs on the network with more resources. The results of Q-PASS and B1 do not change significantly. Our MULTI-R can serve all 20 quantum-user pairs in all test networks. It implies that our algorithms always consider serving more quantum-user pairs, even if the resources in the network are limited. In Figure 6b, the advantage of MULTI-R and MR-REC are more pronounced as the number of switches increases. The throughput of MR-REC improves 4%, 13%, 33%, 55% more than FER on graphs with 50, 100, 150, and 200 switches. A network with more switches has more resources, thus MR-REC has more opportunities to detect recovery paths to improve throughput. B1 and Q-PASS have less throughput when switches increase because their routing metrics decrease significantly when the size of the network is larger. It is hard for the algorithms to detect paths with high throughput in a large network.

3) The Number of Quantum-User Pairs: Figure 7 demonstrates the difference between algorithms on graphs with

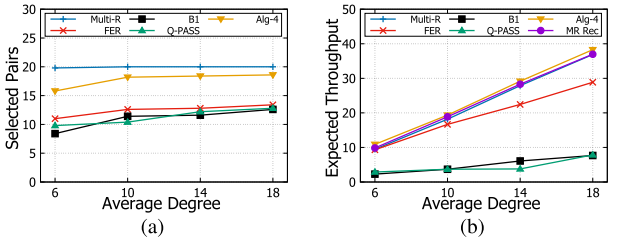


Fig. 8. (a) The selected quantum-user pairs with different average degrees of nodes. (b) The expected throughput with different average degrees of nodes.

different quantum-user pairs. As Figure 7a shows, MULTI-R can serve all quantum-user pairs in all test cases. Compare with FER, the number of served pairs is improved by 35%, 59%, 76%, and 104%. When the number of quantum-user pairs increases, FER serves less proportion of pairs. B1 and Q-PASS serve similar number of quantum-user pairs. Figure 7b shows a trade-off between the number of served pairs and throughput. The throughput of MULTI-R and MR-REC is higher than the throughput of FER when the number of quantum-user pairs is lower than 40, and FER has higher throughput when the number of pairs is 40. This phenomenon demonstrates that MULTI-R and MR-REC sacrifice throughput to improve service point pairs. The throughput of B1 and Q-PASS increases with the number of pairs because it is easier for these algorithms to find paths between more pairs.

4) The Average Degree of a Switch: Figure 8 shows the impact of the average degree of a switch. A larger average degree indicates more edges in the network. Figure 8a shows that MULTI-R serves all quantum-user pairs in all cases. FER, B1 and Q-PASS can serve more pairs when the degree increases. Figure 8b shows throughput variation as degree changes. Compare with the throughput in networks with degree 6, MR-REC improves 92%, 194% when the degree is 10, 14. Compare with FER, MR-REC improves the throughput by 6%, 13%, 26% when the degree is 6, 10, 14. The throughput of B1 and Q-PASS increases with the degree but is much lower than the other algorithms.

5) The Number of Qubits in a Switch: We adjust the number of qubits in each switch, and Figure 9 shows the results. Figure 9a and 9b show that, for all algorithms, the number of served pairs is the same, and the throughput grows proportionally to the number of qubits in a switch. The reason is that when the algorithms are routing in the same network settings, they always find the same routing schedule, and the throughput is directly affected by the number of qubits.

6) Successful Swapping Rate: In Figure 10, we test the effect of different successful swapping rates of switches. The algorithms perform similarly in served quantum-user pairs, as shown in Figure 10a. MULTI-R can serve 20 pairs, while other algorithms can serve 10 to 13 pairs. Figure 10b shows that the throughput of all algorithms increases with the success rate, which is intuitive. Both MULTI-R and MR-REC have higher throughput values and increments than MULTI-R, and MULTI-R is also greater than B1 and Q-PASS.

7) Quantum Link Successful Entanglement Rate: We test different quantum link successful entanglement rates (i.e., p)

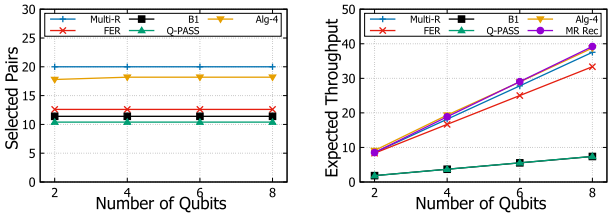


Fig. 9. (a) The served quantum-user pairs with different numbers of qubits in each switch. (b) The expected throughput with different numbers of qubits in each switch.

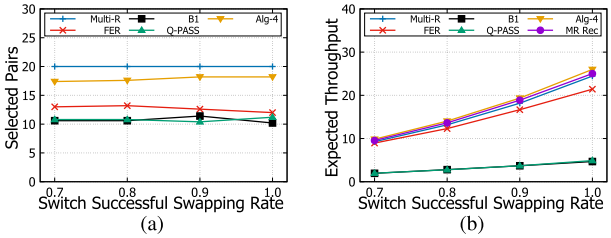


Fig. 10. (a) The served quantum-user pairs with different successful swapping rates. (b) The expected throughput with different successful swapping rates.

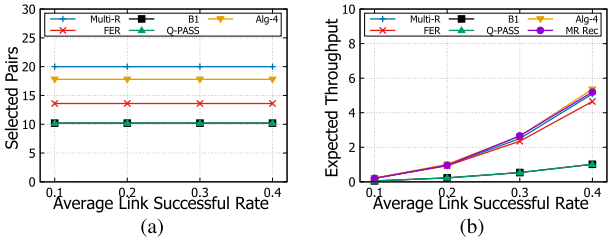


Fig. 11. (a) The served quantum-user pairs with different successful link entanglement rates. (b) The expected throughput with different successful link entanglement rates.

TABLE II

THE TIME COST DIFFERENCE BETWEEN THE OFFLINE STAGE AND ONLINE STAGE OF PROPOSED ALGORITHMS

No. Switches	Offline Stage	Online Stage
50	74162ms	2ms
100	80790.2ms	139.6ms
150	102720.6ms	768ms
200	110737.6ms	2373.5ms

as shown in Figure 11. All links have the same p to avoid the randomness brought by the network generation. When p grows from 0.1 to 0.4, the number of served quantum-user pairs is almost the same. The network expected throughput has an evident improvement up to 775%. MULIT-R still outperforms other algorithms.

8) Running Time of Online Stage: To show the running time, we compare the time cost of MULTI-R in *Offline Stage* (MULTI-R) and the proposed recovery algorithm in *Online Stage*. Table II shows the time cost differences in different number switches. As the number of switches increases, the *Online Stage* costs more time than *Offline Stage*, and the increment of *Offline Stage* is higher than *Online Stage*. Due to recent advancements in long-lived quantum memories, the entanglement duration time T can now last up to one hour [45]. The proposed recovery algorithm in *Online Stage* can be completed during the entanglement duration time T.

9) Fairness: To evaluate the fairness of algorithms, we employ evaluation methods similar to those outlined in reference [18] when referring to discussing fairness, i.e., the average number of served quantum user pairs and the average expected throughput of each served quantum pair. For the average number of served quantum user pairs, our proposed algorithms outperform existing ones as stated before.

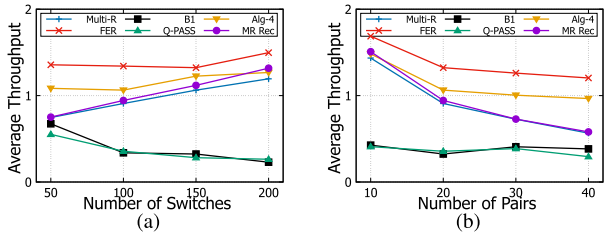
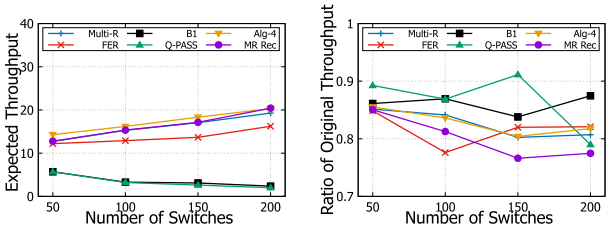


Fig. 12. (a) The average throughput with different numbers of switches. (b) The average throughput with different numbers of pairs.

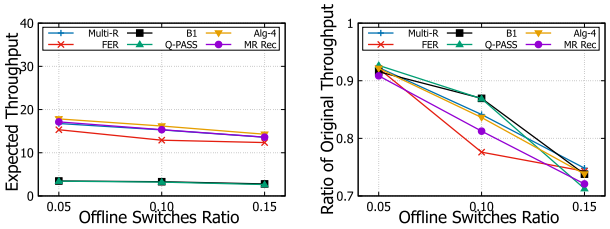
The average expected throughput is calculated by dividing the total throughput by the number of served pairs. We calculate the average expected throughput to illustrate the fairness of the algorithms from another aspect. Figure 12 shows the average expected throughput for different numbers of switching and entanglement user pairs. FER has the highest average throughput, higher than ALG-4, MULTI-R and MR-REC. The reason is that FER has a lower number of service pairs and a higher average throughput per service pair. For the same reason, the average expected throughput of ALG-4 is higher than that of MULTI-R and MR-REC. In conclusion, FER and ALG-4 have higher throughput but sacrifice fairness. MULTI-R and MR-REC balance fairness with high throughput. B1 and Q-PASS have lower average throughput, therefore they perform worst among all algorithms.

10) Robustness: To demonstrate the robustness of the algorithm, it is important to test the network in an environment where switches or links may be broken. In this study, we will focus on the scenario where switches are broken. This is because, when a switch is broken, the links connected to the switch will also be broken. Therefore, by testing the network with broken switches, we can better evaluate the algorithm's ability to handle unexpected failures and ensure that the network can continue to function effectively even in the face of such challenges. We randomly select 10% of quantum switches to be offline and compare the performance between different algorithms. When computing entanglement paths, the algorithms do not know that some switches are offline. We remove the offline quantum switches and calculate the expected throughput. Figure 13a shows the expected throughput for networks with different numbers of switches when 10% of the switches are offline. Compared to Figure 6b, the throughput of each item is reduced, and the relationship of the performance of all algorithms is similar. To show the difference clearly, we calculated the ratio of the throughput when 10% of the nodes are offline to the throughput when they are not offline, and the results are shown in 13b. Intuitively, the reduction would be 10%, since 10% of the quantum switches are offline. However, the reduction in most cases is above 10%. The decreasing trends of ALG-4 and MULTI-R are similar and different from the results of FER. However, the throughput of all three algorithms is about 80% of the original.



(a) (b)

Fig. 13. (a) The average throughput with different numbers of switches when 10% of quantum switches are offline. (b) The ratio of the throughput when 10% of the switches are offline to the throughput when the switches are not offline.



(a) (b)

Fig. 14. (a) The average throughput with different ratios of offline switches. (b) The ratio of the throughput when some switches are offline to the throughput when the switches are not offline.

The throughput of MR-REC is much lower compared to the original because the recovery algorithm makes more attempts to reconnect paths containing offline switches. The reduction of B1 and Q-PASS is relatively low, and their throughput is about 0.87 of the original. In summary, all algorithms differ in their robustness when some switches are offline, but the drop in throughput is in the range of about 0.1 to 0.2. We then test the effectiveness of different ratios of offline switches. The test networks have 100 switches. We further test the impact of different ratios of broken switches in Figure 14. It can be concluded that the ratio of throughput drop is greater than the ratio of offline switches, but less than twice the ratio of offline switches.

11) Simulation Summary: We compare five different algorithms in three simulation networks with different variables. We summarize several simulation observations as follows.

- MULTI-R and MR-REC can serve all quantum-user pairs in all test cases. FER can serve less quantum-user pairs, but higher than B1 and Q-PASS.
- In most cases, MR-REC has the highest throughput, while MULTI-R yields slightly less throughput. FER generates throughput less than MULTI-R in most cases, but is larger than MULTI-R when the number of quantum-user pairs is large. This is because there is a trade-off between the number of served pairs and throughput. ALG-4 has the largest expected throughput among all algorithms since it applies the optimization method that can fully utilize network resources. B1 and Q-PASS yield much lower throughput than FER in most cases.
- When the resources of the networks (i.e., the number of switches, the number of average degrees, and the number of qubits) are increasing, all algorithms yield more throughput in most cases. One exceptional case is that the throughput of B1 and Q-PASS decreases as the number of switches increases because they are more difficult to choose higher throughput path combinations in large networks.
- Varying quantum parameters such as successful swapping rates and quantum link successful entanglement rates do not have an obvious impact on the served number of quantum-user pairs since the capacity of the network is not changed. The parameters that can enlarge the network capacity could let the network serve more quantum-user pairs, e.g., the average degree of a switch and the number of switches. One exceptional case is the number of qubits in a switch, where the improvement is little when the number of qubits is larger. This is because the main limitation for the served number of quantum-user pairs under the default parameter setting is the number of switches, the network cannot serve more quantum-user pairs even if the number of qubits in a switch is larger. However, a larger number of qubits can make the width of paths larger, thus increasing the network throughput.

VIII. RELATED WORK

Quantum networks and their applications have drawn great attention. Several trials for constructing real quantum networks have been conducted, such as DARPA Quantum Network [46], SECOQC Vienna QKD network [47], Tokyo QKD network [48], the mobile quantum network [7], the integrated satellites [8]. These trial networks aim to distribute quantum keys or transmit real qubits for communication. Nevertheless, due to physical and hardware limitations, the application of large-scale quantum networks, in reality, is still not broadly feasible.

A few studies have been conducted on the theoretical network layer for the future large-scale quantum network. Numerical evaluations or simulations on the virtual simulator are the main methods to justify the efficiency. Vardoyan et al. [11] studied the theoretical performance of the switch capacity and the memory occupancy distribution for a single switch with multiple quantum users. Shchhukin et al. [13] analyzed the average waiting time for a single entanglement path based on Markov chain theory. Pant et al. [14] proposed a local routing policy for independent switches both in single-flow and multi-flow. Das et al. [17] presented a routing protocol for two groups of quantum users in a Bravais lattice topology. Li et al. [15] studied the flow-based network performance in a lattice network. Reference [16] proposed a greedy routing design in ring and grid networks. These papers considered the routing design in quantum networks with special typologies. These typologies may bring the advantage for the efficient design of routing protocol but they can not fit arbitrary graphs that are more common in reality. Shi et al. [18] proposed the routing protocol in a random graph. Their protocol was to add the path one by one with the largest expected throughput. Reference [19] enhanced the performance by using the remaining qubits in the network. However, their protocol assigned too many resources for limited quantum-user pairs which may waste the network resources and limit the number of quantum-user pairs that could be served. Their algorithms were greedy-based without considering the time complexity of choosing paths set and lacked performance guarantee.

Farahbakhsh et al. [49] proposed an opportunistic routing method along the routing path to reduce the waiting time. Le et al. [50] considered a reinforcement learning approach to serving quantum-user pair requests. References [36] and [51] considered fidelity as the main limitation for the entanglement which had high-level requirements for the capacity of the network. Zhao et al. [52] proposed a segmented routing design in a room-size network. All of these works assume that quantum switches use Bell Statement Measurements for the swapping. In [53] and [54], the authors adopt a more general swapping method (i.e., n-fusion) through Greenberger-Horne-Zeilinger Measurements for the swapping.

IX. Conclusion

In this paper, we have proposed an effective routing protocol for multi-entanglement routing in quantum networks to maximize the number of quantum-user pairs and their throughput at the same time. We have formulated our goal as two sequential integer programming steps and proposed efficient algorithms both in offline and online stages with low computational complexity and performance guarantees. We have conducted simulations to show that our proposed algorithms have better performance compared with existing algorithms.

References

[1] R. Van Meter, *Quantum Networking*. Hoboken, NJ, USA: Wiley, 2014.

[2] A. S. Cacciapuoti, M. Caleffi, F. Tafuri, F. S. Cataliotti, S. Gherardini, and G. Bianchi, "Quantum internet: Networking challenges in distributed quantum computing," *IEEE Netw.*, vol. 34, no. 1, pp. 137–143, Jan. 2020.

[3] N. Gisin and R. Thew, "Quantum communication," *Nature Photon.*, vol. 1, no. 3, pp. 165–171, 2007.

[4] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd, "Quantum machine learning," *Nature*, vol. 549, no. 7671, pp. 195–202, Sep. 2017.

[5] V. Scarani, H. Bechmann-Pasquinucci, N. J. Cerf, M. Dušek, N. Lükenhaus, and M. Peev, "The security of practical quantum key distribution," *Rev. Mod. Phys.*, vol. 81, no. 3, p. 1301, Sep. 2009.

[6] R. Valivarthi et al., "Teleportation systems toward a quantum internet," *PRX Quantum*, vol. 1, no. 2, Dec. 2020, Art. no. 020317.

[7] H.-Y. Liu et al., "Optical-relayed entanglement distribution using drones as mobile nodes," *Phys. Rev. Lett.*, vol. 126, no. 2, Jan. 2021, Art. no. 020503.

[8] Y.-A. Chen et al., "An integrated space-to-ground quantum communication network over 4,600 kilometres," *Nature*, vol. 589, no. 7841, pp. 214–219, Jan. 2021.

[9] W. K. Wootters and W. H. Zurek, "The no-cloning theorem," *Phys. Today*, vol. 62, no. 2, pp. 76–77, Feb. 2009.

[10] R. Van Meter and J. Touch, "Designing quantum repeater networks," *IEEE Commun. Mag.*, vol. 51, no. 8, pp. 64–71, Aug. 2013.

[11] 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.

[12] P. Nain, G. Vardoyan, S. Guha, and D. Towsley, "Analysis of a tripartite entanglement distribution switch," *Queueing Syst.*, vol. 101, nos. 3–4, pp. 291–328, Aug. 2022.

[13] E. Shchukin, F. Schmidt, and P. van Loock, "Waiting time in quantum repeaters with probabilistic entanglement swapping," *Phys. Rev. A, Gen. Phys.*, vol. 100, no. 3, Sep. 2019, Art. no. 032322.

[14] M. Pant et al., "Routing entanglement in the quantum internet," *NPJ Quantum Inf.*, vol. 5, no. 1, pp. 1–9, Mar. 2019.

[15] C. Li, T. Li, Y.-X. Liu, and P. Cappellaro, "Effective routing design for remote entanglement generation on quantum networks," *NPJ Quantum Inf.*, vol. 7, no. 1, pp. 1–12, Jan. 2021.

[16] K. Chakraborty, F. Rozpedek, A. Dahlberg, and S. Wehner, "Distributed routing in a quantum internet," 2019, *arXiv:1907.11630*.

[17] S. Das, S. Khatri, and J. P. Dowling, "Robust quantum network architectures and topologies for entanglement distribution," *Phys. Rev. A, Gen. Phys.*, vol. 97, no. 1, Jan. 2018, Art. no. 012335.

[18] 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.

[19] S. Zhang, S. Shi, C. Qian, and K. L. Yeung, "Fragmentation-aware entanglement routing for quantum networks," *J. Lightw. Technol.*, vol. 39, no. 14, pp. 4584–4591, Jul. 2021.

[20] P. Gupta and P. R. Kumar, "The capacity of wireless networks," *IEEE Trans. Inf. Theory*, vol. 46, no. 2, pp. 388–404, Mar. 2000.

[21] E. Bastug, M. Bennis, and M. Debbah, "Living on the edge: The role of proactive caching in 5G wireless networks," *IEEE Commun. Mag.*, vol. 52, no. 8, pp. 82–89, Aug. 2014.

[22] M. Maier, *Optical Switching Networks*. Cambridge, U.K.: Cambridge Univ. Press, 2008.

[23] J. Chow, O. Dial, and J. Gambetta. (2021). IBM quantum breaks the 100-qubit processor barrier. IBM Research Blog. [Online]. Available: https://research.ibm.com/blog/127-qubit-quantum-process-or-eagle

[24] J. Koetsier. (2022). Million-qubit quantum computing? How SEEQC plans to scale quantum computers. Forbes. [Online]. Available: https://www.forbes.com/sites/johnkoetsier/2022/01/11/million-qubit-quantum-computing-how-seeqc-plans-to-scale-quantum-computers/?sh=461a8a5f5b46

[25] G. B. Xavier and G. Lima, "Quantum information processing with space-division multiplexing optical fibres," *Commun. Phys.*, vol. 3, no. 1, pp. 1–11, Jan. 2020.

[26] M. Ureña, I. Gasulla, F. J. Fraile, and J. Capmany, "Modeling optical fiber space division multiplexed quantum key distribution systems," *Opt. Exp.*, vol. 27, no. 5, pp. 7047–7063, 2019.

[27] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information: 10th Anniversary Edition*, 10th ed. New York, NY, USA: Cambridge Univ. Press, 2011.

[28] B. Coecke, "The logic of entanglement. An invitation," Dept. Comput. Sci., Oxford Univ., Tech. Rep. RR-03-12, 2003.

[29] 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, no. 26, pp. 5932–5935, Dec. 1998.

[30] A. Dahlberg et al., "A link layer protocol for quantum networks," in *Proc. ACM Special Interest Group Data Commun.*, 2019, pp. 159–173.

[31] J.-W. Pan, D. Bouwmeester, H. Weinfurter, and A. Zeilinger, "Experimental entanglement swapping: Entangling photons that never interacted," *Phys. Rev. Lett.*, vol. 80, no. 18, pp. 3891–3894, May 1998.

[32] W.-H. Zhang et al., "Experimental realization of robust self-testing of bell state measurements," *Phys. Rev. Lett.*, vol. 122, no. 9, Mar. 2019, Art. no. 090402.

[33] D. Bouwmeester, J.-W. Pan, K. Mattle, M. Eibl, H. Weinfurter, and A. Zeilinger, "Experimental quantum teleportation," *Nature*, vol. 390, no. 6660, pp. 575–579, Dec. 1997.

[34] S. Even, A. Itai, and A. Shamir, "On the complexity of time table and multi-commodity flow problems," in *Proc. 16th Annu. Symp. Found. Comput. Sci. (SFCS)*, Oct. 1975, pp. 184–193.

[35] V. Chvatal et al., *Linear Programming*. New York, NY, USA: Macmillan, 1983.

[36] K. Chakraborty, D. Elkouss, B. Rijsman, and S. Wehner, "Entanglement distribution in a quantum network: A multicommodity flow-based approach," *IEEE Trans. Quantum Eng.*, vol. 1, pp. 1–21, 2020.

[37] L. R. Ford and D. R. Fulkerson, "A suggested computation for maximal multi-commodity network flows," *Manage. Sci.*, vol. 5, no. 1, pp. 97–101, Oct. 1958.

[38] C. Barnhart, C. A. Hane, and P. H. Vance, "Using branch-and-price-and-cut to solve origin-destination integer multicommodity flow problems," *Oper. Res.*, vol. 48, no. 2, pp. 318–326, Apr. 2000.

[39] J. Y. Yen, "Finding the K shortest loopless paths in a network," *Manage. Sci.*, vol. 17, no. 11, pp. 712–716, Jul. 1971.

[40] M. B. Cohen, Y. T. Lee, and Z. Song, "Solving linear programs in the current matrix multiplication time," *J. ACM*, vol. 68, no. 1, pp. 1–39, Feb. 2021.

[41] R. Tarjan, "Depth-first search and linear graph algorithms," *SIAM J. Comput.*, vol. 1, no. 2, pp. 146–160, Jun. 1972.

[42] B. M. Waxman, "Routing of multipoint connections," *IEEE J. Sel. Areas Commun.*, vol. 6, no. 9, pp. 1617–1622, Dec. 1988.

[43] D. J. Watts and S. H. Strogatz, "Collective dynamics of small-world networks," *Nature*, vol. 393, no. 6684, pp. 440–442, Jun. 1998.

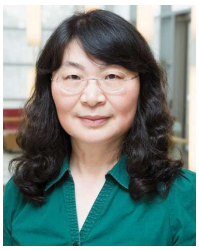
[44] D. Volchenkov and P. Blanchard, "An algorithm generating random graphs with power law degree distributions," *Phys. A, Stat. Mech. Appl.*, vol. 315, nos. 3–4, pp. 677–690, Dec. 2002.
[45] Y. Ma, Y.-Z. Ma, Z.-Q. Zhou, C.-F. Li, and G.-C. Guo, "One-hour coherent optical storage in an atomic frequency comb memory," *Nature Commun.*, vol. 12, no. 1, pp. 1–6, Apr. 2021.
[46] C. Elliott, A. Colvin, D. Pearson, O. Pikalo, J. Schlafer, and H. Yeh, "Current status of the DARPA quantum network," *Proc. SPIE*, vol. 5815, pp. 138–149, May 2005.
[47] M. Peev et al., "The SECOQC quantum key distribution network in Vienna," *New J. Phys.*, vol. 11, no. 7, 2009, Art. no. 075001.
[48] M. Sasaki et al., "Field test of quantum key distribution in the Tokyo QKD network," *Opt. Exp.*, vol. 19, no. 11, pp. 10387–10409, 2011.
[49] A. Farahbakhsh and C. Feng, "Opportunistic routing in quantum networks," 2022, *arXiv:2205.08479*.
[50] L. Le and T. N. Nguyen, "DQRA: Deep quantum routing agent for entanglement routing in quantum networks," *IEEE Trans. Quantum Eng.*, vol. 3, pp. 1–12, 2022.
[51] Y. Zhao, G. Zhao, and C. Qiao, "E2E fidelity aware routing and purification for throughput maximization in quantum networks," in *Proc. IEEE INFOCOM*, May 2022, pp. 480–489.
[52] G. Zhao, J. Wang, Y. Zhao, H. Xu, and C. Qiao, "Segmented entanglement establishment for throughput maximization in quantum networks," in *Proc. IEEE 42nd Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Jul. 2022, pp. 45–55.
[53] Y. Zeng, J. Zhang, J. Liu, Z. Liu, and Y. Yang, "Entanglement routing over quantum networks using Greenberger–Horne–Zeilinger measurements," in *Proc. IEEE 43th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, 2023.
[54] A. Patil, J. I. Jacobson, E. Van Milligen, D. Towsley, and S. Guha, "Distance-independent entanglement generation in a quantum network using space-time multiplexed Greenberger–Horne–Zeilinger (GHZ) measurements," in *Proc. IEEE Int. Conf. Quantum Comput. Eng. (QCE)*, Oct. 2021, pp. 334–345.

Yiming Zeng (Member, IEEE) received the B.Eng. degree in information engineering from Shanghai Jiao Tong University, Shanghai, China. He is currently pursuing the Ph.D. degree in computer and electrical engineering with Stony Brook University, Stony Brook, NY, USA. His research interests include edge computing, quantum networking, and computing.

Jiarui Zhang (Member, IEEE) received the B.Eng. degree in computer science and technology from Shanghai Jiao Tong University in 2017. He is currently pursuing the Ph.D. degree in computer engineering with Stony Brook University. His research interests include blockchain and mobile edge computing.

Ji Liu (Member, IEEE) received the B.S. degree in information engineering from Shanghai Jiao Tong University, Shanghai, China, in 2006, and the Ph.D. degree in electrical engineering from Yale University, New Haven, CT, USA, in 2013. He is currently an Assistant Professor with the Department of Electrical and Computer Engineering, Stony Brook University, Stony Brook, NY, USA. His current research interests include distributed control and optimization, distributed reinforcement learning, resiliency of distributed algorithms, epidemic and social networks, quantum computing, and networking. He is an Associate Editor of the IEEE TRANSACTIONS ON SIGNAL AND INFORMATION PROCESSING OVER NETWORKS.

Zhenhua Liu (Member, IEEE) received the Ph.D. degree in computer science from the California Institute of Technology, under the supervision of Adam Wierman and Steven Low. He is an Associate Professor of operations research with the Department of Applied Mathematics and Statistics and also affiliated with the Department of Computer Science, Stony Brook University (SUNY at Stony Brook). His research aims to develop analytical models, theoretical results, and deployable algorithms to manage complex distributed systems with limited information and network constraints. He has helped HP design and implement the industry's first Net-Zero Energy Data Center, which was named a 2013 Computer World Honors Laureate. He received the IBM 2020 Global University Program Academic Award, for his research on resource management of AI/ML systems. His research work is widely cited and recognized in academia, including the Best Paper or Best Student Paper Awards at IEEE INFOCOM, ACM GREENMETRICS, and IEEE Green Computing Conference. He also received the 2021 ACM SIGMETRICS Rising Star Research Award, the 2021 ACM SIGMETRICS Test of Time Award, the Pick of the Month Award by IEEE STC on Sustainable Computing, the SPEC Distinguished Dissertation Award (honorable mention), the NSF CAREER Award, and several Excellence in Teaching Awards.



Yuanyuan Yang (Fellow, IEEE) received the B.Eng. and M.S. degrees in computer science and engineering from Tsinghua University, Beijing, China, and the M.S.E. and Ph.D. degrees in computer science from Johns Hopkins University, Baltimore, MD, USA. She is a SUNY Distinguished Professor of computer engineering and computer science with Stony Brook University, NY, USA. Before, she was the Program Director with the National Science Foundation. She has more than 30 years of combined experience in parallel computing, cloud computing, optical networking, and quantum computing. She has published more than 500 papers in these areas, including three best paper awards and six best paper runner-ups. She is a Fellow of the National Academy of Inventors (NAI). She has also served as the general chair, the program chair, or the vice chair for several major conferences and a program committee member for numerous conferences. She is currently the Editor-in-Chief of IEEE TRANSACTIONS ON CLOUD COMPUTING and an Associate Editor of IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS and *ACM Computing Surveys*. She has served as an Associate Editor-in-Chief for IEEE TRANSACTIONS ON CLOUD COMPUTING, the Associate Editor-in-Chief and an Associate Editor for IEEE TRANSACTIONS ON COMPUTERS, and an Associate Editor for IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS.