Topology Design with Resource Allocation and Entanglement Distribution for Quantum Networks

***Abstract*—Topology is one of the most critical properties of networks. Quantum networks, as a new type of network, have fundamentally different principles for establishing connections compared to classical networks, leading to distinct challenges in topology design. Finding the optimal topology for quantum networks to meet traffic demands is a crucial yet not fully understood problem. In this paper, we explore the topology design problem for quantum networks, considering both resource allocation and entanglement distribution. We propose and investigate both flow-based and path-based formulations, along with their associated solutions, aimed at minimizing the topology cost. For the path-based formulation, we also provide the first theoretical analysis of the cost associated with swapping strategies over a quantum path. Extensive simulations demonstrate that our enhanced path-based formulation is both efficient and effective.**

I. Introduction

While quantum networks [1], [2] enable many reforming applications and are hence in active development in many countries, we still know little about its design principles at the network scale. Experimental quantum networks typically consist of only a few quantum links, which do not require a careful network-wide design. On the other hand, we foresee that in the near future, quantum networks at larger scales, such as the global quantum internet, will be of great interest. Topology, perhaps the most critical factor in network design [3]–[6], should also be of great importance to these networks. While there have already been many works on connection establishment (i.e., entanglement distribution) for quantum networks with given topology, few researchers consider how we design such topology.

Topology design for quantum networks is different from that of classic networks: they have fundamentally different logic for establishing connections (i.e., quantum entanglements). In classic networks, connections are basically virtual tunnels where hosts can exchange data packages. The packages are sent into the network and transmitted to routers/switches. In quantum networks, such a strategy is impossible: the no-cloning principle in quantum mechanics prevents from perfectly copying quantum status. Instead of pushing quantum status through the network, entanglements are created between two nodes (or among multiple nodes). These entanglements can be consumed by teleportation to transmit quantum status, and thus are the key resource in quantum networks [7], [8].

Usually, entanglements are first generated over short quantum links and then connected to form long-distance ones via *quantum swapping*. We cannot establish long-distance entanglements directly because photons (the matter mostly used to carry quantum information) suffer exponential loss in fibers. To overcome this, multiple auxiliary techniques can be adopted. The simplest one is to shorten the link length, e.g., put intermediate quantum repeaters (similar to the routers in classic networks) to segment long links (so shorter sub-links and higher photon survival rate). The second one is to deploy multiple channels on one link so these channels can generate entanglements simultaneously, resulting in a higher link-level entanglement rate. The last but not the least, repeaters are equipped with quantum memory so asynchronous swappings are enabled. Without quantum memory, we can perform swapping to connect two adjacent entanglements only when both two sides establishes one entanglement successfully at the same time. Since when the photons arrive, we either process them immediately, or they get lost as we cannot store them. This is obviously unfavorable to a higher E2E entanglement rate. Therefore, we can use quantum memory to 'store' photons: they are absorbed by atoms and the information they carry is stored. When we need to perform swapping, we can read that information out.

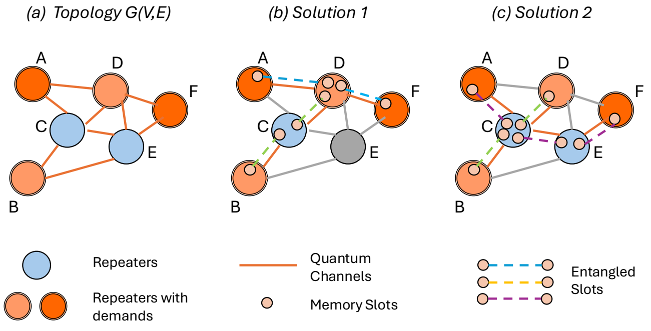


Fig. 1: Topology design problem in quantum networks. (a) A topology $G(V, E)$ shows all possible quantum repeaters/links; (b) and (c) are two different TD solutions, both can serve the two traffic demands while using different quantum resources.

These techniques reveal two important device resources of quantum networks: *quantum channels* and *quantum memory*. In the topology design (TD) problem of quantum networks, we focus on optimizing these resources needed to construct a quantum network for specific traffic demands. For example, in Fig. 1, to fulfill a traffic demand between the source-destination (SD) pair A and F (denoted as A:F), entanglements over two different quantum paths can be used, i.e., the blue path (A $\leftrightarrow$ D $\leftrightarrow$ F) and the purple path (A $\leftrightarrow$ C $\leftrightarrow$ E $\leftrightarrow$ F). The blue path is shorter (in terms of hops), which implies that less memory is needed to establish one single E2E entanglement for A:F. On the other hand, links A $\leftrightarrow$ D and D $\leftrightarrow$ F can be longer than the links in the purple path, which means they are weaker in entanglement generation so more quantum channels (higher cost) may be required to achieve the same capacity. Therefore, there are clear memory-channel trade-offs in topology design. In addition, when we also need a green path (B ↔ C ↔ D) for pair B:D, if we select the blue path (rather than purple path) for A:F, then node E and its adjacent links (i.e., E ↔ C, E ↔ D, and E ↔ F) can be removed to save resource/cost. It is clear that, for the same demands (e.g., one entanglement for each of pair A:F and B:D), different TD decisions (e.g., blue path + green path (Solution 1 in Fig. 1b) v.s. purple path + green path (Solution 2 in Fig. 1c)) have different cost. When the network is large and there are many demanding pairs, it is challenging to find the topology with the optimal cost. Therefore, the *optimal topology design* is critical for large-scale quantum networks.

In this paper, we first formulate a *topology design problem* (TDP) based on two popular entanglement distribution formulations: *flow-based* and *path-based* formulations. The flow-based one (F-TDP), if optimally solved, gives the optimal result of the TDP. However, the complexity of F-TDP makes it challenging to solve for larger and denser networks. The path-based formulation (P-TDP), on the other hand, restricts the entanglement flow over pre-defined paths, thus can be solved more efficiently but with a certain loss on the optimality. To obtain an efficient and effective solution over pre-defined paths, we dive deep into the path-level entanglement distribution and are able to find the optimal path-level solution. Then, once equipped with our optimal path-level solution, the path-based formulation can give similar results to the flow-based one, and remains to be efficient. Lastly, we also propose a greedy algorithm, which is guaranteed to terminate in polynomial time. Overall, our contributions in this paper can be summarized as:

- We investigate the TDP for quantum networks with two formulations (F-TDP and P-TDP), which led to two different TD solutions. In addition, a greedy method is proposed for P-TDP.
- To enhance the P-TDP, we provide a formal analysis of the path-level entanglement distribution. We prove the swapping strategy based on a relaxed complete swapping tree can lead to the optimal expected cost, while the current dominating swapping strategy (sequential swapping) leads to the worst cost. To the best of our knowledge, this is the first theoretical analysis of the cost of swapping strategies on repeater chains (in time slot modeling).
- Extensive experiments are conducted to confirm that the path-based solution with our optimal path-level entanglement is efficient and loses only marginal optimality. In addition, the impact of key factors on topology design is explored, such as segmentation length, device price ratio, demand intensity, etc.

II. Related Works

Entanglement Distribution: Entanglement distribution is the core function of quantum networks, thus has been well-studied most recently. The current methods can be categorized into two groups: *flow-based* and *path-based*.

The *flow-based* formulation, initially proposed by ORED [9], uses three-tuples of edges (or node pairs) to describe all possible swappings in the whole network and formulate the distribution problem as a flow-based optimization to maximize the entanglement throughput. Follow-up works (such as FENDI [10] and ESDI [11]) further improve ORED in terms of latency and/or fidelity. Our formulation in Section IV follows this group but additionally considers resources.

Rather than considering all possible swappings, *path-based* formulation confines the distribution within paths: each SD pair has multiple candidate paths, which are pre-solved (for their distribution strategy), then the path selection is performed at the network scale. The dominant path-level solution typically employs sequential swapping. Farahbakhsh and Feng [12] design an opportunistic routing method to reduce delay. Zeng *et al.* [13] propose a routing algorithm (with predetermined paths) for network throughput maximization. Yang *et al.* [14] develop an online entanglement routing scheme to suffice real-time requests. Zhao *et al.* [15] and Li *et al.* [16] also integrate purification to improve end-to-end connection quality. We consider a path-based formulation in Section V.

Quantum Network Planning: While topology design is a relatively new topic in quantum networks, there are a few pioneer works on quantum network planning, though in a limited range. Yu *et al.* [17] explore the network performance with simple/random topologies, such as chain, ring, and random graphs of specific patterns. Pouryousef *et al.* [18] proposes a formulation where repeaters can be placed at pre-defined positions (typically along the lines between two existing pairs) to maximize the network utility. Chehimi *et al.* [19] explore the scaling problem of quantum networks due to many factors, including noise and fidelity decay. Our formulations are different from one or more of them in the following aspects: i) we use a time-slot-based model [15], [16], [20]; ii) we allow flexible resource allocation for both quantum channels and memory, instead of only repeaters; iii) we allow arbitrary swapping order (and find the optimal one), instead of only using the simplest sequential one.

Analysis on Quantum Repeater Chains: Theoretical analysis of the entanglement distribution rate of the whole network is still an open problem, while most existing works obtain the expected network-wide rate by solving an optimization problem [9], [15]. Most analysis works focused on the path-level (i.e., repeater chains) throughput, such as [18], [21]. A few works aim to obtain the optimal strategy for other metrics, such as latency [22], [23]. However, again, their analysis is typically one-shot based and/or only applies to sequential swapping. Our optimal path-level analysis in Section VI, rather, is time-slot based, allows arbitrary swapping orders (and finds the optimal one).

III. Preliminaries and Modeling

A. Network Model

As shown in Fig. 1, quantum networks include both quantum nodes and quantum links/channels.

Quantum Nodes: Each node v is a quantum repeater equipped with sufficient identical swapping gates and a quan-

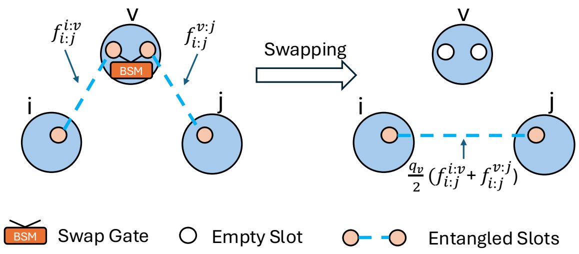


Fig. 2: Illustration of swapping flow at node v for entanglements between i and j.

tum memory with m_v memory slots. m_v is a decision variable for our topology optimization. Here, sufficient gates mean that we can conduct as many swappings as we want in a certain time period. Actually, at most $\lceil \frac{m_v}{2} \rceil$ swapping gates are needed to process all m_v qubits stored in the memory at once. Gate reuse rate depends on the specific implementations. Here we assume that the gate number is proportional to the memory slots so the gate cost can be merged into memory cost. The success probability of swapping at v is q_v, and $q_v \in [0.5, 1]$[1]. Each node costs γ_v^0 to be installed (when $m_v > 0$) and each memory slot on node v costs γ_v. Therefore, the cost of node v used by the topology is $b_v = \gamma_v m_v + \mathbb{I}_v \gamma_v^0$ where $\mathbb{I}_v = 1$ if $m_v > 0$, otherwise 0.

Quantum Links/Channels: Each edge e in the network connects two end nodes with c_e optical channels with length of l_e. Here, c_e is another decision variable for our topology optimization. Each optical channel includes optical fibers, photon sources, and detectors. It can generate k_e entanglements in the given time interval[2]. Therefore, edge e is able to generate $k_e c_e$ entanglements in one time slot. This basically means that we can obtain higher EGR over an edge by deploying multiple channels at a higher cost, e.g., using multi-core fibers (together with more powerful optics). Similarly, we assume that each edge costs γ_e^0 to initialize if $c_e > 0$ and each channel on the edge costs $\gamma_e l_e$, leading to $\gamma_e l_e c_e$ for all channels of edge e. Here the γ_e is the per-kilometer price for one channel. Therefore, the total cost of edge e is $b_e = \gamma_e l_e c_e + \mathbb{I}_e \gamma_e^0$, where $\mathbb{I}_e = 1$ if $c_e > 0$, otherwise 0.

Quantum Network: A quantum network can be modeled as a graph $G = (V, E)$ with all quantum nodes V and all possible quantum links E among V. This graph with the properties of its nodes/edges (such as location, swapping success probability, price of nodes, and length, n_e, price of edges) are known, and thus are the inputs of our TDP.

Quantum Users: We assume that users (applications) sit at quantum nodes and demand a certain number of entanglements between certain SD pairs. We use $D = \{d_{i,j} | \forall i, j \in V\}$ to represent all user demands, where $d_{i,j}$ is the demand between nodes i and j.

B. Problem Formulation

The goal of our topology design problem is to find a network topology (by choosing certain nodes and edges from the initial quantum network and assigning sufficient resources (such as memory slots and channels) to these nodes/edges to serve the user demands (i.e. entanglements between SD pairs) while minimizing the cost of such topology. Such an optimization problem is challenging since not only the resulting topology and allocated resources need to be jointly considered but also the entanglement distribution (or swapping scheduling) among these selected nodes need to be determined and optimized to fulfill the user demands.

Entanglement Distribution Scheduling: A scheduling strategy F_V describes swapping operations on each node $v \in V$. In other words, F_V describes which entanglement is consumed by swapping operations to generate which entanglement. While there are different modeling methods in different formulations, we use the notation introduced by [9] to describe the swappings because it is general enough in our considerations. For an swapping operation on node v as shown in Fig. 2, $f_{i:j}^{i:v}$ and $f_{i:j}^{v:j}$ define the number of entanglements between node pair (i, v) and (v, j) consumed to generate the entanglements between (i, j). A complete scheduling strategy F_V contains all possible three-tuples (i.e., $\forall (i, v, j) \in V$), although many of them could be 0. For example, for path-based scheduling, if i, v, j is not on the same path, then we always have $f_{i:j}^{i:v} = f_{i:j}^{v:j} = 0$.

The formal definition of *Topology Design Problem* (TDP) can then be defined as: given the quantum network and user demands, we need to make the topology, resource allocation and entanglement scheduling decisions to fulfill the demands with minimal cost. We use M_V, C_E, and F_V to represent the sets of memory, channel and swapping decisions, respectively. Note that if the memory/channel is selected as 0, that node/edge is not included in the final topology. We also use P_G to represent the total cost of the final topology, which is the summation of all costs at selected nodes and edges in the topology.

Problem 1: **Topology Design Problem (TDP).** Given a quantum network $G(V, E)$ and user demands D, find the optimal M_V, C_E and F_V, such that i) all user demands are satisfied, and ii) the total cost P_G is minimized.

In the next two sections, we introduce two different sets of methods to solve the TDP or its variations: a 'flow'-based methods and path-selection based methods. They have different complexities and loss of optimality, but we will see that the path-selection based solution is efficient and close to optimal solution if properly designed.

[1] To perform swapping, we need to use Bell State Measurement (BSM), which typically includes one controlled-NOT (CNOT) gate plus classic communications. Linear optical BSM is efficient, but can only distinguish 2 out of the 4 Bell states, which theoretically limits its success probability to up to 50%. Higher rates (e.g., 62.5% [24], 75% [25]) are possible by further improvements. Deterministic BSM gates [26], however, usually do not fail, so they do not have such theoretical upper bounds of success probability. Of course, they are still inevitably affected by noise.

[2] The surviving probability of a photon after traveling through the optical fiber is $p_e = 10^{-\frac{1}{10}\alpha l_e}$, where α is the loss constant of the fiber. The surviving probability drops drastically (exponentially) as the length increases. This is the reason why we need intermediate nodes (repeaters) to establish long-distance entanglements. Besides, each channel is equipped with a light source, which can emit n_e photons at a specific frequency/wavelength in one time slot. Thus, the number of entanglements generated by one single channel in expectation is $k_e = n_e p_e$.

IV. Flow Formulation and Solution

We first introduce the flow-based formulation of TDP (F-TDP). The entanglement distribution part of this formulation is incorporated from ORED [9], which is also the foundation of several existing works, such as FENDI [10] and ESDI [11]. In [9], this formulation solves the entanglement distribution problem alone: there is almost no resource consideration (node memory and channel/edge capacity). It simply uses the edge entanglement generation probability p_e to control the available entanglements in the whole network. To make it suitable for a TDP formulation, we add the resource-related constraints and replace the objective with the total cost. This new formulation of F-TDP is able to jointly optimize topology, resource and entanglement distribution all together.

$$\textbf{F-TDP}: \min_{m,c,\phi,f} \sum_{v\in V} b_v + \sum_{e\in E} b_e \tag{1}$$

s.t. entanglement distribution constraints

$$I(i,j) \leq \phi_{ij} + \sum_{v\in V\setminus\{i,j\}} \frac{q_v}{2}(f_{i:j}^{i:v} + f_{i:j}^{v:j}), \forall i,j \in V \tag{2}$$

$$O(i,j) = \sum_{v\in V\setminus\{i,j\}} (f_{i:v}^{i:j} + f_{v:j}^{i:j}), \qquad \forall i,j \in V \tag{3}$$

$$f_{i:j}^{i:v} = f_{i:j}^{v:j}, \qquad \forall i,j,v \in V \tag{4}$$

$$I(i,j) - O(i,j) \geq d_{i,j}, \qquad \forall i,j \in V \tag{5}$$

resource constraints

$$\sum_{e=(v,v')\in E} \phi_e \leq m_v, \qquad \forall v \in V \tag{6}$$

$$\phi_e \leq k_e c_e, \qquad \forall e \in E \tag{7}$$

decision variables

$$m_v, c_e \in \mathbb{Z}_0^+, \quad \phi_e, f_{i,j}^{i,v}, f_{i,j}^{v,j} \in \mathbb{R}_0^+ \tag{8}$$

The objective (1) is to minimize the total cost of the final topology, which can be written as $\sum_{v\in V} b_v + \sum_{e\in E} b_e$. The decision variables (8) are (i) the memory size m_v at node v, (ii) the number of channels c_e on edge e, (iii) the actual entanglements used ϕ_e at e, and (iv) the number of used entanglements $f_{i,j}^{i,v}$ and $f_{i,j}^{v,j}$ between node pair (i,v) and (v,j) to generate the entanglements between i and j. Note that we introduce ϕ_e to denote the expected number of used entanglements to save the memory. While edge e can generate $k_e c_e$ entanglements, we may only need part of them. For example, when $k_e = 5$ and we need 7 entanglements, we can set $c_e = 2$ to get up to 10 entanglements on this edge. However, if we store all of them, we need 10 memory slots on each end node of this edge. On the other hand, we can simply discard 3 surplus entanglements and keep only 7 in the memory on each end node. In this way, we can save 3 memory slots at each end node.

The entanglement distribution constraints (2)-(5) are similar constraints adopted from ORED [9] (except ϕ). Constraints (2) and (3) define the in-flow and out-flow between every node pair (i,j): in-flow $I(i,j)$ is the sum of the number of directly generated entanglements ϕ_{ij} (when there is an edge between the pair) and the number of entanglements obtained from swappings; out-flow $O(i,j)$ is the number of entanglements used by swappings to generate other entanglements. Fig. 2 shows an example of a swapping happens on node v that generates entanglements between nodes i and j. $f_{i:j}^{i:v}$ and $f_{i:j}^{v:i}$ are the contributions of entanglements between nodes $i : v$ and nodes $v : j$, which are used to generate $i : j$. The coefficient $\frac{q_v}{2}$ follows that the swapping takes two entanglements as input but generates one entanglement as output, and the swapping succeeds with probability q_v (so the flow variable f is the expected value). Constraint (4) requires that when node v performs a swapping to generate entanglements between pair (i,j), an equal number of entanglements from (i,v) and (v,j) should be used. Constraint (5) makes sure enough entanglements are left for the pairs demanding entanglements. For more details and explanations of these constraints, we refer readers to [9], [10]. Constraints (6) and (7) are resource constraints. Constraint (6) ensures that the memory usage of all entanglement generation (ϕ_e) at node v does not exceed the device capability (m_v). Constraint (7) makes sure that the entanglement consumed at edge e does not exceed the edge capability. Note that swappings do not introduce new memory costs because they are in place: they update the entanglement relations among qubits but do not generate/require new qubits. Therefore, all memory cost comes from entanglement generation (to store the link-level entanglements).

This formulated optimization problem F-TDP, if optimally solved, gives us the optimal solution to TDP. On the other hand, it is a Mixed Integer Linear Programming (MILP) problem, which is challenging to solve. The number of decision variables includes $O(|V|)$ m_v, $O(|E|)$ c_e, $O(|E|)$ ϕ_e, and $O(|V|^3)$ of $f_{i,j}^{i,v}$. When the network is large and dense, directly solving this MILP may become very challenging. Thus, we also explore the path-selection based formation and solution.

V. Path Selection Formulation and Solutions

In this section, we first explore a new formulation of TDP by restricting the entanglement scheduling on pre-defined candidate paths, then propose a greedy path selection method. In Section VI, we will discuss how to find the optimal entanglement along a pre-defined candidate path.

A. Path Selection based TDP

Now, we consider a new formulation of TDP, a path-selection based TDP (P-TDP). Path-based formulations are used by most works on entanglement distribution (not including topology and resource), such as routing [15] and utility maximization [18]. Unlike the flow-based formulation, path-based formulation restricts the entanglement scheduling strategy to path selection. In other words, pre-defined candidate paths are given for each source-destination pair, and these paths are solved (find F_V for those paths) in advance (later in Section VI we will provide our analysis and algorithm to determine the optimal entanglement distribution along the path). Therefore, at the network scale, only path selection is required to fulfill user demands and satisfy network resource constraints.

The following P-TDP formulation is basically a path selection problem constrained by available resources in the network which aims to minimize the total resource cost.

$$\textbf{P-TDP}: \min_{m,c,\phi,x} \sum_{v\in V} p_v + \sum_{e\in E} p_e \tag{9}$$

s.t. entanglement distribution constraints

$$\sum_{p\in P_{i,j}} x_{i,j,p} \geq d_{i,j}, \qquad \forall i,j \in V \tag{10}$$

resource constraints

$$\sum_{i,j\in V, p\in P_{i,j}, e\in p} \alpha_{i,j,p,e} x_{i,j,p} \leq \phi_e \qquad \forall e \in E \tag{11}$$

$$(6), (7)$$

decision variables

$$m_v, c_e, x_{i,j,p} \in \mathbb{Z}_0^+, \quad \phi_e \in \mathbb{R}_0^+ \tag{12}$$

Constraint (10) makes sure that each node pair obtains enough entanglements for their demand. In Constraint (11), $\alpha_{i,j,p,e}$ is the number of entanglements used on edge e by the path p for unit entanglement between source-destination pair (i, j). So, this constraint ensures that the total entanglements consumed by all paths do not exceed the generated entanglements on this edge. Here, $\alpha_{i,j,p,e}$ is not a decision variable, instead, it represents the path-level entanglement solution to the predetermined path, which is an input for P-TDP. Constraints (6) and (7), similar to those in F-TDP, limit the used memory and entanglements on edge not to exceed available resources.

It is easy to see that this type of method reduces the complexity of the TDP problem significantly. While F-TDP has to consider all three-tuple of flows ($O(|V|^3)$ in total), a path-level solution only involves $O(L)$ nodes so only needs at most $O(L^3)$ time to solve each path, where L is the path length. In the next section, we will see that $O(L)$ (rather than $O(L^3)$) is sufficient to generate the path-level entanglement solution for one path. Therefore, we can use $O(\delta kL)$ time to obtain solutions to all paths, where δ is the number of demanded user pairs and k is the number of candidate paths between each user pair. Finally, the P-TDP is an MILP, we only have $O(\delta k)$ $x_{i,j,p}$ (other decision variables are the same as F-TDP's). The MILP is greatly simplified when k and δ are constants. That is, the number of decision variables is reduced from $O(|V|^3 + |E|)$ to $O(|V| + |E|)$.

Now P-TDP is straightforward as a resource allocation problem. However, there are still two key problems: (1) the path-level entanglement solution is critical to the final result, but it is unclear what is the optimal path-level solution; (2) even given the optimal path-level solutions, what is the loss of optimality (compared to the flow-based solution)? For the first problem, we prove Theorem 1 in Section VI to find the optimal path-level solution. For the second problem, we show in evaluations that the optimality loss is limited. In addition, compared to the path-based solution with ordinary sequential swapping, our optimal path solution significantly improves the network throughput. Our experiments later also confirm that without our optimal path-level solution, the objective value of P-TDP is obviously worse than F-TDP's.

Algorithm 1 P-TDP-GREEDY

```
Input: network $G(V,E)$, demands $D$, and path number $k$.
Output: $M_V$, $C_E$, $F_V$.
1: Find all candidate paths $P$, where $P_{i,j}$ includes $k$ paths
   for each $d_{i,j} \in D$
2: Solve path-level solutions for each candidate path in $P$
3: $M_V, C_E, F_V = \emptyset$
4: while max($d_{i,j} \in D$) $> 0$ do
5:     Pick $d_{i,j} \in D$, where $d_{i,j} > 0$
6:     Find best $p$ by trying $b_p, m_v, c_e, f_v$ = ESTIMATE-COST($p$) for all $p \in P_{i,j}$
7:     Update $M_v, C_E, F_V$ with $m_v, c_e, f_v$
8:     $d_{i,j} = d_{i,j} - 1$
9: return $M_V$, $C_E$, $F_V$
```

B. Greedy Path Selection Solution

With the P-TDP formulation, the original TDP can be solved more efficiently via existing linear solvers than F-TDP. However, P-TDP is still a MIP which can be challenging to solve in the worst case, especially with large-scale networks. Therefore, we also consider a greedy heuristic with polynomial time in this section.

Compared to P-TDP, which globally optimizes the path selection, the greedy method (Algorithm 1) uses a greedy strategy to select one path at each time, until all user demands are satisfied. It works as follows: i) select an unsatisfied demand (Line 5), and try to prepare one entanglement for this user pair for each candidate path; ii) among all paths for this pair, select the one that increases the least cost of the whole network (Line 6); iii) add necessary resources to M_V and C_E to enable this path, and add its path-level solution to F_V (Line 7). ESTIMATE-COST(p, d) (Line 6) returns b_p, m_v, c_e, f_v, where b_p is the cost change of the whole network by creating a new entanglement via path-level solution p, m_v and c_e are the new resources needed, and f_v is the swapping strategies for all nodes on the path. The candidate path set P (Line 1), in our implementation, for each SD pair, includes k shortest paths given by Yen's algorithm [27].

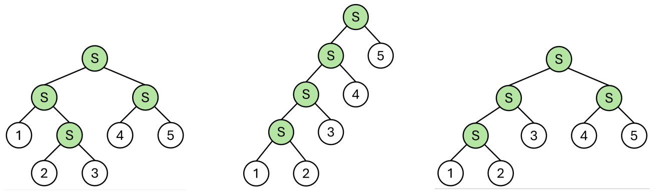
VI. OPTIMAL SWAPPINGS ALONG PRE-DEFINED PATH

In this section, we present our analysis of the path-level entanglement over a fixed path via swapping.

A. Definitions and Main Theorems

We first introduce a few tree definitions used in our analysis.

A **Swapping Tree** (ST) is a binary tree defined over a path that i) is a full binary tree, and ii) its leaves are *properly ordered*. As shown in Fig. 3a, an ST must be a full binary tree because all swappings must have two entanglements as input. An obvious by-product of this property is that there are always at least two leaves as siblings in the deepest layer. This property is useful in later proofs. Properly ordered leaves are those whose order in the in-order traversal of the tree (not



(a) ST (b) SST (c) CST

Fig. 3: Examples of swapping tree (ST), sequential swapping tree (SST) and complete swapping tree (CST) with five leaves (i.e., five edges on the quantum path). Here, a node marked with S is the swapping operation node.

including internal nodes) is the same as they are in the path. This property ensures that only adjacent entanglements are fed to swappings.

A **Sequential Swapping Tree** (SST) is an ST where the right child of any internal node is always a leaf. SST depicts the sequential swapping strategy over a path, as in Fig. 3b. SST is the highest tree among all trees with the same leaves.

A **Complete Swapping Tree** (CST) is an ST that is also a complete binary tree. Note that Fig. 3a is not a CST as its nodes 2 and 3 are not placed leftmost, while Fig. 3c is. CST has the smallest height (among all trees with the same leaves).

A **Relaxed Complete Swapping Tree** (RCST) is an ST whose all leaves are either in the last or penultimate layer. RCST is named as such because it is similar to CST, except that the nodes in the last layer are not required to be placed the leftmost. Thus, RCST has the same height as CST's. For example, Fig. 3a and the two trees in Fig. 4 are all RCST, but none is CST.

Now we present our conclusions of analysis on the optimal path-level entanglement in terms of its expected cost.

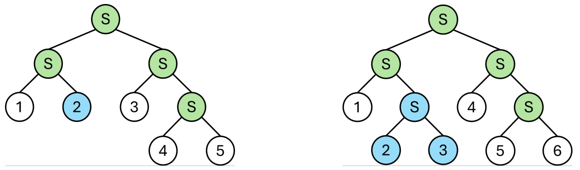
Theorem 1: To establish one E2E entanglement along path $e_1, e_2, ..., e_n$, the least number of consumed entanglements in expectation is given by an RCST at $R_n = \frac{1}{q^d}(2n - 2^d) + \frac{1}{q^{d-1}}(2^d - n)$, where $d = \lceil \log_2 n \rceil$.

Theorem 2: To establish one E2E entanglement along path $e_1, e_2, ..., e_n$, sequential swapping (i.e. by an SST) costs the largest number of consumed entanglements in expectation among all swapping strategies, which is $S_n = \frac{1}{q^d} + \sum_{i=1}^{d} \frac{1}{q^i}$.

B. Two Lemmas on Swapping Tree

Lemma 1: The total resource (entanglement) cost of an n-leaves ST over path $p = \{e_1, e_2, ..., e_n\}$ is $T_n = \sum_e \frac{1}{q^{d_e}}$ where q is the swapping success probability and d_e is the depth of edge e in the ST.

Proof: We know that for any node A with subtrees B and C as its children in an ST, the cost of A in expectation is $r_A = \frac{r_B + r_C}{q}$ where r_B/r_C is the expected cost of subtrees B/C. It is straightforward to see that the cost of a parent node is the sum of its two children as the two input entanglements are consumed in a swapping. Besides, the swapping may fail, so the expected cost is divided by the successful rate. The cost of the whole ST (at the root node) can be recursively calculated as such. Resolving the root cost, we can see that each node contributes to the root cost by dividing q each time it participates in a swapping, i.e., $T_n = \sum_e \frac{1}{q^{d_e}}$. For example, in Fig. 3a, $T_n = \frac{\frac{r_1 + \frac{r_2+r_3}{q}}{q} + \frac{r_4+r_5}{q}}{q} = \frac{r_1}{q^2} + \frac{r_2}{q^3} + \frac{r_3}{q^3} + \frac{r_4}{q^2} + \frac{r_5}{q^2}$. ■



(a) Before (b) After

Fig. 4: Adding a new leaf to an ST as node 2' sibling. (a) before adding the node, (b) after adding the node.

Lemma 2: When adding a sibling to a leaf at the d'th layer of an ST T_n with n leaves, its cost change is $\delta_d = -\frac{1}{q^d} + \frac{2}{q^{d+1}}$; when removing a leaf (whose sibling is also a leaf) at the d'th layer of T_n, its cost change is $-\delta_{d-1} = \frac{1}{q^{d-1}} - \frac{2}{q^d}$.

Proof: Take Fig. 4 as an example. By adding a sibling to node 2 in this tree T_5 in Fig. 4a, we create a new swapping node whose children are node 2 and the newly added node in the tree T_6 in Fig. 4b. Note that the node labels in T_6 are updated as there are more nodes after this addition. Compared to T_n, T_{n+1} no longer has the deleted node, whose cost is $-\frac{1}{q^d}$, but has two more nodes in the next layer, each of whose cost is $\frac{1}{q^{d+1}}$. Thus, the cost change is $-\frac{1}{q^d} + \frac{2}{q^{d+1}}$.

Conversely, when removing a leaf (whose sibling is also a leaf) at the $d+1$'th (not d'th), the cost change is $-\delta_d$; when removing a node at the d'th layer, the cost change is $-\delta_{d-1}$. A corner case is where T_n is a perfect binary tree, so adding a leaf increases the height of the whole tree by 1. However, one can easily verify that the lemma is still true as the depths d and $d+1$ remain the same for the affected nodes. ■

C. Proof of Theorem 1

We prove Theorem 1 in two steps. First, we prove that the cost of an RCST with n leaves is $R_n = \frac{1}{q^d}(2n - 2^d) + \frac{1}{q^{d-1}}(2^d - n)$. Then, we prove that this is optimal among all possible swapping trees.

First, an RCST has the same cost as the CST's because they have the same number of leaves in the same layer. In other words, RCST and CST always have the same cost, due to the same calculation via Lemma 1. Specifically, we know that the numbers of leaves in the last and second last layers (in the CST) are $(2n - 2^d)$ and $(2^d - n)$. Then, the cost contributions of each leaf in the last and the second last layers are $\frac{1}{q^d}$ and $\frac{1}{q^{d-1}}$, respectively. By Lemma 1, we obtain the cost of RCST and CST by summarizing the costs of all leaves.

We then prove that the RCST's cost is optimal by induction on the number of leaves. For a 2-leaves tree, the RCST is optimal because it is the only tree. Suppose RCST R_n is optimal for the n-nodes path, then we want to prove RCST R_{n+1} is optimal for the $(n+1)$-nodes path. Assume RCST is not the optimal but a non-RCST tree Z_{n+1} is the optimal tree for the $n+1$ case. Let d_R be the depth of R_{n+1}, and d_Z be the depth of Z_{n+1}. Because RCST is also CST, d_R is the smallest height for any binary trees with $n+1$ nodes. That is, $d_R \leq d_Z$. In addition, Z_{n+1} is not RCST, so we have $d_R < d_Z$. Removing a leaf (whose sibling is

TABLE I: Network description.

Scale	Topology	Node # $\lvert V \rvert$	Diameter	SD # δ
Small	EEnet	12	351.71 km	72
Medium	NOEL	19	554.61 km	171
Large	Renator	37	1202.5 km	666

also a leaf) in the d_Z'th layer from Z_{n+1}, we obtain an ST Z_n, whose cost is $Z_n = Z_{n+1} - \delta_{d_Z-1}$. Note that such a leaf always exists as Z_{n+1} is an ST, so we can apply Lemma 2 here. Similarly, we have $R_n = R_{n+1} - \delta_{d_{R-1}}$. Since $Z_{n+1} < T_{n+1}$, we have $Z_n + \delta_{d_Z-1} < R_n + \delta_{d_R-1}$. Thus $Z_n < R_n - [(2-q)(\frac{1}{q^{d_Z}} - \frac{1}{q^{d_R}})]$. As $d_R < d_Z$, $(2-q)(\frac{1}{q^{d_Z}} - \frac{1}{q^{d_R}})$ is always positive. That is, $Z_n < R_n$, which contradicts that R_n is optimal.

We can easily construct path-level optimal solutions by constructing RCSTs, among which the CST is the simplest.

D. Proof of Theorem 2

The proof is also by induction, similar to the second part of Theorem 1. It is easy to see that S_4 is the worst swapping tree for 4-hop paths: there are only two distinct 4-leaves trees (in terms of cost): R_4 and S_4. We already know that R_4 is optimal, so S_4 is worst.

If S_n is the worst tree, then we prove that S_{n+1} is also the worst tree with $n+1$ leaves. If not, suppose W_{n+1} is the worst for $n+1$ leaves (and it is not an SST). Then, we have $S_{n+1} < W_{n+1}$. Suppose d_S and d_W are the depths of S_{n+1} and W_{n+1}. Because S_{n+1} is the highest ST, we have $d_S = n > d_W$. By deleting one of the two deepest nodes in S_{n+1} and W_{n+1}, separately, we can obtain the cost of S_n and W_n, which are $S_n = S_{n+1} - \delta_{d_S-1}$ and $W_n = W_{n+1} - \delta_{d_W-1}$. Plug them in $S_{n+1} < W_{n+1}$, $S_n + \delta_{d_S-1} < W_n + \delta_{d_W-1}$. Thus $S_n < W_n - (2-q)(\frac{1}{q^{d_S}} - \frac{1}{q^{d_W}})$. For the same reason, $S_n < W_n$, which contradicts that S_n is the worst.

VII. Performance Evaluations

We first introduce our experiment settings and how to construct the initial topology for the formulated problems. Then, we report our experimental results which aim to answer i) how the the proposed solutions perform in terms of optimality and efficiency; ii) how to find good settings to construct a quantum network for given users and their demands.

A. Simulation Settings

Quantum Network Topology. We consider three real-world networks of different sizes from the Internet Topology Zoo [28]: Small, Medium, and Large, as specified in Table I, for $G(V, E)$. Although these networks come with their own topology, we do not want to limit our optimization within existing edges. Instead, we only use nodes' locations in these networks, and connect each node to its closest ζ neighbors[3]. The distance between two nodes is calculated based on their coordinates using pyGeo [29]. If the graph is still not connected, we repeatedly connect the closest components to make it connected. For long edges (longer than a threshold) in the graph, we need to segment them since long edges can hardly generate entanglements (the original edges are not deleted but still left for larger optimization space). The default threshold is set to 150km, which is set according to later experiments.

Entanglement Settings. Total SD pair number δ is set to half of the total possible pair number, i.e., $\frac{1}{4}|V|(|V|-1)$. Candidate path number k (for each SD pair) is set to 100. The quantum channel parameters are set according to recent physical implementation experiments of entanglement generation [30]–[33]. Specifically, the typical telecommunication wavelength 1550 nm is adopted, and the fiber loss is 0.2 dB/km. With such parameters, the entanglement rate is 100 bps per channel at a 100 km distance (similar to the rate in the above-cited experiments). The concrete capacity/rate for each channel is calculated based on their real length and the above parameters. For all quantum channels, we use the 'meet-in-the-middle' scheme, i.e., the detectors are placed in the middle of fibers and light sources are placed at the two sides. Device initialization cost for repeaters/links (i.e., γ_v^0 & γ_e^0) are set to $10\gamma_v$ & $10\gamma_e$, respectively.

Baselines. We consider the three TD formulations (Flow, Path and Greedy) and their variations as baselines: (1) FLOW: the F-TDP formulation; (2) PATH-SEQ: the P-TDP formulation using the sequential path-level solution; (3) PATH-OPT: the P-TDP formulation using our optimal path-level solution RCST. (4) GREEDY-SEQ: the greedy algorithm using sequential path-level solution; (5) GREEDY-OPT: the greedy algorithm using our optimal path-level solution. Note that the sequential swapping scheme is the most popular default scheme in existing works, such as [15], [16], [18].

Implementation. Initial graphs are constructed and manipulated using networkx [34]. Linear solver Gurobi [35] is used to obtain solutions for FLOW and PATH. The reported running times include all time used by the corresponding methods, such as the model-building time for all methods and path-solving time for PATH and GREEDY variants.

B. Optimality and Efficiency of TD Methods

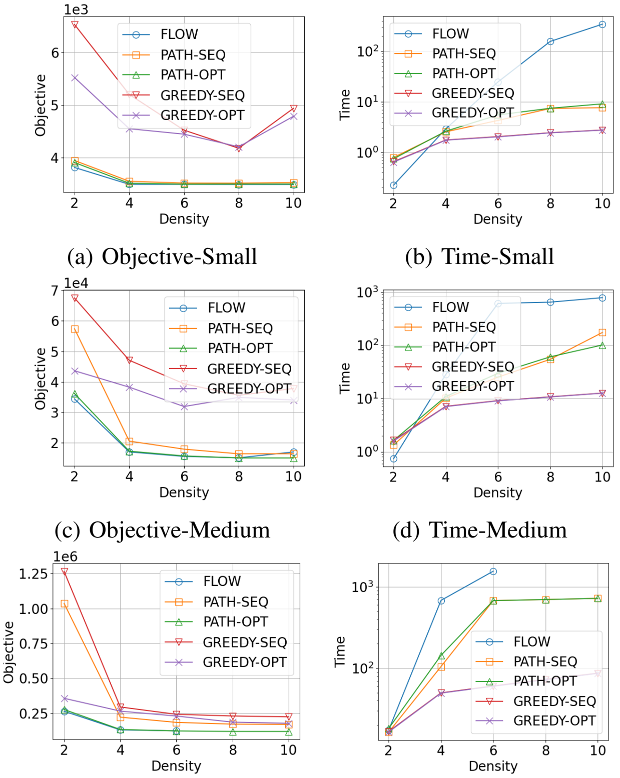
We compare all TD methods for objective values and running times under different settings.

Different Network Density. We know that a denser initial network topology G allows a better solution but requires a longer solving time. Therefore, we first compare the baselines to see how they perform on networks of different density ζ. Results are reported in Fig. 5 and we find the following discoveries.

Generally, FLOW has the best objective value (i.e., smallest cost) and longest running time; two PATH variations cost medium time and have medium costs; two GREEDY variations are the most efficient but their costs are the relevantly worst.

For the FLOW formulation, if optimally solved, gives us the optimal objective on the given candidate graph. However, solving the FLOW formulation may take a longer time. In Medium and Large networks, its running time does not increase on denser initial graphs because we set the solver time to 600 seconds. All methods stop optimization after this upper bound.

[3] We call ζ the graph density parameter. If $\zeta = |V| - 1$, the candidate topology becomes a clique (complete graph). Obviously, a denser candidate graph allows larger optimization space, leading to a possibly better objective value but also harder to solve.



(a) Objective-Small (b) Time-Small

(c) Objective-Medium (d) Time-Medium

(e) Objective-Large (f) Time-Large

Fig. 5: Objective values and running time of all methods with different density ζ and network sizes. Here $q = 0.75$.

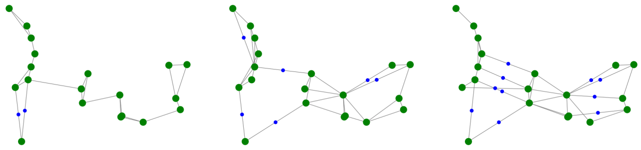
That is why FLOW's running time does not increase much after $\zeta = 6$ in the Medium network. For a Large network, FLOW needs even more time to build the model (as there are too many node pairs) before optimization, which exceeds 10^4 seconds for $\zeta = 7$, so we do not show its performance for a denser Large graph. For $\zeta = 10$ of Medium network, FLOW is terminated before it obtains a good solution, so its objective is no longer optimal and is worse than PATH-OPT's.

When considering the path-level swapping solution, obviously our solution of OPT is better than the SEQ solution. This confirms the conclusion from our proven main theorems.

More importantly, we can see that the objective value given by PATH-OPT is very close to FLOW's, which means that PATH-OPT only loses marginal optimality. At the same time, it is much more efficient in terms of running time.

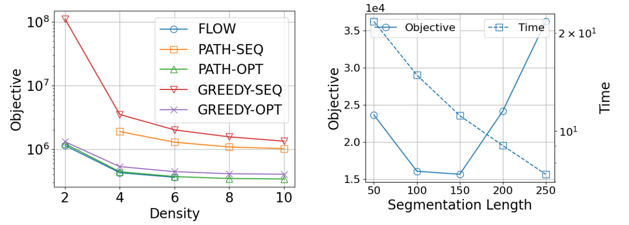
Fig. 6 shows examples of the resulted topology as the density of the candidate network increases. In these topologies, larger green dots represent the user's nodes (SD nodes), while smaller blue dots are repeaters. We can see that a denser initial network has more available edges and repeaters, hence larger optimization space and less cost as shown in Fig. 5.

Swapping Success Probability. We also investigate how lower swapping success probability q affects the topology design. Fig. 7a shows the result over the Large network with $q = 0.5$. First, PATH-OPT keeps being close to FLOW and better than all others. A notable thing is that GREEDY-OPT can now beat PATH-SEQ. That means, when q is smaller, better path-level swapping solution is more important than global path selection. Second, PATH-SEQ cannot even solve the problem when $\zeta = 2$. This is because the paths are typically longer in sparser graphs and SEQ path solutions are extremely ineffective. Last, compared to $q = 0.75$ case in Fig. 5e, PATH-SEQ and GREEDY-SEQ perform even more incompetent than their OPT counterparts. All these show that when E2E entanglements over a path are harder to establish (i.e., lower q and longer path), our path-level optimal swapping solution is getting more critical.



(a) $\zeta = 2$ (b) $\zeta = 4$ (c) $\zeta = 6$

Fig. 6: Resulted topology by PATH-OPT for Medium networks with different initial density ζ.



(a) Objective-Large (b) Objective-Medium

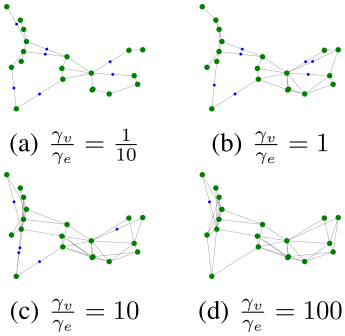
Fig. 7: (a) Objective values of all methods when swapping success probability $q = 0.5$ for Large network. (b) Objective values/running times of all methods with different segmentation lengths for Medium network.

C. Network Planning Consideration

With the help of PATH-OPT, we explore how to find good parameters to build good topology for quantum networks.

Segmentation Threshold. Segmentation threshold (the length to decide the segmentation of quantum link for better entanglements) is perhaps the most interesting parameter. As shown in Fig. 7b, 150 km is the best length, which we use as the default. If the threshold is too small, more intermediate nodes are placed and more memory is used to generate E2E entanglements along the long paths; when the threshold is too large, we need fewer intermediate nodes but have to deploy more channels as each channel is significantly weaker for entanglement generation. Besides, the solving time (i.e., running time) monotonously decreases as the segmentation length increases. A higher threshold results in a sparser graph, leading to a simpler problem and reduced solving time.

Price Ratio. Now we investigate how the price of quantum devices impacts the network topology. We know that quantum optics and memory can be costly, however their prices depend on different technologies that are still under active development. Therefore, we explore the impact on quantum network topology by the relative price of the two important devices, i.e., quantum channels and memory. We define the price ratio as the price of one memory slot to the price of one kilometer of one single quantum channel, i.e., $\frac{\gamma_v}{\gamma_e}$. We explore four ratios: 10:1, 1:1, 1:10, 1:100, where the quantum memory price increases from less to more. Fig. 8 shows that when memory is relatively cheap, the resulting topology uses more intermediate nodes; when memory is more expensive the number of repeaters is significantly reduced, e.g., from 7 ($\frac{\gamma_v}{\gamma_e} = 1$) to 1 ($\frac{\gamma_v}{\gamma_e} = 100$). And in the latter case (with expensive memory) many edges no longer use the intermediate nodes introduced in segmentation, but directly connect far-away nodes via longer direct quantum channels (which may be weak for entanglement generation, but are cheap enough).


(a) $\frac{\gamma_v}{\gamma_e} = \frac{1}{10}$ (b) $\frac{\gamma_v}{\gamma_e} = 1$

(c) $\frac{\gamma_v}{\gamma_e} = 10$ (d) $\frac{\gamma_v}{\gamma_e} = 100$

Fig. 8: Price ratio's impact on Medium network when $\zeta = 5$.

Demand Intensity. Finally, we test different user demand intensities by setting the fraction of SD pairs from total pairs to $0.1, 0.5, 1.0$ with the demand of each pair as $1, 10, 100$ correspondingly. We call these three cases as Low, Medium, and High demand intensity. Fig. 9 shows that the resulted topology becomes denser for higher demand intensity. Note that in lower-intensity case, some user nodes are removed since they do not have demands and are not used for others as intermediate nodes.

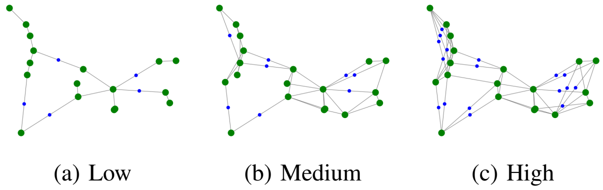

(a) Low (b) Medium (c) High

Fig. 9: Optimized network topology for Medium network, with Low, Medium, and High demand intensity.

VIII. Conclusion

In this paper, we study the topology design problem for quantum networks and introduce two formulations along with their corresponding solutions. Our findings indicate that while the flow-based formulation achieves optimality without loss, it is challenging to solve. On the other hand, the path-based formulation, when enhanced with our path-level entanglement solution (proved to be optimal among all path-level swapping strategies), proves to be empirically efficient with only a marginal loss in optimality. Additionally, we investigate the impact of key factors (such as edge segmentation, device cost, and user demands) on the network topology using the path-based solution. We plan to investigate further improvements of our solutions via smarter repeater placements.

References

[1] A. Dahlberg, M. Skrzypczyk, *et al.*, "A link layer protocol for quantum networks," in *Proc. of ACM SIGCOMM*, 2019.

[2] Y. Wang, A. N. Craddock, *et al.*, "Field-deployable quantum memory for quantum networking," *Physical Review Applied*, 18(4):044058, 2022.

[3] F. Li, *et al.*, "Reliable topology design in time-evolving delay-tolerant networks with unreliable links," *IEEE TMC*, 14(6):1301–1314, 2015.

[4] X. He, S. Chen, *et al.*, "Efficient topology design in time-evolving and energy harvesting wireless sensor networks," in *Proc IEEE MASS*, 2013.

[5] Y. Wang, F. Li, and T. A. Dahlberg, "Energy-efficient topology control for 3-dimensional sensor networks," *IJSNet*, 4(1/2):68–78, 2008.

[6] X.-Y. Li, *et al.*, "Efficient topology control for wireless ad hoc networks with non-uniform transmission ranges," *WINET*, 11(3):255-264, 2005.

[7] X. Wei, J. Liu, *et al.*, "Optimal entanglement distribution problem in satellite-based quantum networks," *IEEE Network*, 2024.

[8] X. Wei, *et al.*, "Optimizing satellite-based entanglement distribution in quantum networks via quantum-assisted approaches," in *QCNC*,2024.

[9] W. Dai, T. Peng, and M. Z. Win, "Optimal remote entanglement distribution," *IEEE J. on Sel. Areas in Commu.*, 38(3):540–556, 2020.

[10] H. Gu, Z. Li, *et al.*, "Fendi: High-fidelity entanglement distribution in the quantum internet," *arXiv preprint arXiv:2301.08269*, 2023.

[11] H. Gu, R. Yu, *et al.*, "ESDI: Entanglement scheduling and distribution in the quantum internet," *arXiv preprint arXiv:2303.17540*, 2023.

[12] A. Farahbakhsh and C. Feng, "Opportunistic routing in quantum networks," in *Proc. of IEEE INFOCOM*, 2022.

[13] Y. Zeng, J. Zhang, *et al.*, "Multi-entanglement routing design over quantum networks," in *Proc. of IEEE INFOCOM*, 2022.

[14] L. Yang, Y. Zhao, H. Xu, and C. Qiao, "Online entanglement routing in quantum networks," in *Proc. IEEE/ACM IWQoS*, 2022.

[15] Y. Zhao, *et al.*, "E2E fidelity aware routing and purification for throughput maximization in quantum networks," in *IEEE INFOCOM*, 2022.

[16] J. Li, M. Wang, *et al.*, "Fidelity-guaranteed entanglement routing in quantum networks," *IEEE Trans. on Commu.*, 70(10):6748–6763, 2022.

[17] R. Yu, R. Dutta, and J. Liu, "On topology design for the quantum internet," *IEEE Network*, 36(5):64–70, 2022.

[18] S. Pouryousef, *et al.*, "Quantum network planning for utility maximization," in *Workshop on Quantum Net. & Distr. Quantum Comp.*, 2023.

[19] M. Chehimi, S. Pouryousef, *et al.*, "Scaling limits of quantum repeater networks," in *Proc. of IEEE QCE*, 2023.

[20] S. Pouryousef, *et al.*, "A quantum overlay network for efficient entanglement distribution," *arXiv preprint arXiv:2212.01694*, 2022.

[21] A. Chang and G. Xue, "Order matters: On the impact of swapping order on an entanglement path in a quantum network," in *Proc. of IEEE INFOCOM Workshops*, 2022.

[22] M. Ghaderibaneh, *et al.*, "Efficient quantum network communication using optimized entanglement swapping trees," *IEEE TQE*, 3:1-20, 2022.

[23] L. Jiang, *et al.*, "Optimal approach to quantum communication using dynamic programming," *PNAS*, 104(44):17 291–17 296, 2007.

[24] M. J. Bayerbach, *et al.*, "Bell-state measurement exceeding 50% success probability with linear optics," *Science Advances*, 9(32):eadf4080, 2023.

[25] F. Ewert and P. van Loock, "3/4-efficient bell measurement with passive linear optics and unentangled ancillae," *PRL*, 113(14):140403, 2014.

[26] A. Kamimaki, *et al.*, "Deterministic bell state measurement with a single quantum memory," *npj Quantum Information*, 9(1):101, 2023.

[27] J. Y. Yen, "An algorithm for finding shortest routes from all source nodes to a given destination in general networks," *Quarterly of applied mathematics*, 27(4):526–530, 1970.

[28] S. Knight, H. Nguyen, *et al.*, "The internet topology zoo," *IEEE JSAC*, 29(9):1765 –1775, 2011.

[29] H. M. Hajdik, *et al.*, "pyGeo: A geometry package for multidisciplinary design optimization," *J. of Open Source Software*, 8(87):5319, 2023.

[30] S. Wengerowsky, S. K. Joshi, *et al.*, "Entanglement distribution over a 96-km-long submarine optical fiber," *PNAS*, 116(14):6684–6688, 2019.

[31] Y. Zhang, Z. Chen, *et al.*, "Long-distance continuous-variable quantum key distribution over 202.81 km of fiber," *PRL*, 125(1):010502, 2020.

[32] W. Li, L. Zhang, *et al.*, "High-rate quantum key distribution exceeding 110 mb s–1," *Nature photonics*, 17(5):416–421, 2023.

[33] Y. Liu, W.-J. Zhang, *et al.*, "Experimental twin-field quantum key distribution over 1000 km fiber distance," *PRL*, 130(21):210801, 2023.

[34] A. Hagberg, *et al.*, "Exploring network structure, dynamics, and function using networkx," Los Alamos National Lab.(LANL), Tech. Rep., 2008.

[35] Gurobi Optimization, LLC, "Gurobi Optimizer Reference Manual," 2023. [Online]. Available: https://www.gurobi.com