Multi-Entanglement Routing Design over Quantum Networks Using Greenberger-Horne-Zeilinger Measurements

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

***Abstract*—Generating a long-distance quantum entanglement is one of the most essential functions of a quantum network to support quantum communication and computing applications. The successful entanglement rate during a probabilistic entanglement process decreases dramatically with distance, and swapping is a widely applied quantum technique to address this issue. Most existing entanglement routing protocols use a classic entanglement-swapping method based on Bell State measurements that can only fuse two successful entanglement links. This paper appeals to a more general and efficient swapping method, namely n-fusion based on Greenberger-Horne-Zeilinger measurements that can fuse n successful entanglement links, to maximize the entanglement rate for multiple quantum-user pairs over a quantum network. We propose efficient entanglement routing algorithms that utilize the properties of n-fusion for quantum networks with general topologies. Evaluation results highlight that our proposed algorithm under n-fusion can greatly improve the network performance compared with existing ones.**

***Index Terms*—Quantum Networks; Entanglement Routing; n-fusion Entanglement-swapping; Greenberger-Horne-Zeilinger (GHZ) Measurements**

I. Introduction

QUANTUM information science is viewed as the next scientific breakthrough that will propel scientific and economic developments for the whole society in the near future, since quantum applications have shown capabilities far beyond the traditional approaches. For instance, quantum computing algorithms have the potential for exponential speedups compared to their classical counterparts. Notable examples include Shor's algorithm [1] and the quantum linear system algorithms [2]. Furthermore, these algorithms enable information generation, storage, and transmission with levels of privacy, security, and computational power that are currently unattainable with conventional methods [3].

In the broad context of quantum information science, quantum networks are expected to be promising next-generation networks. Existing implementations include long-distance (40 kilometers) teleportation via fiber link [4], mobile quantum networks [5], and satellite-based integrated entanglement systems supporting distances over 4600 kilometers [6]. A key characteristic of quantum networks is entanglement, vital for the analysis and implementation of quantum computing and communication. Entanglement creates a unique correlation among quantum bits (qubits), such as in position or spin, that remains even over long distances.

Maintaining long-distance entanglement between qubits is essential for the functionality of quantum networks. Yet, as the distance between qubits grows, the rate of successful entanglement diminishes significantly. In the quantum network, a quantum processor is a device that manipulates qubits to perform computations and facilitate communication of network nodes. Additionally, applications such as distributed computing [7] and sensing [8] require entangling processors across nodes, where each node may host dozens of quantum processors in a warehouse-scale data center. Due to long distances and complex layouts, direct fiber links are often impractical. *Entanglement-swapping* is an important method, allowing the establishment of entanglement paths between quantum processor pairs that previously didn't share an entanglement. Some certain processors in a network, termed as "quantum switches", function as relays that facilitate end-to-end entanglement by leveraging entanglement-swapping.

This paper studies a key problem in quantum networks, called *entanglement routing*, whose goal is to *achieve efficient long-distance entanglement over a quantum network through entanglement-swapping.*

The entanglement routing problem is at the forefront of recent research, yielding several noteworthy outcomes. Various studies have introduced entanglement routing algorithms with associated theoretical frameworks, focusing primarily on specific network topologies [9]–[14]. Advancements for more generic network topologies are detailed in [15], [16]. In these existing quantum routing algorithms, however, switches are restricted to perform a classic swapping method [17]. This method employs the Bell State Measurements (BSMs). Notably, since a BSM operates using two qubits, it can only merge two entanglement links at a time. Figure 1a depicts how BSM achieves this fusion by entangling two qubits within a processor.

Recent studies [18], [19] have introduced switches capable of conducting multi-qubit joint Greenberger-Horne-Zeilinger (GHZ) projective measurements in the GHZ basis termed as n-fusion. This allows for the fusion of $n \geq 2$ successful entan-

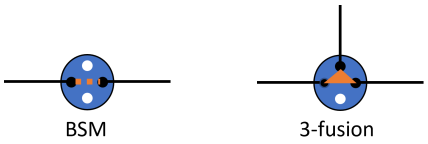
Yiming Zeng is with the School of Computing, Binghamton University, Binghamton, NY, 13902. Email: {yzeng4@binghamton.edu}

Jiarui Zhang, Ji Liu and Yuanyuan Yang are with the Department of Electrical and Computer Engineering, Stony Brook University, Stony Brook, NY, 11794. Email: {jiarui.zhang.2, ji.liu, yuanyuan.yang}@stonybrook.edu

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

‡ Both authors contributed equally to this research.

glement links simultaneously, a process termed n-fusion. For instance, Figure 1b demonstrates a scenario where a processor employs a 3-GHZ measurement, facilitating the fusion of three simultaneous entanglement links. A real experiment has been conducted for $n = 3$ [20]. Importantly, n-fusion, especially when $n \geq 3$, can be seen as an extension of the traditional entanglement-swapping method. It encompasses the conventional BSM-based method (or 2-fusion) as a specific instance. As a more general extension of BSMs, n-fusion can support a wider range of quantum applications beyond quantum state transmission, which is typically limited to $n = 2$. These applications include quantum key distribution [18], quantum telephone exchanges, multiparty cryptography, and distributed quantum computing [20]. This could significantly enhance the network's entanglement performance. However, current studies [18], [19] model the problem as a percolation problem, without delving into detailed routing algorithms. Consequently, a holistic study on quantum entanglement routing utilizing n-fusion remains notably absent in current literature.



(a) (b)

Fig. 1. (a) An example of traditional swapping under BSM measurement in the switch, where two quantum links are fused by connecting two qubits through swapping. (b) An example of 3-fusion, where 3-GHZ measurement in a switch fuses three quantum links by connecting three qubits. In both figures, the small white circle in the switch denotes free qubits that are not for the entanglement, the small black circle in the switch denotes entangled qubits, the orange line and the orange triangle show the connection between qubits to fuse quantum links, and black lines indicate the quantum links to be fused.

Addressing these identified gaps, to the best of our knowledge, this is the first work of a comprehensive entanglement routing design over a general quantum network topology with multiple quantum-user pairs where switches can take a general entanglement-swapping method, n-fusion, with new models, new metrics, and new algorithms. Our core objective is to optimize the entanglement rate within the network, essentially the expected number of quantum states shared between quantum-user pairs.

The pivotal contributions of this research are enumerated below:

1) We introduce a novel comprehensive model for entanglement routing, detailing the entanglement process wherein switches utilize n-fusion. This model serves multiple quantum-user pairs aiming to share quantum states across a general network topology, which can be easily extended to other sophisticated models such as fidelity-aware cases.
2) We define the entanglement routing under n-fusion as a new graph routing problem due to its unique quantum characteristics. We reveal that n-fusion introduces a unique graph structure between two terminals—quantum processors aiming to establish end-to-end entanglement with each other—characterized as a symmetric flow-like graph (see Section V-A). This emergence presents a fresh and uncharted graph routing paradigm, distinct from conventional graph routing methodologies and not readily addressed by their standard solutions (as elaborated in Section VI-B). This development holds significant implications not only for n-fusion in the quantum Internet realm but also signals a nuanced intersection of traditional graph routing with quantum networking.
3) Considering the unique graph structure introduced, we define the routing metric based on the n-fusion. The metric evaluates network performance, which we then optimize using algorithmic design.
4) Efficient entanglement routing algorithms are designed for the entanglement, facilitating the sharing of quantum states among quantum-user pairs.
5) Through extensive simulations, we validate the efficiency of our innovative n-fusion algorithm. When compared in the same network settings, our approach showcases up to $6\times$ enhanced performance relative to existing classic swapping algorithms. Additionally, it substantially outperforms existing n-fusion methods in terms of the network's entanglement rate.

The organization of this paper is as follows: Section II reviews related literature, and Section III outlines the quantum background pertinent to our study. Section IV describes our network model, while Section V formulates the problem. Section VI examines the problem and its associated challenges. Details of our entanglement routing algorithms are in Section VII, and our recovery algorithm is presented in Section VIII. Simulation results are discussed in Section IX, including a comparison with existing methods. Section X concludes the paper.

II. Related Works

System implementation: Several research labs and companies have constructed trial quantum networks for purposes such as quantum key distribution or real qubit transmission. Examples include the DARPA quantum system [21], the SECOQC Vienna QKD system [22], the Tokyo QKD system [23], the mobile quantum system [5], and integrated satellites [6]. However, due to hardware limitations, there are currently no large-scale quantum networks in widespread use.

Entanglement routing under BSMs: This group of studies explored the traditional swapping method based on Bell state measurements (BSMs) in quantum networks. Vardoyan *et al.* [12] investigated the theoretical performance of processor capacity and memory occupancy distribution for a single processor serving multiple quantum users. Shchhukin *et al.* [13] analyzed the average waiting time for a single entanglement path using Markov chain theory. Pant *et al.* [9] proposed a local routing policy for independent processors in both single-flow and multi-flow scenarios. Das *et al.* [14] introduced a routing algorithm for two sets of quantum users in a Bravais lattice topology. Li *et al.* [10] explored flow-based system performance in a lattice network. Chakraborty *et al.* [11] suggested a greedy routing design for ring and grid networks. These works primarily focused on routing design in quantum computing systems with specific topologies.

Shi *et al.* [15], [16] proposed routing algorithms for a random graph to maximize network throughput. These algorithms were greedy-based, selecting the path with the highest throughput until no feasible paths remained. Zeng *et al.* utilized optimization methods to simultaneously optimize network throughput and the number of served users, achieving improved performance. Studies by Chakraborty *et al.* [24] and Qiao *et al.* [25] incorporated fidelity as an entanglement constraint. Zhao *et al.* [26] proposed segmented entanglement establishment, integrating all-optical switching and quantum swapping. Zhao *et al.* [27] developed two distributed protocols for transporting quantum information in quantum data networks. Liu *et al.* [28] focused on designing entanglement protocols for communication. Ghaderibaneh *et al.* [29] examined a tree structure to determine the swapping policy. Ref. [30] and Ref. [31] aimed to maximize the throughput of multiple quantum user pairs. Li *et al.* [32] summarized the research challenges in the quantum network field. Vardoyan *et al.* [33] explored the problem of multi-path entanglement routing between two nodes in a quantum network, considering the potential for multiplexing capabilities in network links. Zeng *et al.* [34] utilized EPRs to establish connection among more than two users. Van Milligen *et al.* [35] proposed a multi-path routing protocol based on local link-state knowledge, utilizing time-multiplexed switches within a grid network.

Entanglement routing under n-fusion: Patil *et al.* [19] reached an intriguing conclusion that in a grid/lattice network where switches employed n-fusion entanglement-swapping, the entanglement rate between a pair of quantum users does not depend on the distance between them. They later extended this finding to a model incorporating a space-time multiplexed method [18]. However, this conclusion was derived from the percolation theorem [36], which applies to graphs with specific structures. Moreover, they only explored the scenario with a single pair of quantum users. Sutcliffe and Beghelli [37] extended n-fusion to enable multi-party entanglement among more than two users. Bugalho et al. [38] presented an algorithm for multipartite entanglement in noisy quantum networks, under a given distribution scheme. Clayton et al. [39] present QuARC, which adaptively clusters the network to cut latency.

III. Background

In this section, we will introduce some important quantum terminologies and mechanisms that we will use in this paper.

A. Basic Terminologies

Qubit: A qubit, the fundamental unit of quantum information, can be an electron, photon, or atomic nucleus and is defined by its quantum state [40]. Unlike a classical bit, which represents 0 or 1, a qubit can exist in a superposition of both states.

Entanglement: Entanglement is a phenomenon in which a group of qubits expresses a high correlation state that cannot be described by the classical theory of probability. n qubits can be maximally entangled as a n-GHZ state, i.e., $\frac{|0\rangle^{\otimes n}+|1\rangle^{\otimes n}}{\sqrt{2}}$. The Bell State that contains the exact two qubits can be viewed as a special case of the n-GHZ state, where $n = 2$.

B. n-fusion

In this study, we consider the entanglement-swapping technique called n-fusion, as discussed in [18], [19]. The n-fusion method is based on GHZ measurements that allow n measured qubits to be projected onto one of the 2^n GHZ states. Figure 2 provides an example where three independent states become entangled in a 6-GHZ state through GHZ measurements. When $n = 1$, the operation corresponds to a single-qubit Pauli measurement [41]. When $n = 2$, the operation is a Bell State Measurement (BSM) on two qubits.

Fig. 2. A 3-fusion in a processor over one Bell state, one 3-GHZ state, and one 4-GHZ state. 3-fusion generates a 6-GHZ state after the swapping.

In a quantum network, this technique can be applied for entanglement swapping, allowing switches to fuse n successful entanglement links simultaneously (see Figure 1b). When $n = 2$, this corresponds to traditional entanglement swapping, which uses BSMs to fuse two quantum links [17] (see Figure 1a).

C. Entanglement Process

In this subsection, we explore a generalized entanglement process that includes n-fusion entanglement-swapping for switches. Traditional swapping is a specific case within this framework. The process comprises two phases: Phase I involves preparatory steps for the upcoming entanglement, and Phase II entails implementing entanglement across optical fibers and performing quantum link fusions through entanglement-swapping within switches. Each phase is detailed below.

- *Phase I*: This phase uses a central controller to design entanglement routes offline and distribute them to the relevant switches. At the start of each entanglement cycle (i.e., the average entanglement lifetime), the central server queries all switches for their latest outcomes and uses this updated status for subsequent routing decisions. Currently, classical computing remains more effective for tasks like solving routing problems. The central device uses offline data, including quantum user details, network structure, and switch specifications, to determine the best entanglement routes given the constraints of switch capacities. These routes are then communicated to the quantum processors via classical channels to prepare for the entanglement activities in Phase II.
- *Phase II*: There are three steps in Phase II. We adopt a slotted-time model similar to Ref [18], [37], where the entire entanglement process is divided into fixed-duration time slots (cycles). Each cycle consists of three phases:
  - Synchronization and Initialization: All switches align their internal clocks at the start of each slot. The central controller then broadcasts the predefined routing paths (from Phase I) to every switch, establishing the target links for this cycle.

- Probabilistic Entanglement Generation & Local Recovery: Within the slot, each switch attempts entanglement generation on its assigned fiber links. Failures are detected immediately, but the slot is too short for global coordination. Instead, switches use only local information (success/failure of their own and neighbor links, up to a few hops) to form on-the-fly recovery paths and retry entanglement within the same slot.
- Entanglement Swapping: At the end of the slot, switches perform n-fusion (or BSMs when $n = 2$) on all successfully generated links. The resulting end-to-end entangled states are then available for the next cycle's quantum operations or higher-level protocols.

This slotted approach ensures a clear timing structure—each slot comprises generation (with local recovery) followed by swapping—while avoiding prohibitive global communication delays.

IV. Quantum Network Model

In this section, we introduce the quantum network model.

A. Network Component

We consider a general network topology where switches take n-fusion entanglement-swapping for the entanglement. We first introduce four main components of the quantum network.

1) Quantum user: A quantum user is a quantum cluster that consists of multiple quantum processors that are tightly coupled, enabling different quantum states to support a variety of applications. For example, Flamingo, a planned quantum computing cluster from IBM, will consist of 7 quantum processors, each equipped with 156 qubits [42]. A quantum user seeks to establish entanglement with another user by sharing quantum states.

The network comprises multiple quantum-user pairs. While each quantum user can utilize different qubits to simultaneously entangle with various other quantum users, any single qubit from a quantum user can participate in only one quantum state at a time. The set of quantum-users is denoted as $\mathcal{U} = \{u_i\}_{i=1}^{U}$.

2) Quantum switch: Quantum switches act as specialized quantum processors, serving as relay nodes to enable remote entanglement through the process of entanglement-swapping. While quantum users possess qubits dedicated to both computation and communication, quantum switches are outfitted solely with communication qubits and are exclusively utilized for the task of entanglement-swapping. The collection of these switches is represented by $\mathcal{V} = \{v_i\}_{i=1}^{V}$. We assume that each switch has the capability to execute a Q_{v_i}-fusion, where Q_{v_i} denotes the quantum memory capacity of switch $v_i \in \mathcal{V}$, representing the maximum number of qubits that can be simultaneously stored and involved in entanglement operations. The capacity of a switch v_i is denoted by Q_{v_i}, representing the number of quantum memory qubits it can hold. The switch can perform l-fusion operations for any $l \leq Q_{v_i}$. In practice, the success rate of n-fusion typically decreases as n increases, since manipulating and coherently measuring a larger number of entangled qubits poses greater experimental challenges. This implies that an l-fusion generally has a higher success rate than a full-capacity Q_{v_i}-fusion when $l < Q_{v_i}$.

However, quantifying the relationship between fusion size n and success probability is still difficult. The fusion process is probabilistic and influenced by platform-specific factors such as optical alignment, control fidelity, and environmental noise. Even for 2-qubit Bell-state measurements, models often use fixed success rates without capturing these physical details. Extending this to general n-fusion introduces further uncertainty, and no standard theory currently models this accurately. To keep the model tractable and consistent, we conservatively approximate all l-fusion attempts at a switch using the success rate corresponding to its full capacity Q_{v_i}. While this may underestimate the success rate for $l < Q_{v_i}$, it avoids introducing unverifiable assumptions and preserves platform independence.

3) Quantum link: The quantum link is established between two qubits from processors over the optical fiber that connects two switches to support the entanglement. When a quantum link is successfully established, the neighboring switches share a Bell pair given by $\frac{|00\rangle+|11\rangle}{\sqrt{2}}$ [40]. The qubits used in the link will be occupied and thus cannot be used for other quantum links simultaneously. The probability of achieving successful entanglement across the quantum link is influenced by the length of the link and the material properties of the optical fiber. It is described as $p = \mathbf{e}^{-\alpha L}$, where α is a constant related to the material, L is the length of the quantum link between the switches and $\mathbf{e}$ is Euler's number with an approximate value of 2.71828.

Considering that multiple optical fiber cables might be present between switches, and each cable has individual cores that can act as quantum links, multiple quantum links can exist on the same edge for a specific quantum state, referred to as a quantum channel. Moreover, several quantum channels for different quantum states can simultaneously exist on the same edge. These optical fibers are also capable of transmitting classical information through the network, namely a bit which is either a 0 or 1.

4) A center server for traditional computing and communication: The center server is crucial for maintaining and relaying network information, including topology and connections, to the switches. It operates under a semi-trust (honest-but-curious) model [43], [44], faithfully executing protocols but attempting to learn all possible information. Additionally, the server handles computational tasks, such as pre-calculating routes for quantum-user pairs, to support the entanglement process.

B. Network Topology

The network can be visualized as an undirected graph represented by $G = (\overline{\mathcal{V}}, \mathcal{E})$. Here, $\overline{\mathcal{V}} = \mathcal{U} \cup \mathcal{V}$ denotes the union of quantum users and quantum switches. An edge that bridges users or switches, $v_i \in \overline{\mathcal{V}}$ and $v_j \in \overline{\mathcal{V}}$, is termed as e_{v_i,v_j}, and the edge set is denoted by $\mathcal{E} = \{e_{v_i,v_j} \subset (v_i, v_j) : v_i, v_j \in \overline{\mathcal{V}}\}$. The topology of the graph can be arbitrary. Figure 3 shows

an example of the proposed quantum network. The topology of the quantum network is relatively stable, with network information being readily available to all switches.

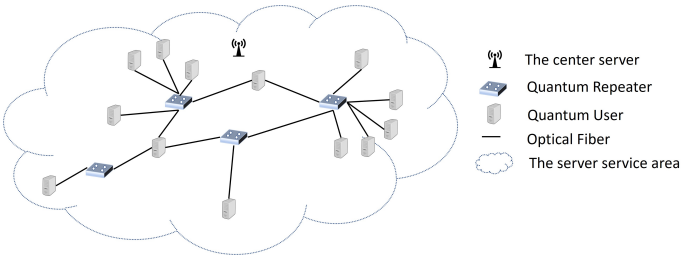


Fig. 3. An example of quantum network. The network consists of a center server, quantum switches, quantum users, and optical fibers, with nodes interconnected through optical fibers and the center server connected to each node in the network. The components are illustrated in Section IV-A.

V. Problem Statement

In this section, we begin by introducing an innovative graph structure, termed the flow-like graph, situated between two quantum users for the purpose of sharing quantum states. This structure is generated from the distinctive attributes of n-fusion which is different from the flow in classic graph theory [45]. Subsequent to that, we define the routing metric rooted in the flow-like graph. Finally, we formally formulate a graph problem.

A. The Flow-Like Graph

Considering a quantum state to be shared between two quantum users, the n-fusion technique enables a switch to fuse any number of quantum links, as long as the qubits utilized in the switches for generating these links do not surpass the switch's capacity. This leads to the creation of a connection topology between one pair of quantum users, which is referred to as a flow-like graph. The definition of this flow-like graph is presented as follows:

Definition 1. *A flow-like graph is an undirected graph $G = (S, E)$, where $S = \{U_1, U_2\} \cup V$ represents the set of nodes (including the two connected nodes U_1 and U_2 and intermediate nodes v), and $E \subseteq S \times S$ represents the set of edges. For each $v \in V$, the capacity $c_v \in \mathbb{N}_+$ satisfies the constraint: $\sum_{u \in S} e_{u,v} \leq c_v$, where $e_{u,v} \in E$ represents the edge between nodes u and v.*

In the quantum network, $\{U_1, U_2\}$ represents one quantum user pair, V denotes the set of switches, and E represents quantum links established by pairs of entangled qubits. A flow-like graph indicates the connection of two quantum users through a shared quantum state. Within the context of a flow-like graph, any node that is shared by more than one path for the same quantum state is termed a "branch node", where a path is a sequence of nodes where each node is connected to the next by an edge. A branch node has two or more child nodes in the flow-like graph; this characterization is independent of the fusion size n.

An example of flow-like graphs is shown in Figure 4, where a pair of quantum users, Alice and Bob, share two quantum states, represented by green and black dots respectively. Focusing on the quantum state represented by the black dots, we see that starting from Alice, switches are interconnected through quantum links over pairs of qubits. Within these switches, qubits are linked internally via the n-fusion process. The flow-like graph is a symmetric flow graph that establishes connections between two end nodes, and thus, inherently no direction between them. For ease of analysis, we define one end node as the source and the other as the destination. Starting from the source node, it is noteworthy that a switch can have multiple branches at once, with each branch corresponding to distinct paths, thereby establishing unique directions for the entanglement. Such switches are termed *branch nodes*. For instance, switch 3 acts as a branch node with two paths dedicated to the shared state represented by the black qubits. Conversely, switch 4, despite being a part of two paths, is not considered a branch node because these paths pertain to two distinct shared quantum states. It is essential to underscore that, for any two separate quantum states shared between the same quantum-user pair, their corresponding flow-like graphs do not overlap in terms of quantum links. This non-overlap holds true for both internal links within switches and external links connecting switches.

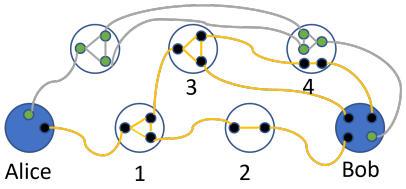


Fig. 4. Two flow-like graphs for two shared quantum states between a quantum-user pair. A Bell state denoted by qubits in green is entangled through gray links between & within switches. A 4-GHZ state, represented by black dots, is entangled via yellow links both between and within switches. In this graph, switches are labeled with index numbers, where Switch 1 and Switch 3 serve as branch nodes for this flow-like graph as they all have two different branches to their next-hop nodes.

Comparison between the Flow-like Graph and the Classic Flow: The flow-like graph differs from the classic flow in traditional graph theory due to the distinctive features of quantum mechanics from n-fusion. The key difference between a flow-like graph and flow is that a flow-like graph is not an actual flow but shares a similar topology. Specifically, a flow is directed and must satisfy the Flow Conservation principle, which states that for any node (except the source and sink), the sum of incoming flows must equal the sum of outgoing flows. In contrast, a flow-like graph is symmetric and undirected, reflecting the nature of quantum entanglement, and it does not need to satisfy Flow Conservation.

B. Routing Metrics

To quantify performance, we define the *entanglement rate* as the expected number of quantum states (including Bell states and GHZ states) generated between quantum-user pairs in the quantum network under n-fusion per cycle, where a cycle represents a time window equal to the average entanglement duration. We start from the simple case of a quantum channel and then extend it to the flow-like graph.

1) Entanglement rate of a quantum channel: Given two neighboring nodes v_i and v_j, a quantum channel consists of parallel quantum links connecting them. Let w represent the width of the quantum channel. This width indicates the number of parallel quantum links positioned between the two switches to facilitate the sharing of a single quantum state. Ref. [46] provides a detailed description of entanglement generation at the link layer. As depicted in Figure 5, the width of the edge connecting Alice and Carol is 2. The entanglement rate between these two neighboring switches is defined as the probability of creating at least one successful entanglement link for a quantum state, i.e.,

$$P_{ij} = 1 - (1 - p_{ij})^w.$$

Here, $p_{ij} = \mathbf{e}^{-\alpha L_{ij}}$ represents the successful entanglement probability over a link with an Euclidean length of L_{ij}. Referring to Figure 5, if we let p be the successful entanglement probability of an individual quantum link and assume all links have this uniform rate, then the entanglement rate between Alice and Carol becomes $1 - (1 - p)^2$.

2) Entanglement rate of a path: For a specified path $\mathrm{A} = \{a_0, a_1, \ldots, a_l\}$, with l representing the path's length in terms of the number of edges and each $a_i \in \mathrm{A}$ is a $v_j \in \overline{\mathcal{V}}$, the entanglement rate for a quantum state is determined by the successful entanglement-swapping probability at each intervening switch and the entanglement rate across each edge. Formally, the entanglement rate for path A is:

$$P_A = \prod_{i=2}^{l-1} q_i \prod_{j=1}^{l} P_{j(j+1)}.$$

Referring to Figure 5, the entanglement rate of a path spanning Alice and Bob is $(1-(1-p)^2)pq$, with q representing Carol's successful probability to employ n-fusion for fusing three quantum links.

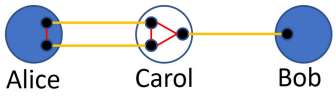


Fig. 5. Example of entanglement along a path: Black qubits are entangled via red internal links within switches through 3-fusion, and orange links between switches. The quantum channel width w is 2 between Alice and Carol, and 1 between Carol and Bob.

3) Entanglement rate of a flow-like graph: To compute the entanglement rate for a shared quantum state within this graph, it is imperative to traverse each branch node and every path encompassing these nodes.

The computational procedure initiates with a single quantum user navigating through every edge and switch sequentially. Consider a flow-like graph represented as $\hat{G} = \{U_1, U_2, \varrho\}$, where the intent is to share the quantum state ϱ between quantum user U_1 and U_2. We designate user U_1 as the root node. Starting from user U_1, neighboring nodes of user U_1 along the direction from user U_1 to user U_2 are called child nodes. Every child node can have its own child nodes.

$P_{\{a,b,\varrho\}}$ represents the entanglement rate from node a to node b for ϱ. When we fix a quantum state ϱ, the notation is shortened as $P_{\{a,b\}}$, where node a and node b are two nodes in $\hat{G} = \{U_1, U_2, \varrho\}$ and node a has a shorter distance to user U_1 compared with node b. $\mathcal{C}_a$ denotes the set of all child nodes of node a in a symmetric flow graph. Then the entanglement rate of $\hat{G} = \{U_1, U_2, \varrho\}$ can be computed via a recursive process:

$$P_{\{U_1,U_2\}} = 1 - \prod_{u \in \mathcal{C}_{U_1}} \left(1 - P_{\{U_1,u\}} P_{\{u,U_2\}}\right). \quad (1)$$

Eq. (1) calculates the overall entanglement rate between two nodes, U_1 and U_2, by combining the probabilities of successful entanglement across multiple paths through child nodes of U_1. For each child node u, the term $P_{\{U_1,u\}}P_{\{u,U_2\}}$ represents the success probability of establishing entanglement along a path from U_1 to U_2 via u, while $1 - P_{\{U_1,u\}}P_{\{u,U_2\}}$ gives the probability of failure along that path. The product $\prod_{u\in\mathcal{C}_{U_1}} \left(1 - P_{\{U_1,u\}}P_{\{u,U_2\}}\right)$ represents the combined probability that all paths fail to establish entanglement, so the final expression $1 - \prod_{u\in\mathcal{C}_{U_1}} \left(1 - P_{\{U_1,u\}}P_{\{u,U_2\}}\right)$ gives the probability that at least one path successfully creates entanglement between U_1 and U_2, effectively calculating the entanglement rate across multiple potential paths.

Each entanglement state generation is independent of others. The entanglement rate of a single state equals its successful entanglement probability. Even if undesired states arise during n-fusion, they remain local to the switch and can be removed via Z-measurements or overwritten in the next cycle. Since our focus is to maximize the total expected number of states across the entire network, such local by-products do not affect the end-to-end distribution of GHZ states.

C. The Entanglement Routing Problem

This work explores an entanglement routing problem in the quantum network, where switches can execute n-fusion entanglement-swapping using GHZ measurements. We focus on a predefined quantum network characterized by an arbitrary network graph $G = (\overline{\mathcal{V}} = \mathcal{U} \cup \mathcal{V}, \mathcal{E})$. Here, a collection of quantum-user pairs, represented as $\langle s_i, d_i \rangle, s_i, d_i \in \mathcal{U}$, aims to establish entangled quantum states (encompassing Bell states and GHZ states) amongst themselves. A quantum user holds the capability to simultaneously share distinct states with various other quantum users, and multiple quantum states can be exchanged between a single user pair.

We assume each quantum user has sufficient qubit memory for entanglement since a quantum user can be regarded as a quantum cluster with extensive qubit capacity, enhanced by integrating multiple quantum processors.

Due to hardware implementation challenges, the capacity of current quantum switches in labs usually ranges from 2 to 8 qubits [47], [48]. In contrast, optical fiber technology provides a more economical alternative at about 0.5 U.S. dollars per kilometer and allows a single cable to include up to 25 cores, each functioning as a separate entanglement link. It's also feasible to install multiple optical fiber cables between switches, ensuring sufficient edge capacity for quantum entanglement. We denote the number of qubits in a switch $v \in \mathcal{V}$ as Q_v. The number of entangled pairs that can be generated over an edge between two neighboring nodes is limited by the smaller memory capacity of those nodes, i.e., $\min\{Q_u, Q_v\}$, where Q_u and Q_v are the capacities of the two nodes.

In this study, the foremost constraint is the number of qubits in each switch, implying that a switch can't allocate more qubits for entanglement than its capacity. We assume that every switch maintains a consistent successful entanglement-swapping probability, represented by $q \in [0, 1]$ for n-fusion. The success rate of establishing a quantum link directly correlates with its Euclidean length, described by the equation $p_{ij} = \mathbf{e}^{-\alpha L_{ij}}$. Our main objective is to optimize the network's entanglement rate, which is the expected number of shared quantum states across quantum-user pairs. The specifics of the entanglement process are elaborated in Section III-C. We articulate the problem as:

Problem. ***n-fusion Entanglement Routing Problem (NERP).*** Let $\mathcal{G} = (\mathcal{U} \cup \mathcal{V}, \mathcal{E})$ be an undirected graph, where $\mathcal{U}$ is the set of quantum-user nodes, $\mathcal{V}$ is the set of quantum-switch (repeater) nodes, and $\mathcal{E} \subseteq (\mathcal{U} \cup \mathcal{V}) \times (\mathcal{U} \cup \mathcal{V})$ is the set of fiber links.

Input. 1. User pairs $\mathcal{M} = \{(s_i, d_i) \mid s_i, d_i \in \mathcal{U},\ i = 1, \ldots, M\}$. 2. Topology of the graph $\mathcal{G}$. 3. Switch capacities for each $v \in \mathcal{V}$, an integer $Q_v \in \mathbb{N}_+$ giving the maximum number of simultaneous entanglement links incident to v.

Output. A collection of flow-like graphs (paths) $\mathcal{R} = \{R_i\}_{i=1}^{M}$ such that each graph connects user pair (s_i, d_i) via quantum links; for every switch $v \in \mathcal{V}$, the total number of links selected on edges incident to v does not exceed Q_v.

Objectives. Choose path sets (and the corresponding fusion operations performed on the selected links) so as to *maximize the overall end-to-end entanglement rate delivered to all user pairs, subject to the above capacity constraints.*

VI. Problem Analysis

A. Impact of n-fusion for the Entanglement Routing

This research sets itself apart from prior work by introducing a broader model, accounting for switches that execute n-fusion in a quantum network with arbitrary topology for entanglement routing. This approach results in flow-like graphs forming between pairs of quantum users. Compared to traditional entanglement swapping that employs BSMs, n-fusion offers several advantages:

1) Increased flexibility in fusing quantum states: BSMs can only fuse two links simultaneously; thus, when the number of qubits in a switch is odd, one qubit will remain unused. In contrast, n-fusion enables the switch to fuse varying numbers of quantum links for different quantum user pairs, thus fully utilizing the qubits within the switch. For example, consider a switch with q qubits, n-fusion permits various combinations that can fully exploit all available qubits within the switch. In comparison, swapping under BSMs can only connect $\lfloor q/2 \rfloor$ pairs of quantum links. Additionally, since n-fusion is a general case of swapping under BSMs, a switch capable of performing n-fusion can also perform BSMs for swapping. Therefore, n-fusion can utilize the qubits within the switches more efficiently and thus enhance flexibility in fusing qubits for entanglement generation.
2) Enhanced adaptability to varying network demands: By enabling switches to fuse an arbitrary number of links, n-fusion allows the network to tailor entanglement resources to user requirements. For example, when users require richer correlations, n-fusion can generate k-qubit GHZ states between a node pair, which cannot be deterministically achieved using only BSMs without local ancilla qubits and additional operations. It enables two-node protocols such as QKD [18], quantum telephone exchanges, and distributed quantum computing [20].

Remark on Entanglement Type. This work focuses on entanglement distribution between *user pairs*, where the shared entanglement could be Bell pairs or multi-qubit states, depending on the network configuration and fusion operation. Notably, traditional BSM-based entanglement swapping is a special case of n-fusion when $n = 2$.

B. Challenges

To successfully tackle the challenges of entanglement routing, the algorithm must primarily address two core issues:

1) **Routing:** This involves identifying the routes for connecting quantum-user pairs to share quantum states.
2) **Allocation:** Once routes are set, the algorithm must efficiently allocate qubits within switches for the selected paths.

Addressing these issues presents a two-pronged challenge.

(1) The main challenge in designing flow-like routes under the n-fusion framework (Section V-A) arises from two core differences from classical models: (i). Non-linear, probabilistic objective. The entanglement rate depends non-linearly on the fused links and their success probabilities, unlike the linear metrics optimized by algorithms such as Dijkstra [49], Yen [50], or Ford–Fulkerson [45]. (ii). No flow conservation. Classical flows enforce conservation at intermediate nodes, whereas our model limits the total number of connected links by each node's quantum memory capacity. This violates standard flow assumptions, making conventional multi-commodity flow algorithms [51] inapplicable.

Moreover, the number of potential flow-like graphs for a single quantum-user pair in a comprehensive graph can be as high as $2^{(|\mathcal{V}|-2)!\mathbf{e}}$, where $|\mathcal{V}|$ is the vertex count in $\mathcal{G}$. The number of simple paths in a complete graph is given by $(|\mathcal{V}|-2)!\mathbf{e}$ [52], and flow-like graphs can be a combination of any simple paths. This complexity results in significant computational overhead when identifying efficient routes under n-fusion.

(2) The secondary challenge involves determining the optimal qubit allocation in switches for routes corresponding to diverse quantum states to maximize the entanglement rate, a task complicated by two key factors.

Firstly, calculating the entanglement rate for specific routes is computationally intensive due to the non-linear and recursive nature of the formula depicted in Equation 1. This calculation requires recursive traversal through all branches and paths of a flow-like graph, significantly increasing computational demands.

Secondly, the design process for qubit allocation within a flow-like graph is complicated by the variability in qubit

numbers allocated by switches to a shared quantum state. Unlike traditional swapping where qubit allocation remains consistent, in this study, the allocation can vary across different switches. Therefore, the proposed algorithm must not only efficiently identify routes from numerous possibilities but also skillfully manage qubit assignments within nodes for complex paths linked to specific pairs.

It is important to note that the **Routing** problem and the **Allocation** problem are intricately linked when addressing the NERP. Specifically, determining the branches at a branch switch necessitates the determination of the qubit allocation for that switch. Consequently, both the **Routing** and **Allocation** problems must be addressed jointly and simultaneously.

Theorem 1. *The NERP is NP-Complete.*

The detailed proof is provided in the appendix.

This paper presents a fundamental routing model under n-fusion, abstracting the quantum network as a graph with fusion-specific constraints that depart from classical routing formulations. While n-fusion may introduce higher fidelity loss or generate unintended states compared to traditional swapping, our focus is on the algorithmic structure of the routing problem. This abstraction enables further scalable analysis (e.g., Ref. [39]) and provides a foundation for future fidelity-aware extensions.

VII. Entanglement Routing Algorithms

A. Algorithm Overview

Notably, a flow-like graph can be conceptualized as a union of multiple paths representing the same quantum state. Given the complexity of directly identifying flow-like graphs and the high time complexity associated with calculating their entanglement rate, we first identify the paths and then merge them to form flow-like graphs to maximize the entanglement rate. We propose an entanglement routing algorithm that unfolds in three distinct steps, with each comprising one or two sub-algorithms. An example of all algorithms is given in Figure 6.

- *Step I*: We develop algorithms to construct a set of paths for unique shared quantum states, prioritizing paths with the highest entanglement rates across different widths. The paths from this ensemble will either be employed directly as routes or be amalgamated to form flow-like graphs for shared quantum states.
- *Step II*: This step is to select paths and then merge them as entanglement routes to maximize the network's overall entanglement rate. We consider two distinct merging policies based on selection orders: the width of a path and the entanglement rate of a path. Starting with paths that either have the largest width or the largest entanglement rate, we sequentially pick paths. Our aim here is to merge chosen paths corresponding to the same quantum state under n-fusion, a process that helps conserve the qubit resources within the network and consequently facilitates the formation of flow-like graphs.
- *Step III*: Any remaining unallocated qubits are harnessed to establish quantum links in this step. These links are appended to paths or flow-like graphs established in *Step II* with the intention of increasing the entanglement rate. For each shared quantum state, the switches aim to merge the maximum possible quantum links, adhering to the routes specified by the entanglement routing algorithm.

B. Step I: Construct a path set

In Step I, our objective is to identify the path with the largest entanglement rate given a specified width. Given that the paths in this set are instrumental in determining the final entanglement routes, network resources can be utilized repeatedly during the selection process. Specifically, Algorithm 1 is initially proposed to determine a w-width path with the largest entanglement rate. Building upon the foundation of Algorithm 1, we then introduce Algorithm 2 to simultaneously discern multiple paths across a range of widths. The output of Algorithm 2 will serve as the path set for the merging in *Step II*.

Algorithm 1 Largest Entanglement Rate Algorithm

Input: $\overline{\mathcal{V}}, \mathcal{E}, S, D, \mathcal{Q}, w$
Output: A $= \{S, a_1, ..., a_{l-1}, D\}, b_D$

$Queue \leftarrow \{S\}$, $b_i \leftarrow 0, pre_i \leftarrow 0 (\forall i \in \overline{\mathcal{V}})$, $b_S \leftarrow 1$
if $Q_S < w$ or $Q_D < w$ **then**
  Return no solution
end if
while $Queue \neq \emptyset$ **do**
  Select $v_i \in Queue$ s.t. b_{v_i} is max
  Remove v_i from $Queue$
  for all $v_j \in \overline{\mathcal{V}}, e_{v_i,v_j} \in \mathcal{E}$ **do**
    if $b_{v_j} < b_{v_i}$ through e_{v_i,v_j} and $Q_{v_j} \geq 2w$ **then**
      $b_{v_j} \leftarrow b_{v_i}$ through e_{v_i,v_j}, $pre_{v_j} \leftarrow v_i$
      if $v_j \notin Queue$ **then**
        $Queue \leftarrow Queue \cup v_j$
      end if
    end if
  end for
end while
if $b_D = 0$ **then**
  Return no solution
end if
$t \leftarrow pre_D$
while $t \neq S$ **do**
  $a_l \leftarrow v_t, t \leftarrow pre_t, l \leftarrow l - 1$
end while

1) Algorithm 1: The w-width Path Selection: The algorithm 1 finds a path with the highest entanglement rate on a graph. The approach is reminiscent of Dijkstra's algorithm [49], but the objective is to maximize the entanglement rate rather than minimize path length. Given the graph $\mathcal{G} = (\overline{\mathcal{V}} = \mathcal{U} \cup \mathcal{V}, \mathcal{E})$, Algorithm 1 selects a path A between S, D with a specified width w. We assume each edge provides link capacity exceeding the adjacent nodes' memory capacities, ensuring memory remains the only bottleneck for entanglement generation. S and D are quantum users or quantum switches aiming to share a state. $Q_{v_i} \in \mathcal{Q}$ is the number of qubits in a node $v_i \in \overline{\mathcal{V}}$. If the node is a quantum user, we assume its capacity Q_{v_i} is infinity. A represents the path from S to D, and b_D denotes the entanglement rate of the path from S to D. The process of the algorithm is as follows:

- Line 1: Create a priority queue $Queue$ to store possible nodes boasting the largest entanglement rate.
- Line 2 to 4: Ensure S and D have enough qubits for the path with width w.
- Line 5 to 7: Select a node v_i from $Queue$ with the maximum b_{v_i}, while b_{v_i} indicates the entanglement rate of the path from S to v_i. This procedure iterates until there is no element in $Queue$.
- Line 8 to 15: Enumerate the neighbor nodes v_j for all edge $e_{v_i v_j} \in \mathcal{E}$ to update b_{v_j}. If b_{v_j} is updated, it can be a potential selected node of subsequent iterations.
- Line 17 to 19: Check if an entanglement path exists from S reaches D.
- Line 20 to 23: Construct the path A by tracing the array pre_t, while pre_t records the previous node that belongs to the maximum entanglement path ends with node t. pre_t is recorded when updating b_t in line 10.

The algorithm's validity is underpinned by the non-increasing nature of the metric. In each step, p_{ij}, which reflects the success rate of a hop as $\mathbf{e}^{-\alpha L_{ij}}$ within the [0,1] range, is added to an existing path. Since the success rate of the existing path also falls within [0,1], the combined success rate remains within this range and does not exceed the original path's rate. Consequently, once a node is identified as having the highest entanglement rate path, its rate will not be updated further.

Algorithm 2 Paths Selection Algorithm

Input: $\overline{\mathcal{V}}, \mathcal{E}, \mathcal{S}, \mathcal{D}, \mathcal{M}, \mathcal{Q}, h$
Output: $\mathcal{A}$
1: **for all** $\varrho_{SD} \in \mathcal{M}$ **do**
2: **for all** w from W to 1 **do**
3: $Alg1(\overline{\mathcal{V}}, \mathcal{E}, S, D, \varrho_{SD}, \mathcal{Q}, w)$
4: Obtain path A $= \{S, a_1, a_2, ..., D\}$ and metric b_D
5: $PQueue \leftarrow \{(\emptyset, \text{A}, b_D)\}, Ext \leftarrow 0$
6: **while** $PQueue \neq \emptyset$ and $Ext < h$ **do**
7: Extract $(E, \text{A} = \{a_0 = S, a_1, a_2, ..., a_l = D\}, b_D) \in PQueue$ with maximum b_D
8: $\mathcal{A} \leftarrow \mathcal{A} \cup (\text{A}, w), Ext \leftarrow Ext + 1$
9: Remove (E, A, b_D) from $PQueue$
10: **for all** $e \leftarrow (a_i, a_{i+1}) \in$ A **do**
11: $Alg1(\overline{\mathcal{V}}, \mathcal{E} \setminus (E \cup \{e\}), a_i, D, \varrho_{SD}, \mathcal{Q}, w)$
12: Obtain sub-path $\{a_i, a'_1, a'_2, ..., D\}$ and metric b'_D
13: Construct A$' \leftarrow \{S, a_1, ..., a_i, a'_1, a'_2, ..., D\}$
14: Compute metric b'_D
15: $PQueue \leftarrow PQueue \cup (E \cup \{e\}, \text{A}', b'_D)$
16: **while** $|PQueue| + |\mathcal{A}| > h$ **do**
17: Remove item from $PQueue$ with min b_D
18: **end while**
19: **end for**
20: **end while**
21: **end for**
22: **end for**

2) Algorithm 2: Multi-path Selection: Algorithm 2 is designed to compute paths with the peak entanglement rates amongst all quantum-user pairs in $\mathcal{M} = \{\varrho_{SD}, S \in \mathcal{S}, D \in \mathcal{D}\}$. Here, Yen's algorithm is adapted to utilize Algorithm 1. The maximum possible width W $= \max\{Q_{v_i}, \forall v_i \in \mathcal{V}\}$, satisfies the largest qubit count in a single switch. The algorithm outputs a set $\mathcal{A}$ that records the information of selected paths. Each element in $\mathcal{A}$ is a tuple (A, w), while A is the path and w is the width of the path. The process of the algorithm is as follows:

- Line 1 to 2: Enumerate the quantum-user pair and width. For a designated quantum state ϱ_{SD} and width w, the algorithm identifies h paths with the highest entanglement rates and includes them, along with their corresponding width w, in the set $\mathcal{A}$. h denotes the predefined number of paths identified using Algorithm 2. Generally, a larger h increases the chance of finding higher-entanglement-rate paths but also incurs greater computational cost. h is set to 10^5 by default.
- Line 3 to 5: Identify a path A with the largest entanglement rate as the initial path in a priority queue $PQueue$. Each entry in $PQueue$ consists of three components: an excluded edge set E, the path A, and the entanglement rate b_D.
- Line 6 to 9: Select the path A from $PQueue$ with the peak entanglement rate, not already in $\mathcal{A}$, which is considered the next largest entanglement rate path. Ext limits the number of loops to no more than h.
- Line 11 to 15: Construct potential paths by A and store them in $PQueue$. For path A $= \{a_0 = S, a_1, ..., a_l = D\}$ which includes the hop $e = (a_i, a_{i+1})$, the algorithm calculates a sub-path with the largest entanglement rate from a_i to D. Notably, A should exclude edge e; therefore, e needs to be added to the excluded edge set. In subsequent iterations, any edge in the excluded edge set will not be considered for inclusion in A. Line 11 to line 13 computes and constructs the new potential path A$' = \{S, a_1, a_2, ..., a_i, a'_1, a'_2, ..., D\}$ by merging the sub-path $\{a_i, a'_1, a'_2, ..., D\}$ with original $\{S, a_1, a_2, ..., a_i\}$ from A.
- Line 16 to 18: Remove items from $PQueue$ when the size of $PQueue$ plus the size of $\mathcal{A}$ exceeds h.

C. Step II: Merge paths as flow-like graphs

With a set of feasible paths from Step I, we propose Algorithm 3 to select the set of routes denoted as $\mathcal{A}$ to maximize the network entanglement rate. We denote $\mathcal{R} = \{r_{\varrho_{SD},u,v} | \varrho_{SD} \in \mathcal{M}, u, v \in \overline{\mathcal{V}}\}$ to record the qubit assignment, where $r_{\varrho_{SD},u,v}$ is the number of qubits between quantum users or switches u and v assigned to entangle the quantum-user pair ϱ_{SD}. The paths with the same shared quantum states between a quantum-user pair could be merged as a flow-like graph to save qubits in the network. The details are summarized in Algorithm 3.

- Line 2 to 3: Sort and enumerate paths in $\mathcal{A}$ by the merging policy.
- Line 4 to 16: Assess each quantum switch a_i in the enumerated path A with length l, and calculate the needed qubits for merging adjacent edges. There are two scenarios for merging an edge: First, if the edge already exists in previous paths for the same quantum-user pair, it merges directly. Second, if the edge isn't part of previous paths but both endpoints have enough remaining qubits, it can be merged by consuming tmp qubits (Line 7 to 12). If there are insufficient qubits, the edge cannot merge, making the path unavailable (Line 13 to 15).

- Line 18 to 25: Update remaining qubits $\mathcal{Q}$ and qubit assignment $\mathcal{R}$. Once all switches have adequate qubits, the variable *good* is set to 1, allowing the path A to be merged by assigning required qubits from $\mathcal{Q}$. If an edge is already included in a previous path, its qubits are reused; otherwise, qubits are deducted from both endpoints.
- Line 27: Remove (A, w) from $\mathcal{A}$ if A cannot be merged.

Algorithm 3 Paths Merging Algorithm

Input: $\overline{\mathcal{V}}, \mathcal{E}, \mathcal{S}, \mathcal{D}, \mathcal{M}, \mathcal{Q}, \mathcal{A}$
Output: $\mathcal{A}, \mathcal{R}$
1: $r_{\varrho_{SD},u,v} \leftarrow 0, \forall \varrho_{SD} \in \mathcal{M}, u, v \in \overline{\mathcal{V}}$
2: Sort $\mathcal{A}$ by the merging policy
3: **for all** $(\mathrm{A}, w) \in \mathcal{A}$ **do**
4: $\quad good \leftarrow 1$
5: $\quad$ **for all** $i \in [1, l)$ **do**
6: $\quad\quad tmp \leftarrow 0$
7: $\quad\quad$ **if** $r_{\varrho_{a_0 a_l}, a_{i-1}, a_i} < w$ **then**
8: $\quad\quad\quad tmp \leftarrow tmp + w - r_{\varrho_{a_0 a_l}, a_{i-1}, a_i}$
9: $\quad\quad$ **end if**
10: $\quad\quad$ **if** $r_{\varrho_{a_0 a_l}, a_i, a_{i+1}} < w$ **then**
11: $\quad\quad\quad tmp \leftarrow tmp + w - r_{\varrho_{a_0 a_l}, a_i, a_{i+1}}$
12: $\quad\quad$ **end if**
13: $\quad\quad$ **if** $Q_{a_i} < tmp$ **then**
14: $\quad\quad\quad good \leftarrow 0$, break
15: $\quad\quad$ **end if**
16: $\quad$ **end for**
17: $\quad$ **if** $good = 1$ **then**
18: $\quad\quad$ **for all** $i \in [0, l)$ **do**
19: $\quad\quad\quad$ **if** $r_{\varrho_{a_0 a_l}, a_i, a_{i+1}} < w$ **then**
20: $\quad\quad\quad\quad Q_{a_i} \leftarrow Q_{a_i} - (w - r_{\varrho_{a_0 a_l}, a_i, a_{i+1}})$
21: $\quad\quad\quad\quad Q_{a_{i+1}} \leftarrow Q_{a_{i+1}} - (w - r_{\varrho_{a_0 a_l}, a_i, a_{i+1}})$
22: $\quad\quad\quad\quad r_{\varrho_{a_0 a_l}, a_i, a_{i+1}} \leftarrow w$
23: $\quad\quad\quad\quad r_{\varrho_{a_0 a_l}, a_{i+1}, a_i} \leftarrow w$
24: $\quad\quad\quad$ **end if**
25: $\quad\quad$ **end for**
26: $\quad$ **else**
27: $\quad\quad \mathcal{A} \leftarrow \mathcal{A} \backslash (\mathrm{A}, w)$
28: $\quad$ **end if**
29: **end for**

Given the paths from *Step I*, we propose two distinct merging policies to match line 2 of Algorithm 3: width-preferred and entanglement rate-preferred.

- Merging Policy I, width-preferred: sort $(\mathrm{A}, w) \in \mathcal{A}$ by w from high to low. For items with the same w, sort them by b_A from high to low.
- Merging Policy II, entanglement rate-preferred: sort $(\mathrm{A}, w) \in \mathcal{A}$ by b_A from high to low.

In both merging policies, if multiple items have the same entanglement rate, we break ties using a predetermined order to ensure consistent and reproducible selection. The order is determined by the sequence in which the paths are selected by Algorithm 2. According to the simulation results, the result of applying the entanglement rate-preferred result is worse than the result of applying the width-preferred policy.

D. Step III: Utilize remaining qubits

After running Algorithm 3, the majority of the qubits in the network are allocated to construct entanglement routes. Nonetheless, a handful of unassigned qubits may still exist within the network. These residual qubits can be utilized

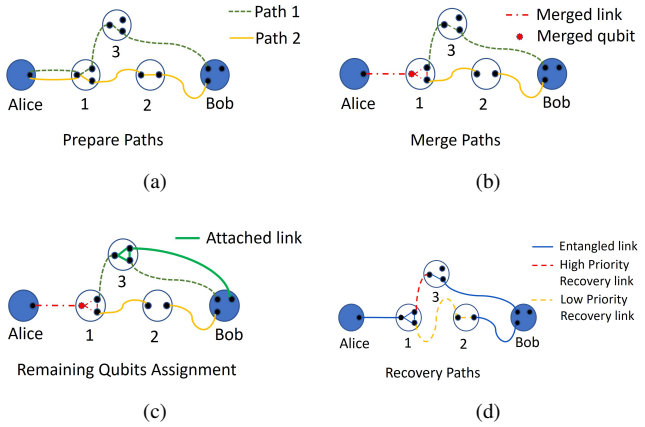


Fig. 6. An example of the entanglement process using the proposed algorithms is shown. (a) *Step I*: Prepare two independent paths to entangle a quantum state between a quantum user pair. (b) *Step II*: Merge the two single paths into a flow-like graph by removing commonly utilized qubits between Alice and Node 1. (c) *Step III*: Enhance the entanglement rate by adding a quantum link between Node 3 and Bob to the flow-like graph. (d) Building recovery paths: Nodes attempt to build recovery paths to improve the entanglement rate. As shown in the figure, Node 1 and Node 3, Node 1 and Node 2 are not entangled, so Node 1 attempts to build recovery paths to entangle with Node 1 and 2. The entanglement between nodes 1 and 3 has a higher priority because the qubits in Node 2 are not entangled resulting in a lower entanglement rate.

to enhance the overall network entanglement rate. We will incorporate these remaining qubits into the routes selected by Algorithm 3, aiming to augment the entanglement rate by expanding the width of quantum channels. The collection of these newly incorporated links is represented as $\mathcal{A}'$. However, searching for a path in the manner of Algorithm 3 proves inefficient at this stage, as paths with high entanglement rates have already been selected.

As a result, we propose Algorithm 4 to allocate remaining quibits edge by edge. The process of the Algorithm 4 is as follows:

- Line 1 to 2: Enumerate all edges with remaining qubits.
- Line 3: Prioritize the link to the route that offers the most significant enhancement in the entanglement rate relative to its previous value.
- Line 4 to 5: Assign the qubits to the quantum-user pair ϱ_{SD}. While determining the increased rate may be computationally intensive, the total execution time remains reasonable, especially considering the limited number of residual qubits in the graph post-Algorithm 3.

Algorithm 4 Remaining Qubits Assignment Algorithm

Input: $\overline{\mathcal{V}}, \mathcal{E}, \mathcal{S}, \mathcal{D}, \mathcal{M}, \mathcal{Q}, \mathcal{R}$
Output: $\mathcal{R}$
1: **for all** $e_{v_i, v_j} \in \mathcal{E}$ **do**
2: $\quad$ **while** $Q_{v_i} > 0$ and $Q_{v_j} > 0$ **do**
3: $\quad\quad$ Find ϱ_{SD} in $\mathcal{M}$ that maximize the increment of expected entanglement probability if $r_{\varrho_{SD}, v_i, v_j}$ increased by 1
4: $\quad\quad r_{\varrho_{SD}, v_i, v_j} \leftarrow r_{\varrho_{SD}, v_i, v_j} + 1, r_{\varrho_{SD}, v_j, v_i} \leftarrow r_{\varrho_{SD}, v_j, v_i} + 1$
5: $\quad\quad Q_{v_i} \leftarrow Q_{v_i} - 1, Q_{v_j} \leftarrow Q_{v_j} - 1$
6: $\quad$ **end while**
7: **end for**

We summarize the entanglement routing process as follows. First, Algorithm 2 is used to identify multiple simple paths with the highest entanglement rates. It repeatedly calls Algo-

rithm 1, which finds the best path with the highest entanglement rate for a given graph and source-destination pair. To conclude the entanglement routing strategy, we integrate the routes derived from Algorithm 3 with the subsequent qubit allocations determined by Algorithm 4. All established routes utilize the n-fusion entanglement approach. This means that when a specific route is allocated, switches will maximize internal qubit entanglement to fuse links for a single shared quantum state between a quantum-user pair. A detailed time complexity analysis of the algorithms is provided in the appendix.

VIII. Recovery Path Design

In this section, we introduce an online algorithm designed to quickly recover failed entanglement links from Phase II. Due to the probabilistic nature of entanglement, switches can repurpose qubits from failed attempts to engage with nearby switches. However, the limited duration of entanglement prevents communication over long distances, restricting switches to only local entanglement information from immediate neighbors. Consequently, there is a need for an efficient algorithm that can establish recovery paths quickly within these constraints. To meet this need, we have developed an online recovery path algorithm.

The algorithm is based on the perspective of the switch v_x and includes two processes: the information exchanging process and the entanglement establishment process. The quantum switch collects information in H-hop distance during the information exchange process. Based on the information collected, the quantum switches build recovery paths in the entanglement establishment process.

To recover the failed entanglement links, the proposed online algorithm (Algorithm 5) computes link-level entanglements specifically for constructing recovery paths. These link-level entanglements are then integrated into existing flow-like graphs to facilitate end-to-end entanglement.

Algorithm 5 Recovery Path Algorithm

Input: $\mathcal{G} = (\mathcal{V}, \mathcal{E}), \mathcal{S}, \mathcal{D}, \mathcal{M}, \mathcal{Q}, \mathcal{R}, v_x, t, T$
Output: $\mathcal{R}$
1: $\mathcal{F} \leftarrow \emptyset, \mathcal{W} \leftarrow \emptyset$
2: **for all** $v \in \mathcal{V}, v$ in H-hop from v_x **do**
3: Apply known entanglement result of v
4: **end for**
5: **for all** $e = (v_i, v_j) \in \mathcal{E}, v_i, v_j$ in H-hop from v_x **do**
6: Apply known entanglement result of e
7: **end for**
8: **for all** $v \in \mathcal{V}, v$ in H-hop from v_x **do**
9: **for all** $m = \varrho_{SD} \in \mathcal{M}$ **do**
10: Compute the increment c of the successful entanglement probability when entangle (v_x, v) for pair r
11: $\mathcal{F} = \mathcal{F} \cup (c, v, m)$
12: **end for**
13: **end for**
14: Sort $\mathcal{F}$ in the decreasing order of c
15: **for all** $i \in [1, min(|\mathcal{F}|, \lfloor T/t \rfloor)]$ **do**
16: $f_i = (c_i, v_i, m_i)$
17: **for all** $(R1, v_j, m_j, q_j) \in \mathcal{W}$ **do**
18: **if** $v_i = v_j$ and $m_i = m_j$ **then**
19: Reply $(R2, v_x, m_i, min(Q_{v_x}, q_j))$ to v_j
20: $Q_{v_x} \leftarrow Q_{v_x} - min(Q_{v_x}, q_j)$
21: $Qsave_{v_j, m_j} \leftarrow min(Q_{v_x}, q_j)$
22: **end if**
23: **end for**
24: **if** No request indicates entangle with v_i for pair r_i in $\mathcal{W}$ **then**
25: Send entanglement request $(R1, v_x, m_i, Q_{v_x})$ to v_i
26: **end if**
27: **while** Current time in $[(i-1)t, it)$ **do**
28: **if** Receive $(R1, v, m, q)$ **then**
29: **for all** $k \in [1, i]$ **do**
30: $f_k = (c_k, v_k, m_k)$
31: **if** $v_k = v$ and $m_k = m$ **then**
32: Reply $(R2, v_x, m, min(Q_{v_x}, q))$ to v
33: $Q_{v_x} \leftarrow Q_{v_x} - min(Q_{v_x}, q)$
34: $Qsave_{v,m} \leftarrow min(Q_{v_x}, q)$
35: **end if**
36: **end for**
37: **if** $v_k \neq v$ or $m_k \neq m, \forall k \in [1, i]$ **then**
38: $\mathcal{W} \leftarrow \mathcal{W} \cup (R1, v, m, q)$
39: **end if**
40: **else if** Receive $(R2, v, m, q)$ **then**
41: Reply $(R3, m, min(Q_{v_x}, q))$ to v
42: $Q_{v_x} \leftarrow Q_{v_x} - min(Q_{v_x}, q)$
43: $r_{m,v_x,v} \leftarrow r_{m,v_x,v} + min(Q_{v_x}, q)$
44: **else if** Receive $(R3, v, m, q)$ **then**
45: $Q_{v_x} \leftarrow Q_{v_x} + Qsave_{v,m} - q$
46: $r_{m,v_x,v} \leftarrow r_{m,v_x,v} + q$
47: **end if**
48: **end while**
49: $i \leftarrow i + 1$
50: **end for**

A. Information Exchanging Process

This process involves collecting information within an H-hop distance from v_x to prepare for the initialization of recovering failed entanglement links. The detailed process goes from line 1 to line 14 in Algorithm 5.

- Line 2 to 7: Collect entanglement information from all switches (lines 2 to 4) and edges (lines 5 to 7) within an H-hop distance, where H is the maximum communication distance during the entanglement period.
- Line 8 to 9: Enumerate all potential recovery paths.
- Line 10 to 11: Evaluate potential increases in entanglement by pairing with each neighbor, and construct tuples $f = (c, v, m)$, where c, v, and m represent the increment, adjacent switch, and associated pair, respectively.
- Line 14: Sort tuples in descending order by c. Assuming ordered tuples are $f_1, f_2, \ldots, f_{|\mathcal{F}|}$, where $|\mathcal{F}|$ represents the total number of tuples.

B. Entanglement Establishment Process

This process is the main process that interacts with other switches to construct the recovery paths. We assume the upper bound of the runtime for the entanglement establishment process to be T, which equals the average coherence lifetime of an entangled pair in memory, ensuring it is less than the total duration to allow time for link recovery. Based on T, a time slot t is chosen, which serves as the waiting period for switches to exchange information and achieve consensus.

Ideally, switch v_x aims to entangle as many qubits as

possible with the switch for the pair that yields the largest increment, as described by tuple $f_1 = (c_1, v_1, m_1)$. However, from the viewpoint of the neighboring switch v_1, it might decline the entanglement proposition due to potentially more favorable alternatives. The decision-making protocol requires three rounds to finish the consensus of the entanglement.

1) First round: is to let a switch send a first-round request to another switch to initiate building a recovery path. A first-round request is indicated as $(R1, v, m, q)$, where v is the sender, m is the entanglement pair, and q is the maximum remaining qubits that can be used by v.

- Line 17 to 18: Check if the first-round request that matches v_i and m_i has been received before. If yes, v_x will enter the second round.
- Line 24 to 25: Send a first-round request to express the intention that entangles with v_i for pair m_i if no first-round request in $\mathcal{W}$ matches the pair.

2) Second round: is to confirm the path purposed by the first-round request. When a switch receives a first-round request and wants to build the purposed path, the switch sends the second-round request for confirmation to the other switch of the purposed path. A second-round request has the format $(R2, v, m, q)$. Here, v is the sender, m is the entanglement pair, and q is the maximum number of qubits available, determined by comparing qubits from the initial request and v's remaining qubits.

- Line 19: Reply a second-round request prompted by the first-round request remembered by switch v_x.
- Line 20 to 21: Update the number of available qubits of v_x, and reserve qubits for the entanglement after dispatching the second-round request.
- Line 27: Repeatedly check newly received requests during the i-th time slot.
- Line 28 to 31: Check if the received first-round request can be triggered.
- Line 32: Reply a second-round request for the received first-round request.
- Line 33 to 34: Update the number of available qubits of v_x, and reserve qubits for the entanglement after dispatching the second-round request.
- Line 37 to 38: Save the first-round request to $\mathcal{W}$ for later use when receiving a request that does not want to proceed in the current time slot.

3) Third round: is to request signals for the start of the recovery path construction. Upon receiving a second-round request, a switch sends a third-round request and begins building the recovery path. The receiving switch, upon getting this request, also starts constructing the path, knowing that its peer has already begun. A third-round request has the format $(R3, v, m, q)$, while v is the sender of the third-round request, m is the entanglement pair, and q is the maximum number of qubits that can be used.

- Line 40 to 41: Reply a third-round request to the sender of the second-round request.
- Line 42 to 43: Immediately start to entangle and assign qubits. Note that, q in the third round cannot be greater than the q in the corresponding second round, which is trivial.
- Line 44 to 47: Start to entangle when receiving a third-round request. The receiver has reserved qubits during the second round. It not only assigns qubits for entanglement but also releases the extra qubits after receiving the third-round request.

We summarize Algorithm 5, which runs on each quantum switch. The switch first collects known entanglement results from its neighbor switches. Then it ranks all potential solutions in decreasing order of successful entanglement probability. Next, the switch initiates a three-round interaction protocol with its neighbor switches. The core idea of the protocol is to gradually expand the range of candidate solutions over time while waiting to reach consensus with the neighbor switches.

IX. Simulation Results

In this section, we present the results of our simulations. We have implemented the proposed algorithms and compared their performance to existing methods. We have conducted extensive evaluations by varying multiple parameters to increase the reliability of the simulations. We follow the simulation setup described in Ref. [15], but independently implemented our simulation in C++. All simulations were conducted on a desktop equipped with an Intel Core i7 processor and 32 GB RAM.

A. Network Setting

The default network settings are as follows, with different network parameters being tested separately later. We generate the network using the Waxman method [53]. The quantum network's area is set to a 100×100 unit square, where each unit is considered as 1 kilometer. Switches and quantum users are nodes randomly placed within the area. The network comprises 100 switches. There are 20 unique quantum-user pairs, and each pair shares a unique quantum state. We fix a time window matching the typical entanglement lifetime and measure performance by the expected number of entangled states shared within this period. As the window is fixed, we report the count rather than a time-normalized rate. Edge generation follows the work in [53]. Quantum-user nodes are connected directly to switches but not to other quantum-user nodes. The distance of each edge is at least $\frac{0.5}{\sqrt{|\overline{\mathcal{V}}|}}$, where $|\overline{\mathcal{V}}|$ represents the number of nodes in the network. The number of edges is determined by the average degree of the switches, set at 10. For reference, the average minimum path hop number between all users is approximately 4.2, but due to random placement, there are significant differences in paths. We assume that each edge has sufficient capacity to serve quantum users according to our model's assumptions. The classical bit transmission speed is set as $2.07 * 10^8 m/s$ [46].

The main limitation of the network is the switch capacity, capped at 10 qubits as determined by real experiments [54]. To account for network topology randomness, we generate and analyze five random networks, averaging their entanglement rates. The default success probability for entanglement swapping in switches is set at 0.9 [15], and the average

TABLE I
NETWORK ENTANGLEMENT RATE UNDER DIFFERENT NETWORK GENERATION METHODS

Graph Method	Alg-N-Fusion	NF-Rec	Q-CAST	B1	Alg-3	Merge-II
[Waxman]	15.60266	15.70904	6.43951	5.64729	14.11699	13.980894
[Watts-S]	14.54874	15.80869	6.88530	6.03613	12.64844	12.95601
[Aiello]	14.28436	16.00000	5.69976	5.06450	12.33531	11.30182

successful entanglement probability across links is modeled as $P_{ij} = e^{-\alpha L_{ij}}$ with α at 0.01. When running the recovery path algorithm, the maximum hops H is set to 5. This ensures that the total swapping and communication delay remains well within the coherence time (1.46 s [55]) for quantum memory. The default simulation parameters are summarized in Table III in the appendix.

B. Algorithm Benchmarks

We compare the network performance with the following algorithms.

- ALG-N-FUSION: We name our proposed entanglement routing algorithm as ALG-N-FUSION, which employs Merging Policy I.
- NF-REC: NF-REC represents the combined result of ALG-N-FUSION and Algorithm 5.
- ALG-3: This approach involves performing only Algorithm 3, which employs Merging Policy I and does not utilize the remaining qubits through Algorithm 4.
- MERGE-II: The results obtained using Merging Policy II are denoted as MERGE-II.
- Q-CAST: This algorithm is a specific version of ALG-N-FUSION where $N = 2$. This indicates that switches only perform traditional swapping through BSMs. This version closely resembles the Q-Cast algorithm proposed in [15], [16], so we name it Q-CAST.
- BASELINE-1(B1): This algorithm extends the one in [18] from a single pair to multiple pairs. For each pair, we run the algorithm once and remove the occupied resources. The result represents the total entanglement rate of all pairs.

C. Results

1) Network generation methods: We utilize two additional methods to generate networks. The first method is the Watts-Strogatz approach [56], known for producing networks that bear properties akin to real-world communication systems. The second method is based on Aiello algorithm [57], which creates scale-free power-law random graphs mirroring the topologies commonly observed in real-world networks.

Table I presents the results of comparing network performance under three network generation methods. As shown, NF-REC achieves the highest network entanglement rate when compared to other benchmarks in networks generated by different methods. This suggests that our proposed algorithm can adapt to general network topologies and achieve good performance.

2) Algorithm 4 (Remaining Qubits Assignment Algorithm) Performance: Table I further illustrates the enhancements delivered by Algorithm 4. This is evident when comparing the performance of ALG-N-FUSION with the outcome when solely employing Algorithm 3, without the incorporation of Algorithm 4. Notably, incorporating Algorithm 4 can boost the network entanglement rate by as much as 16.3%. Such improvement is attributed to the fact that some qubits might be overlooked for path selection in Step I due to width constraints. Leveraging n-fusion allows for the more efficient utilization of these qubits in Step III, courtesy of Algorithm 4, which in turn bolsters network performance. This underscores the significance of Algorithm 4 as an integral facet of our entanglement routing algorithm.

3) Algorithm 5 (Recovery Algorithm) Performance: Algorithm 5 improves the entanglement rate in most cases. As shown in Table I, Figure 7a, Figure 7b, Figure 7c, and Figure 7d, there's a notable improvement when comparing ALG-N-FUSION to NF-REC, reaching up to 16.6%. This underscores our recovery path algorithm's ability to effectively salvage numerous paths that weren't initially entangled. Notably, Figure 8a indicates that when the average link success probability stands at 0.1, Algorithm 5 boosts the original entanglement rate by a whopping 366%. This surge can be attributed to the fact that many edges initially fail to entangle, and the few that succeed aren't optimally used. Recovery paths intervene, leveraging the successful edges to substantially elevate the probability. However, as seen in Figure 8b, the improvement margin attributed to Algorithm 5 remains relatively constant. Given that all quantum switches in these simulations share identical successful swapping probabilities, variations in this probability don't influence Algorithm 5's routing outcomes.

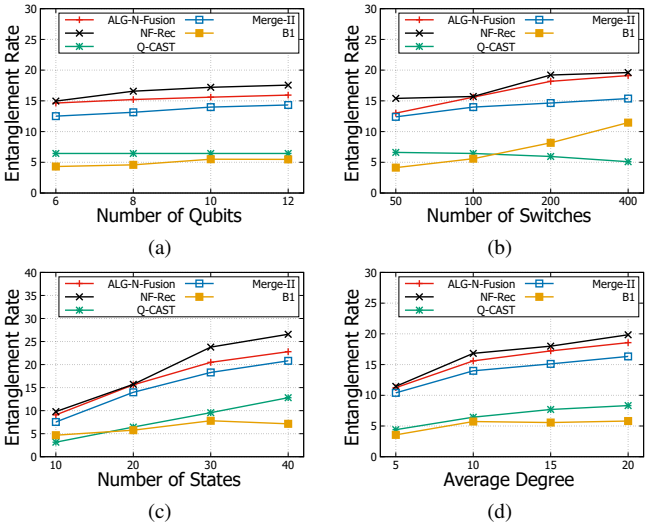


Fig. 7. (a) The network entanglement rate vs. the number of qubits in a switch. (b) The network entanglement rate vs. the number of switches. (c) The network entanglement rate vs. the number of quantum states to be shared. (d) The network entanglement rate vs. the average degree of a switch.

4) Quantum Parameters: Figure 8a and Figure 8b elucidate the network entanglement rates as influenced by the quantum link successful entanglement probability (i.e., p), and the entanglement-swapping probability (i.e., q), respectively.

From the observations in Figure 8a, as the average quantum link successful entanglement probability p is varied, noticeable shifts in the network entanglement rates become evident. By making an assumption that all links have a consistent p value,

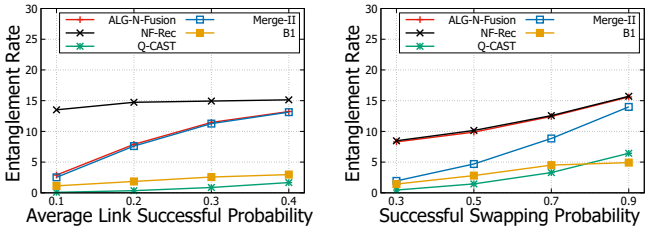


Fig. 8. (a) The network entanglement rate vs. average quantum link successful entanglement probability (i.e., p). (b) The network entanglement rate vs. switch entanglement-swapping probability (i.e., q).

we negate the variability stemming from network generation. With an escalation of p from 0.1 to 0.4, there's a pronounced surge in the network entanglement rate, peaking at an impressive 775% growth. Across all tested values of p, ALG-N-FUSION consistently delivers superior performance over other considered algorithms. Interestingly, the performance differential between ALG-N-FUSION and its competitors narrows as p increases. This accentuates the capability of ALG-N-FUSION to efficiently harness network resources, especially under real-world conditions where p tends to be on the lower side. Compared with our algorithm, the performance of Merge-II increases more with the increase of successful swapping probability. It may be because the width-preferred merge strategy significantly improves the entanglement rate when the probability of successful swapping of a single edge is low.

Moreover, the results highlight that n-fusion can still achieve a higher entanglement rate, even when its swapping rate is significantly lower than that of BSMs. For example, when the GHZ entanglements have swapping rates of 0.3, the entanglement rate calculated by our algorithm is still better than that of BSM entanglement algorithms with 0.9 swapping rate, i.e., Q-CAST.

5) Network Parameters: We test four network parameters in our simulations: the number of qubits in a switch; the number of switches; the number of quantum states to be shared between quantum-user pairs; and the average degree of a switch in the network. The results of varying these parameters are presented in Figure 7a, Figure 7b, Figure 7c, and Figure 7d respectively.

In this paper, we consider qubits in switches as the main limitation for entanglement routing. Increasing either the number of switches or the quantum capacity of each switch (i.e., more qubits per switch) directly expands the network's overall quantum capacity.

From Figure 7a, it is evident that a switch's performance is enhanced with a higher quantum capacity, i.e., more qubits. Figure 7b highlights a clear upward trend in the entanglement rate as the number of switches grows. However, a notable exception is Q-CAST. Its entanglement rates actually diminish with a greater number of switches, a consequence of its routing metrics struggling with extended distances, resulting in suboptimal resource utilization.

In Figure 7c, as the quantum state demands from user pairs swell, there is a corresponding surge in the entanglement rate. Meanwhile, Figure 7d delves into the relationship between a switch's average degree and the network entanglement rate. A higher average degree, indicative of more connections per switch, is correlated with an elevated entanglement rate. This increase in connections broadens the array of available paths for quantum users to share entangled states. The benefits of n-fusion become increasingly pronounced in such scenarios, with a denser network granting more flexibility for link fusion.

In summary, increasing network parameters like the number of qubits, switches, and node degree can expand the overall capacity of the network to serve quantum users and enhance the entanglement rate. Moreover, network performance is linked to user requests (the number of states to be shared). With a fixed set of requests, the incremental increase in entanglement rate due to network capacity becomes slower once most requests are satisfied. When physical distances are not significantly changed, increasing the number of switches could potentially reduce the entanglement rate due to longer hop distances between users. Therefore, it is crucial to carefully adjust network parameters when designing real quantum networks. Our results also underscore the efficacy of n-fusion, particularly in dense networks, and the robustness of our proposed algorithm across a variety of network configurations. It not only adapts well to diverse topologies but also capitalizes on network resources judiciously.

6) Simulation Summary:

1. N-fusion versus the traditional swapping. Through our simulations, it becomes evident that within a consistent network framework, with unchanged resources, the network's performance, as evaluated by the entanglement rate, is significantly superior under the n-fusion entanglement-swapping mechanism (employing GHZ measurements) as opposed to the traditional swapping technique (using BSMs).

To put this into perspective, when comparing to the Q-CAST, our algorithms ALG-N-FUSION enhance the network entanglement rate by an impressive 655%. This pronounced improvement can be attributed to the fact that n-fusion offers a more resource-efficient swapping mechanism relative to traditional methods. The capability of switches to fuse a greater number of quantum links bolsters the successful probability of entangling qubits between quantum-user pairs, even when the available network resources remain constant.

2. Performance under n-fusion. Under the same network entanglement-swapping method with n-fusion in the same network, ALG-N-FUSION can improve the network entanglement rate by up to 293% compared to B1 respectively. This indicates that our proposed ALG-N-FUSION is the most efficient algorithm among them as it can fully utilize the network resources to improve the network performance. Moreover, ALG-N-FUSION shows incomparable performance when the link's successful entanglement probability and the switch's successful swapping probability are small, which is a case closer to reality. For example, $p = 0.1$ from Figure 8a and $q = 0.3$ from Figure 8b. The reason for this is that ALG-N-FUSION can efficiently utilize qubits through merging links to increase the entanglement rate.

3. Merging Policies Comparison. The simulation data reveals that in the majority of scenarios, Merging Policy I (ALG-N-FUSION) demonstrates marginally superior efficacy compared to Merging Policy II (MERGE-II). This divergence

TABLE II
THE RUNNING TIME OF PROPOSED ALGORITHMS

No. Switches	Offline Stage	Total Online Stage	Average Online Stage
50	20460.2ms	37.6ms	0.752ms
100	61353.2ms	173.6ms	1.736ms
200	101670.6ms	590.6ms	2.953ms
400	105391.2ms	4728.8ms	11.822ms

in performance can be traced back to the construction of $\mathcal{A}_w$ from paths of the greatest width. Notably, when mapping paths for a singular shared quantum state, paths of varying widths might overlap.

Under this methodology, paths characterized by narrower widths may take precedence in route selection. This can inadvertently cause scenarios where broader paths are precluded from selection, especially if the qubits from the overlapping sections have already been appropriated by the narrower path. Such an occurrence can render a set of broad-width paths, which inherently have greater entanglement rates, ineligible for routing as the algorithm progresses.

On the other hand, prioritizing paths based on their larger widths, as seen in Algorithm 3, circumvents this issue. In this algorithm, paths characterized by their broader widths and inherently higher entanglement rates are selected with a higher priority. This mechanism ensures that Algorithm 3 consistently outperforms the aforementioned merging strategy.

4. Running Time of Online Stage

Quantum memory lifetimes of 1.46 seconds have been demonstrated on a microscopic nuclear-spin environment [55]. Given this short duration, our approach predetermines the entanglement routes under n-fusion in Phase I and then performs quantum operations like quantum link generation and swapping based on these routes, and recovers failed entanglement links to achieve end-to-end entanglement within a single entanglement period. To evaluate this, Table II shows the running time of our proposed algorithms in Phase I and Phase II, respectively. The running time is measured using standard C++ timing functions from the <time.h> library. The offline stage includes the total running time of Algorithm 1, 2, 3, and 4. The online stage includes the total running time of Algorithm 5, the entanglement distribution time, and the classical communication latency. Since different nodes have different running times, we compute the average running time of nodes. The results reveal that the running time of the offline stage is significantly higher than the running time of the online stage. As the number of switches in the network increases, the time consumed by the offline stage and the online stage increases, while the difference between them decreases. In a network with 400 quantum switches, the offline stage is 8915 times that of the online stage. The simulation results also show that our proposed online algorithms can be completed within the short existing entanglement time.

X. CONCLUSION

In this paper, we have introduced a general entanglement routing model for quantum networks with arbitrary topologies, where switches employ n-fusion, a general entanglement-swapping method. We have developed efficient algorithms to maximize the network entanglement rate for multiple quantum-user pairs. Extensive numerical evaluations have demonstrated that our proposed algorithms outperform existing approaches in terms of network performance under n-fusion. There is a vast potential for future research on the topic of n-fusion in quantum networks. Among the possibilities, we have identified two research topics directly related to this work: 1) The fundamental model presented in this paper can be readily extended to more sophisticated cases, such as fidelity-aware scenarios, and the proposed algorithms can be applied to support related quantum networking research areas, including quantum mapping and quantum network architecture design. 2) Design approximation algorithms for flow-like graphs in quantum networks. We expect that this paper will stimulate further research on the application of n-fusion in related fields, significantly contributing to the eventual success of quantum networks.

ACKNOWLEDGMENT

This work is supported in part by US National Science Foundation under grant numbers 1717731, 1730291, 2231040, 2230620, 2214980, 2046444, 2106027, and 2146909.

REFERENCES

[1] B. P. Lanyon, T. J. Weinhold, N. K. Langford, M. Barbieri, D. F. James, A. Gilchrist, and A. G. White, "Experimental demonstration of a compiled version of shor's algorithm with quantum entanglement," *Physical Review Letters*, vol. 99, no. 25, p. 250505, 2007.

[2] A. W. Harrow, A. Hassidim, and S. Lloyd, "Quantum algorithm for linear systems of equations," *Physical review letters*, vol. 103, no. 15, p. 150502, 2009.

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

[4] R. Valivarthi, S. I. Davis, C. Peña, S. Xie, N. Lauk, L. Narváez, J. P. Allmaras, A. D. Beyer, Y. Gim, M. Hussein *et al.*, "Teleportation systems toward a quantum internet," *PRX Quantum*, vol. 1, no. 2, p. 020317, 2020.

[5] H.-Y. Liu, X.-H. Tian, C. Gu, P. Fan, X. Ni, R. Yang, J.-N. Zhang, M. Hu, J. Guo, X. Cao *et al.*, "Optical-relayed entanglement distribution using drones as mobile nodes," *Physical Review Letters*, vol. 126, no. 2, p. 020503, 2021.

[6] Y.-A. Chen, Q. Zhang, T.-Y. Chen, W.-Q. Cai, S.-K. Liao, J. Zhang, K. Chen, J. Yin, J.-G. Ren, Z. Chen *et al.*, "An integrated space-to-ground quantum communication network over 4,600 kilometres," *Nature*, vol. 589, no. 7841, pp. 214–219, 2021.

[7] L. Yang, Y. Zhao, L. Huang, and C. Qiao, "Asynchronous entanglement provisioning and routing for distributed quantum computing," in *IEEE INFOCOM 2023-IEEE Conference on Computer Communications*. IEEE, 2023, pp. 1–10.

[8] X. Guo, C. R. Breum, J. Borregaard, S. Izumi, M. V. Larsen, T. Gehring, M. Christandl, J. S. Neergaard-Nielsen, and U. L. Andersen, "Distributed quantum sensing in a continuous-variable entangled network," *Nature Physics*, vol. 16, no. 3, pp. 281–284, 2020.

[9] M. Pant, H. Krovi, D. Towsley, L. Tassiulas, L. Jiang, P. Basu, D. Englund, and S. Guha, "Routing entanglement in the quantum internet," *npj Quantum Information*, vol. 5, no. 1, pp. 1–9, 2019.

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

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

[12] G. Vardoyan, S. Guha, P. Nain, and D. Towsley, "On the stochastic analysis of a quantum entanglement switch," *ACM SIGMETRICS Performance Evaluation Review*, vol. 47, no. 2, pp. 27–29, 2019.

[13] E. Shchukin, F. Schmidt, and P. van Loock, "Waiting time in quantum repeaters with probabilistic entanglement swapping," *Physical Review A*, vol. 100, no. 3, p. 032322, 2019.

[14] S. Das, S. Khatri, and J. P. Dowling, "Robust quantum network architectures and topologies for entanglement distribution," *Physical Review A*, vol. 97, no. 1, p. 012335, 2018.

[15] S. Shi and C. Qian, "Concurrent entanglement routing for quantum networks: Model and designs," in *Proceedings of the Annual conference of the ACM Special Interest Group on Data Communication on the applications, technologies, architectures, and protocols for computer communication*, 2020, pp. 62–75.

[16] S. Zhang, S. Shi, C. Qian, and K. L. Yeung, "Fragmentation-aware entanglement routing for quantum networks," *Journal of Lightwave Technology*, 2021.

[17] J.-W. Pan, D. Bouwmeester, H. Weinfurter, and A. Zeilinger, "Experimental entanglement swapping: entangling photons that never interacted," *Physical review letters*, vol. 80, no. 18, p. 3891, 1998.

[18] A. Patil, M. Pant, D. Englund, D. Towsley, and S. Guha, "Entanglement generation in a quantum network at distance-independent rate," *npj Quantum Information*, vol. 8, no. 1, pp. 1–9, 2022.

[19] 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 *2021 IEEE International Conference on Quantum Computing and Engineering (QCE)*. IEEE, 2021, pp. 334–345.

[20] C.-Y. Lu, T. Yang, and J.-W. Pan, "Experimental multiparticle entanglement swapping for quantum networking," *Physical review letters*, vol. 103, no. 2, p. 020501, 2009.

[21] C. Elliott, A. Colvin, D. Pearson, O. Pikalo, J. Schlafer, and H. Yeh, "Current status of the darpa quantum network," in *Quantum Information and computation III*, vol. 5815. International Society for Optics and Photonics, 2005, pp. 138–149.

[22] M. Peev, C. Pacher, R. Alléaume, C. Barreiro, J. Bouda, W. Boxleitner, T. Debuisschert, E. Diamanti, M. Dianati, J. Dynes *et al.*, "The secoqc quantum key distribution network in vienna," *New Journal of Physics*, vol. 11, no. 7, p. 075001, 2009.

[23] M. Sasaki, M. Fujiwara, H. Ishizuka, W. Klaus, K. Wakui, M. Takeoka, S. Miki, T. Yamashita, Z. Wang, A. Tanaka *et al.*, "Field test of quantum key distribution in the tokyo qkd network," *Optics express*, vol. 19, no. 11, pp. 10 387–10 409, 2011.

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

[25] C. Qiao, Y. Zhao, G. Zhao, and H. Xu, "Quantum data networking for distributed quantum computing: Opportunities and challenges," in *IEEE INFOCOM 2022-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 2022, pp. 1–6.

[26] G. Zhao, J. Wang, Y. Zhao, H. Xu, L. Huang, and C. Qiao, "Segmented entanglement establishment with all-optical switching in quantum networks," *IEEE/ACM Transactions on Networking*, 2023.

[27] Y. Zhao and C. Qiao, "Distributed transport protocols for quantum data networks," *IEEE/ACM Transactions on Networking*, 2023.

[28] M. Liu, J. Allcock, K. Cai, S. Zhang, and J. C. Lui, "Quantum networks with multiple service providers: Transport layer protocols and research opportunities," *IEEE Network*, vol. 36, no. 5, pp. 56–62, 2022.

[29] M. Ghaderibaneh, C. Zhan, H. Gupta, and C. Ramakrishnan, "Efficient quantum network communication using optimized entanglement swapping trees," *IEEE Transactions on Quantum Engineering*, vol. 3, pp. 1–20, 2022.

[30] Y. Zhao, G. Zhao, and C. Qiao, "E2e fidelity aware routing and purification for throughput maximization in quantum networks," in *IEEE INFOCOM 2022-IEEE Conference on Computer Communications*. IEEE, 2022, pp. 480–489.

[31] H. Gu, Z. Li, R. Yu, X. Wang, F. Zhou, and J. Liu, "Fendi: High-fidelity entanglement distribution in the quantum internet," *arXiv preprint arXiv:2301.08269*, 2023.

[32] Z. Li, K. Xue, J. Li, L. Chen, R. Li, Z. Wang, N. Yu, D. S. Wei, Q. Sun, and J. Lu, "Entanglement-assisted quantum networks: Mechanics, enabling technologies, challenges, and research directions," *IEEE Communications Surveys & Tutorials*, 2023.

[33] G. Vardoyan, E. van Milligen, S. Guha, S. Wehner, and D. Towsley, "On the bipartite entanglement capacity of quantum networks," *arXiv preprint arXiv:2307.04477*, 2023.

[34] Y. Zeng, J. Zhang, X. Shang, J. Liu, Z. Liu, and Y. Yang, "Multi-user entanglement routing design over quantum internets," in *2024 IEEE 44th International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 2024, pp. 266–276.

[35] E. A. V. Milligen, E. Jacobson, A. Patil, G. Vardoyan, D. Towsley, and S. Guha, "Entanglement routing over networks with time multiplexed repeaters," 2024. [Online]. Available: https://arxiv.org/abs/2308.15028

[36] J. W. Essam, "Percolation theory," *Reports on progress in physics*, vol. 43, no. 7, p. 833, 1980.

[37] E. Sutcliffe and A. Beghelli, "Multiuser entanglement distribution in quantum networks using multipath routing," *IEEE Transactions on Quantum Engineering*, vol. 4, pp. 1–15, 2023.

[38] L. Bugalho, B. C. Coutinho, F. A. Monteiro, and Y. Omar, "Distributing multipartite entanglement over noisy quantum networks," *quantum*, vol. 7, p. 920, 2023.

[39] C. Clayton, X. Wu, and B. Bhattacharjee, "Efficient routing on quantum networks using adaptive clustering," in *2024 IEEE 32nd International Conference on Network Protocols (ICNP)*. IEEE, 2024, pp. 1–12.

[40] M. A. Nielsen and I. Chuang, "Quantum computation and quantum information," 2002.

[41] O. Crawford, B. van Straaten, D. Wang, T. Parks, E. Campbell, and S. Brierley, "Efficient quantum measurement of pauli operators in the presence of finite sampling error," *Quantum*, vol. 5, p. 385, 2021.

[42] IBM. (2023) Ibm debuts next-generation quantum processor & ibm quantum system two, extends roadmap to advance era of quantum utility. [Online]. Available: https://newsroom.ibm.com/2023-12-04-IBM-Debuts-Next-Generation-Quantum-Processor-IBM-Quantum-System-Two,-Extends-Roadmap-to-Advance-Era-of-Quantum-Utility

[43] A. Paverd, A. Martin, and I. Brown, "Modelling and automatically analysing privacy properties for honest-but-curious adversaries," *Tech. Rep*, 2014.

[44] O. Goldreich, *Foundations of cryptography: volume 2, basic applications*. Cambridge university press, 2001, vol. 2.

[45] L. R. Ford and D. R. Fulkerson, "Maximal flow through a network," *Canadian journal of Mathematics*, vol. 8, pp. 399–404, 1956.

[46] A. Dahlberg, M. Skrzypczyk, T. Coopmans, L. Wubben, F. Rozpędek, M. Pompili, A. Stolk, P. Pawełczak, R. Knegjens, J. de Oliveira Filho *et al.*, "A link layer protocol for quantum networks," in *Proceedings of the ACM Special Interest Group on Data Communication*, 2019, pp. 159–173.

[47] A. Tantillo, "Quantum repeaters use defects in diamond to interconnect quantum systems," http://https://news.mit.edu/2023/quantum-repeaters-use-defects-diamond-interconnect-quantum-systems-0927, 2023, september 27, 2023.

[48] N. Benchasattabuse, M. Hajdušek, and R. Van Meter, "Engineering challenges in all-photonic quantum repeaters," *IEEE Network*, 2024.

[49] E. W. Dijkstra, "A note on two problems in connexion with graphs," in *Edsger Wybe Dijkstra: His Life, Work, and Legacy*, 2022, pp. 287–290.

[50] J. Y. Yen, "Finding the k shortest loopless paths in a network," *management Science*, vol. 17, no. 11, pp. 712–716, 1971.

[51] Y. Zeng, J. Zhang, J. Liu, Z. Liu, and Y. Yang, "Entanglement routing design over quantum networks," *IEEE/ACM Transactions on Networking*, 2023.

[52] M. Hassani, "88.18 cycles in graphs and derangements," *The Mathematical Gazette*, vol. 88, no. 511, pp. 123–126, 2004.

[53] B. M. Waxman, "Routing of multipoint connections," *IEEE journal on selected areas in communications*, vol. 6, no. 9, pp. 1617–1622, 1988.

[54] M. Pompili, S. L. Hermans, S. Baier, H. K. Beukers, P. C. Humphreys, R. N. Schouten, R. F. Vermeulen, M. J. Tiggelman, L. dos Santos Martins, B. Dirkse *et al.*, "Realization of a multinode quantum network of remote solid-state qubits," *Science*, vol. 372, no. 6539, pp. 259–264, 2021.

[55] M. H. Abobeih, J. Cramer, M. A. Bakker, N. Kalb, M. Markham, D. J. Twitchen, and T. H. Taminiau, "One-second coherence for a single electron spin coupled to a multi-qubit nuclear-spin environment," *Nature communications*, vol. 9, no. 1, p. 2552, 2018.

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

[57] D. Volchenkov and P. Blanchard, "An algorithm generating random graphs with power law degree distributions," *Physica A: Statistical Mechanics and its Applications*, vol. 315, no. 3-4, pp. 677–690, 2002.

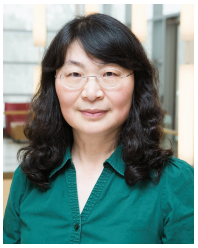
[58] S. Even, A. Itai, and A. Shamir, "On the complexity of time table and multi-commodity flow problems," in *16th Annual Symposium on Foundations of Computer Science (sfcs 1975)*. IEEE, 1975, pp. 184–193.

Yiming Zeng is an assistant professor in the School of Computing at Binghamton University (SUNY at Binghamton). He earned his Ph.D. from Stony Brook University, New York, USA, and his B.Eng. degree from Shanghai Jiao Tong University, Shanghai, China. His research currently centers on quantum networking and quantum computing.

Jiarui Zhang received the BEng degree in computer science and technology from Shanghai Jiao Tong University in 2017. He is currently working towards the PhD degree in computer engineering at Stony Brook University. His research interests include blockchain, mobile edge computing, quantum networking and computing.

Ji Liu 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 in the Department of Electrical and Computer Engineering at Stony Brook University, Stony Brook, NY, USA. He is an Associate Editor of the IEEE Transactions on Signal and Information Processing over Networks. His current research interests include distributed control and optimization, distributed reinforcement learning, resiliency of distributed algorithms, epidemic and social networks, quantum computing and networking.

Zhenhua Liu is an Associate Professor of Operations Research in the Department of Applied Mathematics and Statistics and affiliated with the Department of Computer Science at Stony Brook University (SUNY at Stony Brook). He received his PhD in Computer Science from California Institute of Technology, under the supervision of Adam Wierman and Steven Low. 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 was recently awarded an 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, 2021 ACM SIGMETRICS Rising Star Research Award, 2021 ACM SIGMETRICS Test of Time Award, the Pick of the Month award by IEEE STC on Sustainable Computing, a SPEC Distinguished Dissertation Award (honorable mention), an NSF CAREER award, and several Excellence in Teaching awards.



Yuanyuan Yang received the BEng and MS degrees in computer science and engineering from Tsinghua University, Beijing, China, and the MSE and Ph.D. degrees in computer science from Johns Hopkins University, Baltimore, Maryland. She is a SUNY Distinguished Professor of computer engineering and computer science at Stony Brook University, New York, and an IEEE Fellow and National Academy of Inventors (NAI) Fellow. She worked as a Program Director at the National Science Foundation before. She has 30+ years of combined experience in parallel computing, cloud computing, optical networking, and quantum computing, and published 500+ papers in these areas, including 3 best paper awards and 6 best paper runner-ups. Her work has culminated in the development of fundamental theoretical advancements and empirical simulation tools, all important background and experiences for the proposed project. She is currently the Editor-in-Chief for IEEE Transactions on Cloud Computing and an Associate Editor for 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, Associate Editor-in-Chief, and Associated Editor for IEEE Transactions on Computers, and Associate Editor for IEEE Transactions on Parallel and Distributed Systems. She has also served as a general chair, program chair, or vice chair for several major conferences and a program committee member for numerous conferences.

APPENDIX

PROOF OF THEOREM 1

Proof. The decision version of NERP is to determine whether there exists an n-fusion entanglement routing solution for k quantum user pairs. To show the NP-Completeness of this problem, we give the definition of the simple Two-Commodity Integral Flow in Directed graphs (SD2CIF) problem [58].

Given a directed finite graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. A directed edge from u to v is denoted $e_{u,v} \in \mathcal{E}$. The capacities of all edges are equal to one. Given sources $s_1, s_2 \in \mathcal{V}$, and terminals $t_l, t_2 \in \mathcal{V}$. The problem is to determine whether there exist two flow functions f_l and f_2 such that

(a) For every $e_{u,v} \in \mathcal{E}$, $f_1(u,v) + f_2(u,v) \leq 1$.

(b) For each commodity $i \in \{1,2\}$ and each vertex $v \in \mathcal{V} - \{s_i, t_i\}$, $\sum_{e_{u,v} \in \mathcal{E}} f_i(u,v) = \sum_{e_{v,w} \in \mathcal{E}} f_i(v,w)$.

(c) For each commodity $i \in \{1,2\}$ let the total flow be $F_i = \sum_{e_{s_i,v} \in \mathcal{E}} f_i(s_i,v) - \sum_{e_{v,s_i} \in \mathcal{E}} f_i(v,s_i)$. Then it is required that $F_1 + F_2 = R$, while R is a positive integer called requirement.

An input of SD2CIF can map to an input of NERP. All vertices from SD2CIF are mapped one-to-one to quantum users or switches in NERP with infinite capacity, as capacity constraints in SD2CIF are applied to edges, not vertices. While SD2CIF is set on a directed finite graph, we can regard each undirected edge in NERP as two directed edges in different directions. The main difference in inputs between NERP and SD2CIF is that each edge in SD2CIF has 1 capacity limit, while each switch v in NERP has capacity Q_v. To transfer the capacity conditions, we can add a vertex for each edge in SD2CIF. To represent edge $e_{i,j}$ in SD2CIF, we use one switch v_{ij} and two links $e_{i,v_{ij}}$, $e_{v_{ij},j}$ in NERP. v_{ij} with capacity 1 can implement the edge capacity limitation of SD2CIF.

A solution to NERP can be reduced to a solution to SD2CIF. If we have a NERP solution, we can trace two entanglement paths that connect the sources to the terminals, each representing a solution for SD2CIF. Conversely, a solution to SD2CIF can serve as a solution to NERP as a 2-fusion entanglement routing problem. To verify a solution to NERP in polynomial time, we can compute the entanglement rate by the methods proposed in Section V. Therefore, NERP is in NP.

Since SD2CIF has been proven to be NP-Complete [58], and we can reduce SD2CIF to NERP in polynomial time, the decision version of NERP is NP-Hard. As NERP is in NP, this establishes NERP as an NP-Complete problem. □

TIME COMPLEXITY ANALYSIS

Algorithm 1

The time complexity of Algorithm 1 is $O(|\overline{\mathcal{V}}| \log |\overline{\mathcal{V}}| + |\mathcal{E}|)$. The algorithm will traverse all $v_i \in \overline{\mathcal{V}}$, and all $e_{v_i,v_j} \in \mathcal{E}$.

Algorithm 2

The time complexity of Algorithm 2 is $O(|\mathcal{M}|\mathrm{W}h|\overline{\mathcal{V}}|(|\overline{\mathcal{V}}| \log |\overline{\mathcal{V}}| + |\mathcal{E}| + \log h))$. The terms $|\mathcal{M}|$, W, h, and $|\overline{\mathcal{V}}|$ correspond to the loop iterations in lines 1, 2, 6, and 10, respectively. The inner loop involves running Algorithm 1 and operations on a priority queue, which together have a complexity of $O(|\overline{\mathcal{V}}| \log |\overline{\mathcal{V}}| + |\mathcal{E}| + \log h)$.

Algorithm 3

The time complexity of Algorithm 3 is $O(h(\log h + |\overline{\mathcal{V}}|))$. The algorithm sorts paths in $\mathcal{A}$ in line 2, with a time cost of $O(h \log h)$. The algorithm enumerates all paths in $\mathcal{A}$ in line 3, with a time cost of $O(h)$. In line 5, it iterates over all edges within each path, incurring a cost of $O(|\overline{\mathcal{V}}|)$ per path. The total time complexity is $O(h(\log h + |\overline{\mathcal{V}}|))$.

Algorithm 4

The time complexity of Algorithm 4 is $O(\mathrm{W}|\mathcal{E}||\mathcal{M}|(|\overline{\mathcal{V}}| + |\mathcal{E}|))$. Theoretically, the loop covered by line 1 runs $O(\mathcal{E})$ times, and line 2 runs $O(\mathrm{W})$ times. Line 3 computes the increment of expected entanglement probability with $O(|\overline{\mathcal{V}}| + |\mathcal{E}|)$ complexity for $O(|\mathcal{M}|)$ times.

The actual time cost is much lower for Algorithm 4, since most switches may not retain W free qubits, thus the $\mathrm{W}|\mathcal{E}|$ term is loose. Theoretically, considering the W dependency, the algorithm is pseudo-polynomial. However, the algorithm's time complexity remains acceptable since the W term is relatively loose.

As a comparison, the algorithm from [16] that resolves the problem under BSM has the time complexity $O(|P|(|\mathcal{V}| \log |\mathcal{V}| + |\mathcal{E}|(h_m \mathrm{W}))$, when $|P|$ is the number of found paths during the algorithm, and h_m is the maximum possible length of path. Comparing this complexity to the complexity of Algorithm 3, $O(|P|) = O(h|\overline{\mathcal{V}}|)$, so the two algorithms are at the same level in terms of time complexity.

Algorithm 5

We analyze the time complexity of Algorithm 5 in two segments: the information exchange process and the entanglement establishment process. The complexity for the information exchange, spanning Line 1 to Line 14, stands at $O(|\mathcal{E}| + |\mathcal{S}||\mathcal{V}| \log(|\mathcal{S}||\mathcal{V}|))$. However, the practical time consumption might be less, given that $v' \in \mathcal{V}$ needs to be within H-hops from v. For the entanglement establishment, since it's set to operate over a fixed total time T divided into $\lceil T/t \rceil$ slots, we focus our analysis on individual time slots. Within each slot, v evaluates $\mathcal{R}$ once and then waits for incoming requests. Upon receiving a first-round request, v inspects up to $\lceil T/t \rceil$ elements in $\mathcal{F}$. This results in a time complexity of $O(|\mathcal{S}||\mathcal{V}|)$ for each slot. Each processed first-round request has a complexity of $O(\lceil T/t \rceil)$. In conclusion, the given time complexity is sufficient for v to execute Algorithm 5.

TABLE III
DEFAULT SIMULATION PARAMETERS

Network size	100×100 unit square
Number of unique quantum-user pairs	20
Number of quantum switches	100
Average minimum path hop	4.2
Success n-fusion probability	0.9
α	0.01.
H	5
Classical communication delay	$2.07 * 10^8 m/s$